A formal approach for clustering classes in software components

Dan Laurenţiu Jişa

The components based development (CBD) tends to revolutionize the software development process, opening the possibility to construct software applications through the assembling of building blocks, represented by components. A software component must assure a better management of the application complexity, a reducing of the development costs, flexibility for applications. These goals can be achieved through a proper identification of the classes in components.

This paper describes a model for the identification of the software components based on the design models of the application, as well as a genetic algorithm used to achieve the clustering of classes in software components.

t considers a design model which consists of n classes, where n > 1. The goal is to clustering the classes in components, in order to obtain values for the quality attributes of the components as better as possible.

The following model is utilized:

- a matrix $M = (m_{ij})$, which describes the inheritance relationships among the classes of the model, where $i = \overline{1,n}$, $j = \overline{1,n}$;

- a matrix $A = (a_{ij})$, which describes the association relationships among classes, where $i = \overline{1,n}$, $j = \overline{1,n}$, a_{ij} = number of the association relationships between classes i and j.

- a matrix $Msg = (msg_{ij})$, for the exchanged messages among the classes of the model, where $i = \overline{1,n}$, $j = \overline{1,n}$, msg_{ij} = the number of messages sent by class i to class j.

- a matrix $Met_t = (met_{ij})$, where $t = \overline{1,r}$, for each of the internal metrics of the classes (it is made the assumption that are used r internal metrics, r > 0).

$$met_{ij} = \begin{cases} \text{value of the metric t for class i, if i=j} \\ 0, \text{ otherwise} \end{cases}$$

The purpose is to establish k components, where $1 \leq k \leq n-1$. For each component, it imposes constraints for the number of classes (for example: $1 \leq dim \leq n-1$). The classes will be heuristically grouped in components and for each cluster will be computed quality attributes.

The matrix $S = (s_{ij})$ represents a solution, where:

$$s_{ij} = \begin{cases} \text{index of the component to which the class belongs} \\ 0, \text{ if the class does not belong to any component} \end{cases}$$

For the identified components, the sum of the quality attributes (reusability, understandability etc.) values is computed. This will represent the function to be maximized.

References

[1] Hemant Jain, Naresh Chalimeda, Navin Ivaturi, Balarama Reddy, "Business Component Identification - A Formal Approach", Fifth IEEE International, Enterprise Distributed Object Computing Conference, 2001.

[2] Tao Xie, Huang Huang, Xiangkui Chen, "Object Oriented Software Metrics Technology", `www.cs.washington.edu/homes/taoxie/RicohmiddleReport.pdf` , 1999.

[3] Khaled El Emam, "Object Oriented Metrics: A Review of Theory and Practice", `http://citeseer.nj.nec.com/479219.html`, 2001.

[4] Michalewicz Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, Heidelberg, 1996.

[5] Booch G., Raumbaugh J., Jacobson I., "The Unified Software Development Process", Addison / Wesley 1999.

[6] Booch G., Raumbaugh J., Jacobson I., "The Unified Modelling Language. User Guide", Addison / Wesley 1999.