Development of a Design Optimization Interface to ABAQUS[1]

Dipankar K. Ghosh
and
Garret N. Vanderplaats

Vanderplaats Research & Development, Inc.
1767 8th Street South, Suite M-210, Colorado Springs, CO 80906

This presentation will describe the development of a design study tool to couple optimization technology with sophisticated analysis software graphical user interface. Primary objective has been to create a graphical user-interface (GUI) to the general purpose optimization software DOC/DOT, from V R&D, Inc. The second objective was to interface DOC/DOT with a number of third-party analysis software products to provide flexible optimization capabilities. DOC/DOT has already been interfaced with ABAQUS and other commercially available software, and the details of the developed capabilities will be discussed in this presentation.

A graphical user interface for the existing DOC software has been developed. This provides a useful tool for developing and testing new ideas as well as creating a user-friendly interface to the DOC software. Different GUI objects are used to generate project information consisting of a catalog of variables and responses, design variables, objective function(s), design constraint(s) and other problem parameters.

A catalog of variables and responses to be used in the optimization process is created first. The basic idea is to identify all the input and output parameters that will be used during optimization, and include them in the project catalog. From this project catalog, user can create multiple "task files" to solve particular design optimization problems. The project catalog information will be created only once, and could be used and accessed by one or more users to create their particular optimization task.

When the design variable window is opened, the user sees only those entries in the catalog that were identified as inputs, because these are the parameters that can be changed during optimization. Here the user specifies the variable(s) to be used during optimization, provides lower and upper bounds, if any, and initial values, if he wants to override the values specified in the user's program input data. He may also specify if a particular variable is continuous or discrete, or if it is linked linearly or non-linearly to other variable(s).

Similarly, when the user opens the objective function window, he sees all the variables, both input and output, listed here. He may tag one or more variable/response(s) as objective functions. Here multi-objective optimization is allowed, and the user may specify target value(s), weighting factor(s), etc. Like design variables, objective function(s) could also be linked with other variables, as well as with other responses.

If constrained optimization is to be performed, the user will open the constraint window. In this window, only variables that were tagged as output parameters (responses) in the project catalog are displayed. The user may select all or some of these responses as constraint function(s) to be used during optimization, and provides lower and/or upper bounds, and optional scaling factors. Constraints also can be linked to other variables and responses.

There are also a number of other windows and sub-windows to specify related information such as sets of discrete values of variables, equations for synthetic variables and responses, etc. The user can also specify various control parameters such as whether to use direct or approximate optimization, print control, convergence criteria, etc. This is done using the Control window.

Once the optimization task has been defined, the user can compile and link his/her analysis subroutine, if any, with DOC/DOT from within the GUI, and then run the optimization. The user may also view the results of optimization such as optimization history and critical constraint history by using appropriate windows and visualization tools provided within the graphical user interface.

A general methodology for interfacing third-party analysis software with DOC/DOT using the present GUI has a been developed. This directly couples third-party analysis software with the general purpose DOC/DOT optimizer in a user-friendly manner. An interface to ABAQUS has already been developed, and many other third-party candidate software are presently being considered for future implementations. This provides the optimization capability to analysis software wherein optimization has not yet been hard coded. This approach may not be the most efficient one, but provides usable optimization capabilities in third-party software.

Here user will open appropriate windows in the GUI, and edit his/her analysis input and output files to create a catalog of variables and responses. The input file(s) is used to create the catalog of possible design variables, and the output file(s) is used to create the catalog of responses, which could be used as possible objective function(s) and/or constraint(s) later in the optimization process. This is performed using a very flexible interactive approach. Information regarding the input and output variables, their values, locations in the files, and formats are stored in appropriate project files for later use. Once the project catalog has been defined, the above-mentioned approach could be followed to set up the design optimization tasks.

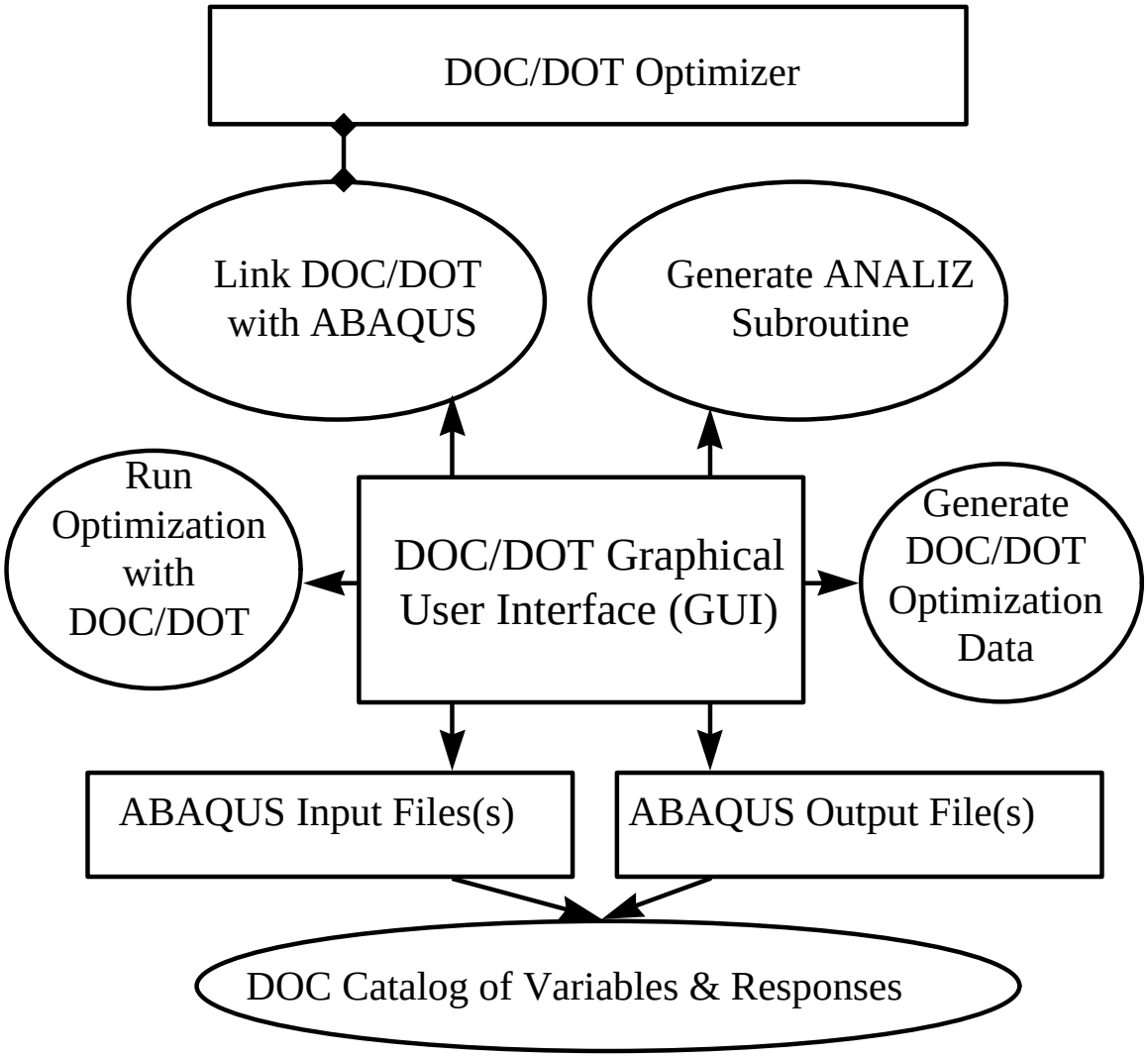
Based on the design project file just created, the GUI automatically generates a FORTRAN subroutine ANALIZ, which calls a particular third-party analysis software via a FORTRAN system call. This ANALIZ subroutine is compiled and linked with the DOC/DOT optimizer from within the GUI. This can also be done from outside the GUI. The ANALIZ subroutine makes use of the information stored in the project files to modify the analysis input file, run third-party software, extract output values of variables from the analysis output file, and provide DOC/DOT optimizer with all this information.

The present implementation is a very basic one, and many refinements are being incorporated on an ongoing basis.

The overall program capabilities will be demonstrated using physical examples.

References:

1. DOT USERS MANUAL, Version 4.20, Vanderplaats Research & Development, Inc., Colorado Springs, CO.
2. DOC USERS MANUAL, Version 1.30, Vanderplaats Research & Development, Inc., Colorado Springs, CO.
3. Ghosh, Dipankar K., and Garret, Vanderplaats N., ‘Development of a flexible design optimization capability’, presented at the *Optimization In Industry* Conference held on March 23-27, 1997, Palm Coast, Florida, USA.



DOC/DOT Graphical User Interface to Abaqus

[1] Presented at Regional ABAQUS Users' Meeting, Los Angles Airport Hilton and Towers, September 25, 1997