SOFTWARE SUPPORT FOR AUTOMATUCALLY GENERATED PROCEDURES

D.E. VETROV, S.S. CHERNOVA
International Research Institute for Advanced Systems
Moscow, RUSSIA
e-mail: vetrov@irias.ru
Institute for System Analysis Russian Academy of Sciences
Moscow, RUSSIA
e-mail: chernova@cpt-ran.ru

Abstract

The paper explicates a construction technique and provides a usage example of a software integrator designed to create diverse data processing implementations based on the automatically generated data-specific processing procedures.

1 Introduction

The design and implementation of a complex engineering object involves the analysis and comparison of various engineering solutions with regard to the object's structure, operating mechanisms, the selection of its parameters and other elements. Significant advance in mathematical modelling and enhanced capabilities of up-to-date computers allow a wide spectrum of studies to be performed without full-scale experiments. Therefore, mathematical modelling which involves series of computational experiments for the investigation of analytical models of an object and its environment has become one of the most common methods of analysis and optimization of an engineering object's structure.

Various computer-aided design systems are being created in order to reduce the design time and full-scale experiment costs and improve the quality of design. They are devised as special-purpose simulation and decision support systems that allow developers to investigate a large number of design alternatives (configuration, parameters, etc.), predict an object's performances and find the best (rational) solutions.

Modern cognitive technologies open up new possibilities for the simulation and decision support systems, and primarily for constructing data-based computer models. A distinctive feature of a data-based model is that it is trained on some set of prototype data and is sure to be efficient only if applied to the input data similar to the set of prototypes the model is based upon. To put it otherwise, the model has to be rebuilt each time a new learning data set is introduced.

These opportunities impose new challenges in terms of software support for the data modelling and analysis systems:

- automatic generation of data processing procedures;

- integrating the existing and generated procedures into a single computational model.

The paper explains how to construct and use a software integrator which allows creating various data processing implementations based on the automatically generated data-specific processing procedures.

2 Purposes

Software integrator is a software tool for designing and running computational schemes that the user creates from individual data processing computer models in dialog mode.

Computer model is a software component that implements some mathematical data processing algorithm or its part. Being an element of the software, the computer model is built from dynamic link libraries (DLL). Therefore, the functionality of the computer model's modules is linked to the integrator at the execution phase and not the assembly (compilation) phase.

Computational scheme is an oriented graph in which the nodes are associated with individual computer models and the arcs show consecutive interactions between the models.

Computer models included in the computational scheme can be either selected from a set of existing models or created anew. There must be a possibility to connect automatic procedure generators to the software integrator for the user to create new computer models while designing the computational scheme.

The software integrator groups both the individual data processing procedures and their generators into a single data processing and analysis system. External to the software integrator, either of these components can be used independently of one another in designing other engineering applications, with the software integrator operating as a graphic shell the computer models are linked to. Note that the computer models stay passive at the software integrator assembly phase and are activated by calling their functions.

Thus the software integrator provides the user with a graphic interface for:

- designing computational schemes grouping several computer models from a given set;
- specifying the input data and conducting the experiments;
- saving and viewing the process status with the progress and completion of individual steps and experiment results;
- tuning and refining the computational logic and inter-model communication formats;
- adding new models to the set by generating new procedures implementing data-specific processing algorithms without compilation or assembly of the integrator's software.

3 Implementation example

The authors designed the software integrator as a Windows application with a graphic user interface allowing the user to:

- create data-based software modules based on a given data set using automatic procedure generators;
- validate the generated procedures by applying them to the specified test data set;
- create computational schemes with the aid of the created procedures;
- conduct computational experiments.

The developed integrator rests upon several automatic procedure generators:

- dimension reduction procedure generator;
- high dimensional approximation procedure generator;
- geometry generation procedure generator.

The software integrator is oriented towards engineers and analysts specializing in data handling. The software engineers can use the data-based software modules designed by the authors to create new applications using C, C++ or Python compilers.

The integrator has the following functionality:

- importing an input file(s) of the specified formats containing the user data set;
- connecting any user software implemented as an import module and conducting experiments with the import module to create an input data set;
- automatic generation of a software model based on the specified input data using the selected procedure generator;
- validating the software model by applying it to the user-specified test data set;
- calculating a set of standard descriptive statistic characteristics commonly used in mathematical software packages;
- comparing the generated procedures belonging to one class in terms of quality.

4 Application

The software integrator was successfully applied to the wing aerodynamic airfoil generation task. The task was addressed as follows. Since the digital description of the initial object comes in the form of a high dimension vector, the first step was to find a manifold of smaller dimension that would provide a fairly accurate approximation of the input data set. The next step was to generate a new object in the smaller dimension space and then reconstruct the full dimension vector. The solution was obtained using the following computational scheme:

- specifying an arbitrary data set consisting of civil passenger aircraft airfoil descriptions and stating the requirement for the accuracy of the dimension reduction procedure to be created;
- automatically creating a dimension reduction procedure using the procedure generator;
- selecting an arbitrary subset from the specified set of airfoils and using it as the input set of prototypes for automatic creation of the random generation procedures;
- applying the procedure generator to automatically create a random airfoil generation procedure;
- generating new airfoils using the automatically created random generation procedure.

The computational experiments performed with automatically generated dimension reduction and geometry generation procedures demonstrated that the computational scheme, when applied to different sets of input prototypes, can be used for generating new aerodynamic airfoils with pre-specified or predicted properties.

References

[1] Kuleshov A. (2008). Cognitive technologies in adaptive models of complex objects. *Information technologies and computer systems.* Vol. **1**, pp. 18-29.

[2] Bernstein A., Kuleshov A. (2008). Mathematical Methods in Engineering Cognitive Technologies. *Survey in Applied and Industrial Mathematics, Ser. "Probability and Statistics"*. Vol. **15**, No. **3**, pp. 451-452.

[3] Bernstein A., Burnaev E., Dorofeev E. Sviridenko Y. Chernova S. (2008). Cascade dimension reduction procedures. *Proceedings of the 11th National Conference with International Participation on Artificial Intelligence, Dubna.* Vol. **1**, pp. 241-250.

[4] Bernstein A., Kuleshov A. (2008). Cognitive technologies in the problem of dimension reduction of geometrical object descriptions. *Information technologies and computer systems.* Vol. **2**, pp. 6-19.

[5] Ivanova E., Chernova S. (2009). Reduction of complex geometrical object dimension in the presence of particular parametric models. *Artificial Intelligence and Decision Making.* Vol. **3**, pp. 53-58.