ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: MECHANIKA z. 108 Nr kol. 1161

International Conference on
COMPUTER INTEGRATED MANUFACTURING
Internationale Konferenz über
RECHNERINTEGRIERTE FERTIGUNGSSYSTEME
Zakopane, March 24-27 1992

Marek KRZACZEK

The Institute of Technology and Building Materials
Technical University of Gdansk, Gdansk, Poland

NEW METHOD OF DATA EDITING FOR FE AND FD METHODS

Summary: In particular, this is the case when the FE numerical methods are concerned. This paper presents Graphic Data Editor (**GDE**) – a new method for data editing for 2D problems solved by the FE (FD) numerical methods. **GDE** makes it possible to input data in graphic form. It is also possible to make use of selected fragments of the technical drawing which describe the construction element under analysis. **GDE** recognizes the drawing, and generates a file containing numerical data which unequivocally describe the computational task. The problem of recognition and interpretation has been solved by means of the theory of fuzzy sets.

1. Introduction

This paper presents a new method for data editing, called Graphical Data Editor (**GDE**). It should be mentioned, that data editing is interpreted here as a method for generating, correcting and modifying the description of a computational task.

GDE enables defining a computational task by means of a natural language of graphic symbols, unequivocal and easy to understand by a designer. An additional advantage is the opportunity for creating a model of the analyzed construction element directly out of the technical drawing.

The **GDE** method can be employed to describe various engineering problems, and it can be used along with various computational programs. Originally, it has been developed to cooperate with the FE methods in 2D space.

2. Problem Formulation

GDE is a programming system providing the following functions:

- creating drawings (or importing drawings from a drawing editor) which are the graphical description of a task (geometry, boundary conditions, loadings),

- recognizing the drawing of a construction model (e.g. static structure in 2D member systems, or multiconnected regions in 2D plate statics problems) and its understanding,
- recognizing the logical description of working conditions of a construction element,
- checking whether the logical construction description is correct.
- generating a numerical description of the computational task,
- invoking a computational program.

The last four functions are realized by a Task Identification Module (TIM).

The basic assumption taken here is, that **GDE** should enable generating both the computational model of the construction based on a technical drawing, and the logical description of the construction with the help of a set of predefined graphic symbols supplied by **GDE** (e.g. symbols of loading types, boundary conditions, etc.). It should be as easy and natural to define a computational task as ordinary draft designing.

GDE analyzes by oneself the drawing describing the model of the geometry of the analyzed element, and working conditions. Such assumptions considerably complicate the problem. A technical drawing is essentially a set of graphic symbols which are unequivocal to a substantial degree, and which are drawn with precision adapted to man's perceptive abilities. This means that a technical drawing, although unequivocal and clear for a man, is not always a precise mathematical description of the construction (from a numerical method point of view).

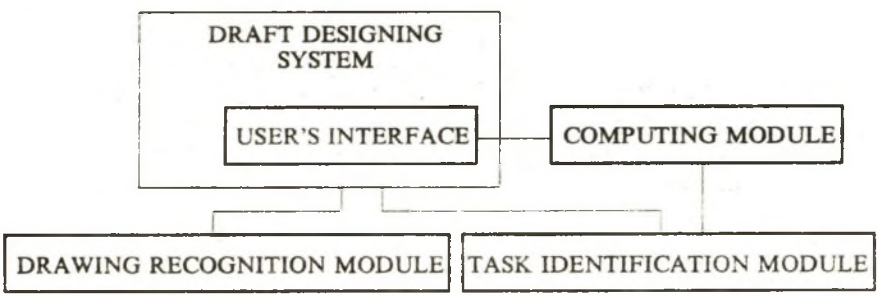


Fig. 1. General Architecture of GDE

GDE must generate an unequivocal and mathematically precise description of a computational task which must be clear for numerical programs. It seems quite natural that the problem should be solved based on procedures for description conversion, similar to image (technical drawing in this case) perception and recognition processes going on in human brain.

3. Drawing Recognition

A technical drawing made with the help of a drawing editor (DE) is not free from inaccuracies which are inessential from the point of view of technical documentation. Such a drawing is an unequivocal mathematical description of a real construction, and consequently, it is worthless. In the method for data editing suggested here, the technical drawing is recognized and interpreted by an artificial intelligence module **DRM** which has been developed based on an indeterministic algorithm for interpreting drawing inaccuracies and determining intersection points of graphic symbols. DRM imitates the drawing recognition processes going on in human brain.

From the DE point of view the graphic symbols make up a list of symbols, called the list of predefined objects O. A predefined object O_i is defined as follows:

Def. 1. Object O_i is a finite subset of cartesian product of a pair $\langle x_i, y_i \rangle$ in set theory meaning, on which the relation φ is imposed. It can be written:

$$\{\langle x,y \rangle : \varphi(x,y)\},$$

where: $\varphi(x,y)$ belongs to a family R of relations accepted in a particular solution.

The list of predefined objects O describes a drawing created with the help of DE.

The aim of the recognition is to recreate the drawing, to make it as consistent with the user's intentions as possible, and, at the same time, free from inaccuracies and equivocations. As a result, the recognition process creates a set of nonintersected objects, and points of intersection. The drawing inaccuracy results from certain psycho-physical features of the user and the quality of his computer equipment (e.g. graphical adapters).

To simplify further considerations, the list O is reconstructed, and thus a new list **Opf** of simple objects is created. The simple objects are defined as follows:

Def. 2. Object $\mathbf{Opf}_i$ is a subset of cartesian product $F(X \times Y)$, in the sense of set theory, where for each $x \in X$ there is one and only one y, such that $\langle x,y \rangle \in F(X \times Y)$.

To solve the problem of drawing recognition, the theory of fuzzy sets is used to generate the drawing description. To do this, fuzzy objects are introduced according to the following definition:

Def. 3. The simple object $\mathbf{Op}_i$ is a fuzzy relation **R**, which is a fuzzy subset of a Cartesian product of sets X_i and Y_i: $R \in F(Z_i \times Y_i)$.

To each subset X and Y a membership function is assigned:
$\mu_{xi}(x)$, for subset X_i
$\mu_{xi}(x)$, for subset Y_i

Further idea of drawing recognition is based on the assumption that the graphic objects included in the **Opf** list will be understood according to Def. 3. Consequently, a list of fuzzy objects can be generated out of the **Opf** list.

It is assumed, that if X and Y are spaces, X_i is a fuzzy set in X and Y_i is a fuzzy set in Y, then all fuzzy sets X_i and Y_i are normal, i.e.:

$$\sup_{x \in X} \mu_x(x) = 1, \quad \text{and} \quad \sup_{y \in Y} \mu_y(y) = 1.$$

To be able to interpret graphic objects by means of fuzzy sets, and to condition the interpretation by the degree of the drawing precision, the concept of α-cross-section of fuzzy sets X_i and Y_i is introduced as:

$$X_\alpha = [x \in X_i \mid \mu_x(x) \geqslant \alpha], \quad \alpha \in [0,1], \quad X_\alpha \in P(x), \tag{1}$$

$$Y_\alpha = [y \in Y_i \mid \mu_x(y) \geqslant \alpha], \quad \alpha \in [0,1], \quad Y_\alpha \in P(y), \tag{2}$$

From now on, fuzzy sets X_i and Y_i will be identified with their α-cross-sections.

Relation **R** is characterized by membership function $\mu_R(x,y)$. The value of the membership function determines how closely the elements of set X_i are related to the elements of set Y_i. Relation **R** is chosen individually for each type of simple object $\mathbf{Op}_i$, e.g. the following membership function of fuzzy relation $R_i(X \times Y)$ (for line segment) is assumed to be:

$$\mu_{RI} = \exp[-w \cdot k \cdot (y - c \cdot x - d)^2], \qquad \text{for } x_p < x \leqslant x_k$$

$$\mu_{RI} = \exp\{-k \cdot [(y - y_p)^2 + (x - x_p)^2]\}, \qquad \text{for } x \leqslant x_p \tag{3}$$

$$\mu_{RI} = \exp\{-k \cdot [(y - y_k)^2 + (x - x_k)^2]\}, \qquad \text{for } x_k < x$$

where: $w = \cos^2\beta$,
$k = f(\alpha)$, coefficient related to α-cross-section value.

As the next step in solving the problem, the intersection points of simple objects $\mathbf{Op}_i$ are determined and interpreted, and the **Opd** list of nonintersected objects is created. The process of drawing recognition presented here is similar to that going on in human brain (P.H. Lindsay and D.A. Norman, 1972), [4].

The determination of intersection points of simple objects results in the **Opd** list of nonintersected objects, such that:

$$R_i \cap R_j = \emptyset, \qquad \text{for } j \neq \begin{cases} i-1 \\ i+1 \end{cases}, \tag{4}$$

$$R_i \cap R_j = A, \qquad \text{for } j = \begin{cases} i-1 \\ i+1 \end{cases}, \tag{5}$$

where **A** is a fuzzy set, which is the common part of two fuzzy relations.

It can be shown that for two simple objects $\mathbf{Op}_i$ and $\mathbf{Op}_j$, the membershipfunction of fuzzy set **A** is given by the following formula:

$$\mu_A(x,y) = \min[\mu_{Ri}(x,y), \mu_{Rj}(x,y)]. \tag{6}$$

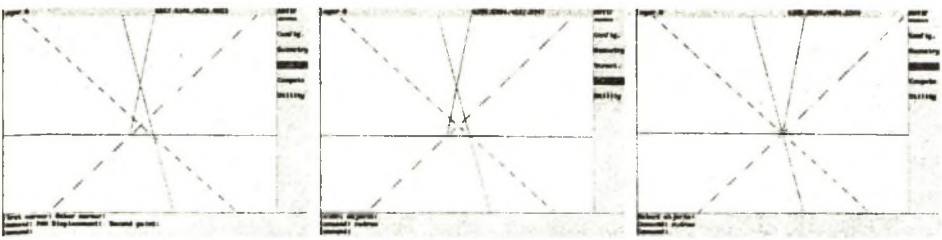
Taking into account the above assumptions and formulae (4) and (5), an intersection point can be defined as follows:

Def. 4. An intersection point of two simple objects is a pair (x_p, y_p), such that:

$$\mu_A(x_p, y_p) = \sup_{x,y \in A} [\mu_A(x,y)]. \tag{7}$$

The $\mu_A(x,y)$ membership function is continuous and not smooth. In a real computational algorithm a stochastic algorithm for determining local extremum of a not smooth function is applied. The mechanism of interpretation of intersection points of simple objects $\mathbf{Op}_i$ is presented in Fig. 2.

The **Opd** list is the result of drawing recognition, and is an unequivocal mathematical description of the drawing that is seen by the recognition system. The drawing recognition aims at realizing the intentions of the user who creates the drawing. How precise the drawing analysis is, and how much the drawing meets the user's expectations is determined by the values of α. It can be said that the values of α indicates how inaccurate the drawing is and determines its final form. This way the system learns and adapts itself to the specific features of its working environment (e.g. the level of precision of the technical drawing created by the user, or the quality of the equipment used to present graphically the drawing).



a) b) c)

Fig. 2. Mechanism of interpretation of Intersection Point of three Simple Objects.

Practice has proved that it is extremely essential to adopt the optimal strategy for choosing the order in which simple objects are analyzed. Therefore the strategy for analyzing the **Op**$_i$ objects can be modified depending on how far the analysis results are consistent with the user's intentions, and thus inreaching the knowledge possessed by the system.

The drawing after recognition still requires further analysis (e.g. additionally, in the case of static analysis of 2D plate systems, it is necessary to generate unequivocal description of multiconnected regions that would enable the FE mesh generator to carry out the In/Out tests). Depending on the engineering and computational problem, **DRM** can cooperate, for example, with a 2D modeler as a tool for generating a description of multiconnected regions.

4. Task Identification

The recognized drawing must be completed with description of boundary conditions, loadings, etc. The User's Interface enables to create such a description. Further, a task description should be identified and interpreted. In **GDE**, Task Identification Module (**TIM**) enables to do it.

The basic functions of **TIM** are as follows:
- logical identification of graphic symbols,
- geometrical localization of graphic symbols,
- test of the physical sense of the description of objects,
- transformation of the graphic description (base of facts) into a numerical description according to rules accessible in **TIM**.

The functions listed above have determined the architecture of **TIM**. It has been assumed that the most effective solution is a module based on expert system techniques [3]. The knowledge in **TIM** is represented in two ways:
- fundamental knowledge base consisting of rules of inference,
- dynamic base of facts (called hereafter context), generated by **UI** individually for every computational example, and modified by the Inference Engine during the inference process.

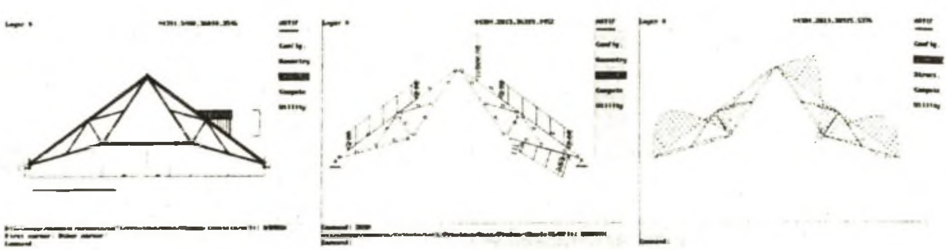
To achieve the optimal organization of the knowledge contained in facts, it is assumed that the context is composed of objects belonging to several classes in the sense of Object Oriented

Programming. The knowledge contained in the context is dynamically processed during the inference, but according to the rules of inference. Two following groups of rules can be distinguished in the base of inference rules:

- classical rules of the type: IF *smth* THEN *action*,
- fuzzy rules, where a specific action is taken after a fuzzy condition has been satisfied.

Fuzzy rules are the element of fuzzy knowledge. There is, however, a specific relation between fuzzy rules and the mechanism of drawing recognition. The level of confidence necessary to satisfy a fuzzy rule is specified by the values of fuzzy α-cross-section, understood and determined like for the **DRM** module.

To sum up, the inference amounts to processing the facts stored in the context according to the rules contained in the knowledge base. After the inference process has been completed, the context contains a full numerical description of the analyzed computational task. The results of inference depend on the knowledge consisting of facts, rules, values of α-cross-section and recognition strategy.



a) b) c)

Fig. 3. Computational task identification: a) technical drawing, b) task definition, c) results of calculation.

5. Concluding-Remarks

The Graphical Data Editor presented in this work is a new tool for editing data for the FE numerical methods and, after adaptation, for many other numerical methods (e.g. FD methods) of construction analysis.

GDE enables the user to generate the description of the analyzed problem in the form of a task drawing and a minimal number of numerical quantities supplementing the description. The geometry can be entered directly with the help of the graphic processor which cooperates with GDE, or it can be a selected part of the technical drawing.

GDE can adapt itself to the psycho-physical features of the user and the computer equipment used through learning. However, if differences occur during interpretation of successive drawings, it is possible to supplement the GDE knowledge. Therefore, GDE is a convenient tool that aids designing, gives engineer access to greater number of modern computational methods, and provides integrated designing environment.

REFERENCES

[1] Czogala, E., Pedrycz, W.: Elementy i metody teorii zbiorow rozmytych, PWN, Warszawa 1985.

[2] Hinton, E., Owen, D. R. J.: An introduction to Finite Element Computations, Pineridge Press Limited, Swansea, U.K., 1979.
[3] Krzaczek, M.: System expertowy generacji opisu zadania obliczeniowego zastosowany w systemie ARTIF, VIII Konferencja Metody i Środki Projektowania Wspomaganego Komputerowo, Warszawa 1991.
[4] Lindsay, P. H., Norman, D. A.: Human Information Processing. An Introduction to psychology, Academic Press Inc., New York 1972.
[5] Zadeh, L. A.: Fuzzy sets and systems, Proc. Symp. System Theory, Polytechn. Inst. of Brooklyn 1965.

DIE NEUE METHODE EINES GRAFISCHEN DATENVERARBEITUNGSSYSTEMS NACH EINEM FE (FD) -VERFAHREN

Zusammenfassung

Vorführung eines neuen grafischen Datenverarbeitungssystems (GDB) nach einem FE (FD)-Verfahren, in dem 2D-Aufgaben rechnerisch gelösst werden. Im GDB-Verfahren können grafische Daten in das System eingeführt werden. Es besteht auch die Möglichkeit, Fragmente technischer Zeichnungen in die Beschreibung des Berechnungsmodells einzusetzen. GDB erkennt die Zeichnung des Modells der Konstruktionsgeometrie und generiert eine Datensammlung (file) mit numerischer Beschreibung der Berechnungsaufgabe, welche eindeutig diese Aufgabe formuliert. Das Erkennungs- und Interpretierungsproblem wurde mit Hilfe der Theorie der unscharfen Mengen gelösst.

NOWA METODA EDYCJI DANYCH DLA METOD MES I MRS

Streszczenie

W pracy omówiono Graficzny Edytor Danych (GDE) – nową metodę edycji danych dla zagadnień 2D rowiązywanych za pomocą metod MES (MRS). Metoda GDE pozwala na graficzne wprowadzanie danych. Umożliwia również przenoszenie do opisu modelu obliczeniowego fragmentów rysunku technicznego. GDE rozpoznaje rysunek modelu geometrii konstrukcji i generuje plik danych zawierający opis numeryczny zadania obliczeniowego, który jednoznacznie opisuje zadanie obliczeniowe. Zagadnienie rozpoznawania zostało rozwiązane w oparciu o teorię zbiorów rozmytych.

Wpłynęło do redakcji w styczniu 1992 r. Recenzent: Ryszard Knosala