

# A NEW SYSTEM FOR PROBLEMTYPES CREATION

M. R. Peyrau<sup>1\*</sup>, R. Ribó<sup>2\*</sup>

Compass Ingeniería y Sistemas  
Tuset 8, 7 2  
08006 Barcelona, Spain

1: E-mail: [peyrau@compassis.com](mailto:peyrau@compassis.com), web page: <http://www.compassis.com>

2: E-mail: [ramsan@compassis.com](mailto:ramsan@compassis.com), web page: <http://www.compassis.com>

**Key words:** Problemtypes, Interface creation, Finite element method

**Abstract.** *In order to use GiD to prepare data for a specific analysis program, it is necessary to customize it. The customization involves the predefinition of all the information that must be entered by the user, as well as the definition of the way the final information is given to the corresponding solver. The configuration process is performed by means of a problemtype, which is a collection of files used to configure GiD for a particular analysis. In this way, the traditional GiD problemtype is defined with a directory and a set of files, which contains the full functionality for customizing the preprocess. The new Toolkit for problemtypes creation is based on these features and files, but uses additional capabilities provided with the Toolkit and does not need most of the original files anymore. Therefore, the new problemtype is also defined with a directory and a set of files, which contains a main XML configuration file, a main TCL file with the initialization routines, a file defining the way in which the final information has to be written inside the input files that will be read by the corresponding solver and several other TCL files. In order to add conditions, materials, general data or units systems information to the problem type, it will be necessary to modify the main configuration file, which is defined in XML format and contains all the definition of all the data (except the geometry) that must be filled by the user to perform the analysis. In this way, the Toolkit permits to process easily XML documents on a physical tree view inside the GiD window, to couple entities into groups using standard windows and to edit their properties. This powerful tool also groups all the common features of the different problemtypes, which facilitates its creation. A realistic application example of the Toolkit is the Plaxis-GiD interface program, which performs the simulation of 3D geotechnical problems by means of the finite element method.*

## 1 INTRODUCTION

When GiD is to be used for a specific type of analysis, it is necessary to predefine all the input dates required from the user like geometry properties and boundary conditions

(i.e. constraints, loads, velocities). It is also necessary to define the way in which this information should be given to the solver. In this way, the customization is done by creating a problemtype, which is a collection of files used to configure GiD for a particular analysis program. In fact, the traditional GiD problemtype is defined with a directory with the problemtype name and the extension .gid containing several files, where most of them have the same name than the directory with different file extensions referring to their particular functions. A new system for problemtypes creation has been developed, which is based on these original capabilities, but uses additional capabilities provided with the so called Toolkit.

## 2 THE NEW PROBLEMTYPES STRUCTURE

The new problemtype is defined using a directory with the problemtype name, but does not need most of the original files. In this way, the main configuration file with .spd extension is defined in XML format and contains the definition of all the data (except the geometry) necessary to perform a numerical analysis. Therefore, it is necessary to modify this XML file in order to add conditions, general data or units information to the problemtype. Moreover, a TCL file has to be introduced in the directory scripts containing the way in which the output information has to be written inside the input files that will be read by the solver. It is also possible to create new sophisticated windows, menus or new functionalities to the program inside the initialization file using the TCL/TK programming language. Table 1 shows the use of the main files in the new problemtypes system.

File extension	Description	Problemtype with Toolkit
name.prb	Problem and intervals data	Not used
name.cnd	Conditions definition	Used, should not be modified
name.mat	Material properties	Not used
name.bas	Information for data input file	Not used, should be void
name.tcl	Main TCL file, initialization	Used
name_default.spd	Main configuration file, XML-based	Used

Table 1: Main files that configure the new problemtypes with the Toolkit

## 3 WHY A NEW SYSTEM FOR PROBLEMTYPES CREATION?

There are various motivations for using the new Toolkit for problemtypes creation. On the one hand, XML is a W3C<sup>[3]</sup> standard, endorsed by software market leaders, it supports Unicode and it is becoming more and more popular in the area of storing and describing information<sup>[1],[4]</sup>. On the other hand, the syntax rules of XML are very simple, logical and easy to use. Moreover, XML documents are human-legible and reasonably clear. Besides,

using the Toolkit the information is stored in plain text format in a self-describing syntax, which facilitates the aggregation of data. It should be important to note that the elements in an XML document form a tree structure that starts at "the root" and branches to "the leaves" with different relationships between the properly nested elements. XML 'elements' are defined using not predefined "Opening and closing tags" and can have 'attributes', which provide additional information<sup>[3],[4],[5]</sup>. In fact, the tree structure of XML allows to aggregate efficiently elements and the new Toolkit permits to convert automatically XML documents to its physical tree view in the GiD window for interfaces creation. In this way, the Toolkit allows to hide parts of the data tree for specific types of analysis or to couple entities into a same 'group' using standard windows inside the interface. Moreover, the Toolkit permits to edit easily the 'group' properties. Furthermore, this tool is coupling all the common features of the different problemtypes, which facilitates the problemtypes creation.

## 4 TOOLKIT CONVENTIONS

Due to XML tags are not predefined, the Toolkit defines its own tags in order to create automatically a tree view and simple windows on the left of the GiD window, which facilitates the aggregation of data. In this way, we call 'field' to the 'Element name' and 'parameter' to the 'attribute'. Therefore, all data is contained in fields and parameters, where the parameters can contain a literal value, a xpath expression or a [TCL command]. It should be emphasized here that XPath is used for addressing parts of a XML tree. In fact, XPath is based on a tree representation of the XML document and provides the ability to navigate through its hierarchical structure selecting nodes from the tree<sup>[5]</sup>. In the case of considering a xpath expression, a search is performed in the tree and the result is substituted in the parameter when necessary. In the case of using a [TCL command], the command between brackets is executed when necessary and the return value is replaced inside the parameter. Typically, the TCL procs called 'proc' are defined at the end of the XML document. An realistic example of the Toolkit applications in order to create a tree view, interface windows and to simplify data storage is the Plaxis-GiD program, which is showed in Figure 1.

## 5 CONCLUSIONS

A new Toolkit has been developed for problemtypes creation in order to facilitate the introduction of all the input data to transfer to a specific analysis program. It is based on a XML hierarchical structure and a physical tree view on the GiD window. In order to add conditions, materials, general data or units systems to the problemtype, it is necessary to modify a XML tree inside the main configuration file, which contains all the data (except the geometry) necessary to perform a numerical analysis. The Toolkit permits to process easily XML documents on a physical tree view, but it also allows to hide parts of the data tree if this is convenient for a particular type of analysis. It also permits to couple

entities into groups using standard windows inside the interface and to edit their own properties easily. Summarizing, using the Toolkit it is possible to create tree interfaces easily and to store data more efficiently and automatically due to this tool is coupling all the common features of the different problemtypes. Thus, the Plaxis-GiD program is a realistic application of this powerful tool.

## 6 ACKNOWLEDGEMENTS

The authors wish to thank all those from Plaxis BV, who provide the project geometry and input dates given in Figure 1.

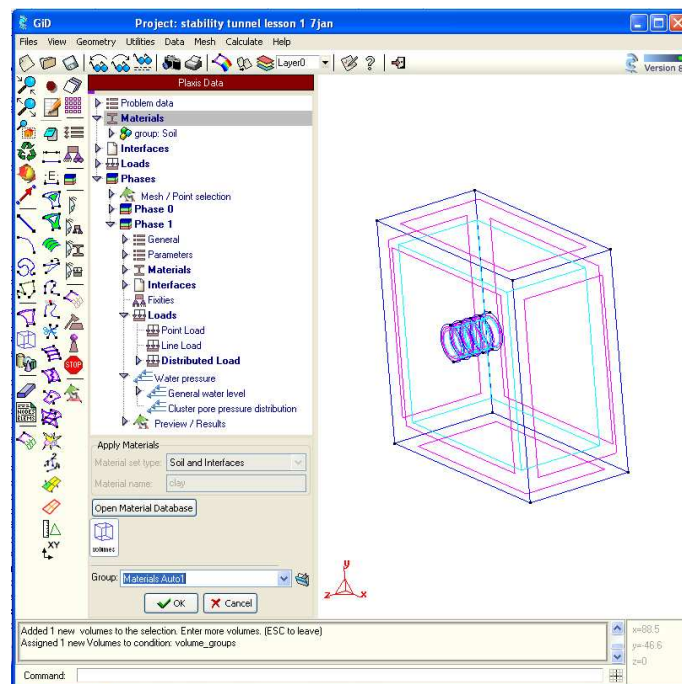


Figure 1: Problem data tree structure in the Plaxis-GiD interface program. A group is applied to volume entities using the Toolkit for windows creation

## REFERENCES

- [1] R. Light, *Presenting XML*, Sams. Net Publishing, (1997).
- [2] C. F. Goldfarb and P. Prescod *The XML Handbook*, Prentice Hall, (1998).
- [3] World Wide Web Consortium (W3C), available from [www.w3.org](http://www.w3.org).
- [4] M. Kay *XPath 2.0 Programmers's Reference*, Wiley Publishing, (2004).
- [5] A. Skonnard, M. Gudgin *Essential XML QuickReference*, Addisonj-Wesley, (2001)