SOME ASPECTS OF COMPUTER AIDED DECISION MAKING FOR THE CRISIS MANAGEMENT OF UNSTABLE SLOPES

R.M. Faure, T.Pairault, M.Pham, A.Bernardeau-Moreau, G.Fayolle, J.C.Robinson & G.Foucheyrand
Ecole Nationale des Travaux Publics de l'Etat
Rue Maurice Audin
69518 Vaulx-en-Velin
France
Voice: +33-72 04 70 25
Fax: +33-72 04 62 54
E-Mail: faure@mails2.entpe.fr

KEYWORDS: crisis management, object oriented database, GIS, ground model.

ABSTRACT

We present here the developments in our risk management research; software tools based on object oriented techniques, an image and graphics based man-machine interface, a new algorithm which allows the quick construction of a GIS, strong links with analysis software with the possibility of using fuzzy logic reasoning. The case of the threatening landslide at Sechilienne (Isere, France), studied using these tools, is briefly presented. We will show that its management is facilitated through the use of networks as in the WASSS project approach (Faure *et al* 1995a).

RESUME

Nous presentons ici l'evolution de nos recherches en matiere de gestion du risque. Les outils informatiques mis au point s'appuyent sur les techniques orientée objets, les interface hommes machine comprenant l'image et le graphisme, un nouvel algorithme permettant de construire rapidement un SIG, les liaisons fortes avec des logiciels de calcul et des possibilités de raisonnement en logique floue. Le cas de Sechilienne traité avec ces outils est brievement presenté et nous montrons que sa gestion est facilitée par la mise en reseau dans la philosophie du projet WASSS (Faure *et al* 1995a).

INTRODUCTION

At the TIEMEC '93 symposium in Arlington we presented a paper (Faure *et al* 1993) which described the basis of an array of computer tools that improve risk management. This article is the follow-up to that paper, presenting the research and applications developed since to provide the crisis manager with quick easy to use tools.

The purpose of these tools is to provide the decision makers, as rapidly as possible, with the necessary decision making elements, in both surveillance and crisis period, while, at the same time, trying to evaluate the consequences of that decision. The computer hardware and software choices result in optimal product maintenance and portability, without forgetting the networks aspect, which is ever increasingly necessary for large area management.

AN OBJECT ORIENTED DATABASE FOR REASONING AND ANALOGICAL REASONING

In this section, the choices made for data management are explained. We have already described what kind of help an engineer should obtain with our crisis management system. The amount of data needed to properly identify the context of an impending landslide is very large and varied. In fact, the nature of information stored about an existing slope is both quantitative and qualitative: requiring simultaneous management of geotechnical, mechanical, hydraulic, sociological, historical and infrastructural data.

This data must be well organized, structured and classified in order to facilitate information searches and to answer an engineer's request. We will see here, that such information can be represented in several ways through *graphic user interfaces* (GUI). It would also be convenient if data extracted from a database did not require organization and restructuring in memory before presentation.

However, the system should be able to simulate disaster senarios in order to study crisis situations: computation and reasoning codes are applyed to some data to propose remedial works. Those codes place constraints on the organization of the data. Each program needs to retrieve the data in a particular fashion: for example, slope stability processing requires geotechnical and hydraulic characteristics. The reasoning component can require every type of information including results of data processing. The best way to take care of these ideas consists of structuring the data with complex objects. The notion of object models provides many concepts which are useful for our modular system; the principles of abstraction, classification, modularity and *point of view* can be easily implemented with object-oriented programming. Thinking of

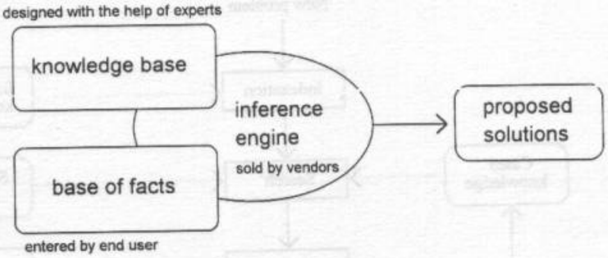

Fig.1: *A flowchart depicting the classic expert system.*

data as complex objects, or, in other words, object-oriented design allows to make the most of the expressive power of object-oriented programming languages. Also, It allows software components to be re-used and to increase system reliability against modifications (Booch 1992).

The whole system handles data as complex objects, especially at the database level. We decide to use an object database management system to create an object schema (Gardarin *et al* 1991) – which is a description of a specific object database, including classes, attributes, methods and links between classes – well adapted to the subsequent processing and also to to reasoning. For example, a slide stability computing program answers a request and retrieves from the objects database the desired objects to process. The use of a newly available *objects server* (ILOG 1994) ensures internal object model coherence (inter-object relations, derived attributes, origin and context functions) and external coherence. In addition, the object database manages simultaneous access to data, data integrity, security and persistency.

Since these objects model the situation of a slope, including geographical and mechanical information, they can be used as a base of facts for an expert system. Reasoning can thus be launched to deduce conclusions about the risk or about remedial solutions, with the empirical approach of an expert. The conclusion comes from a knowledge base applied to a base of facts (the application of the base of facts is performed by an inference engine), cf. Fig.1.

The object database also stores information about remedial solutions which have been adopted for past landslides. This data represents known cases of solved problems in slope stability. The object database has to keep useful information for an analogical reasoning, or more especially for a case-based reasoning. It consists of referring to the most similar existing and solved cases to find a solution to a new problem. Fig.2 presents a potential model to understand how this reasoning process works. This approach requires a large data base which the WASSS project attempts to provide.

As a first step, case-based reasoning can be used as a simple help for slope stability experts. The system just proposes judiciously repaired landslide cases and lets engineers think about the best solution they should adopt.

THE NEED FOR FUZZY LOGIC

One of the characteristics of the problems in soil mechanics is that the data is imprecise. The reasons for this are numerous. Firstly, the scale of the problems is far larger than the soil sample or human scale. The extent of the mass of soil can reach hundreds of metres. Secondly, soil, the material being studied, is not directly accessible and mainly hidden from human sight. Furthermore, knowledge of the great mass of soil is only punctual since it comes from a limited number of drill holes.

All this makes the knowledge of the geometry (the different layers) very imprecise, not to mention the geotechnical characteristics, as soil is highly heterogeneous. The question of groundwater is even more complex. Not only is the position of the water table difficult to determine, but the flow net varies with precipitation and the seasons.

The design of an expert system for soil mechanics has to take the imprecision into account. The problem is that most inference engines deal only with precise data. The vast majority of them consider symbols in their inferences. So in soil mechanics, the symbol will be vague, with lots of "small", "large", "medium", "high", etc.

The question is; what does "small" mean? This notion is not common to all experts, and it is not easily understood by those who use the expert system. And since "small" and other notions are not clearly defined, the reasoning (that is the rules) will be highly imprecise and, as consequence, useless. It is one of the reasons why it is apparently so difficult to design an expert system for soil mechanics.

Fuzzy logic has been introduced to solve the problem by quantifying the imprecision. Instead of being qualified by a symbol, a variable is known by a distribution of possibility. This can only be applied to numeric variables, but they are the ones that are affected by imprecision. For a variable, each value is given a possibility coefficient. If the coefficient is equal to

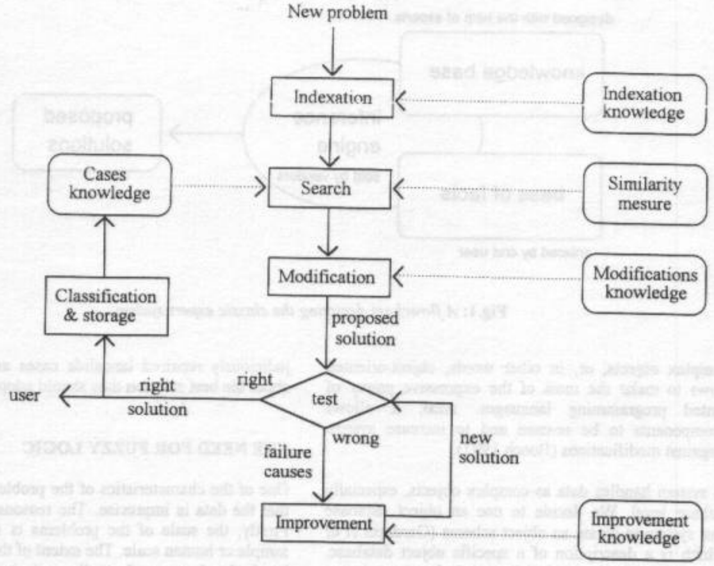

Fig.2: *A flowchart depicting the analogical reasoning process.*

one, it is absolutely possible the variable has this value. On the opposite, if the coefficient if equal to zero, it is absolutely impossible the variable has the value. Thus, each value gets a coefficient between zero and one. An infinite number of values can be absolutely possible or absolutely impossible (this is one of the great differences with the probabilities).

With this formulation, it is possible to represent the notion of "small" or "large" with precision, while they are imprecise. And they can be used to write rules. Roughly speaking, a fuzzy rule is something like:

"**if** variable1 is more or less between value1 and value2

then variable2 is more or less between value3 and value4"

(**Note:** this is only an example to help understand fuzzy logic, this ***is not*** fuzzy logic).

The real representation of variable1 being a distribution of possibility too, fuzzy inferences provide a mean to determine the distribution of possibility of variable2 given the distribution of possibility of the rule and of variable1.

With fuzzy logic, the design of an expert system in soil mechanics should be easier. As a part of the XPENT project, rules are being *fuzzified* to verify the feasibility of a fuzzy expert system for slope stability.

SPATIAL POSITION OF THE FACTS

It is necessary when building an Object Oriented Database for analogical reasoning on slide classification to have an easy to use interface for entering and consulting the data. It must provide an open structure so that the user entering his information doesn't have to consider their constituants. Of course, the less he has to do, the better and the faster it is. For this, we have choosen to provide a graphical interface based on the usual widgets (radio button, etc.) following international propositions (based on the World Landslide Inventory works (Cruden 1991)) on the way slides should be described. This allows the classical data common to each case to be entered quickly. For more specific information, such as images, maps, texts, illustrations, or everything else, we just make a link between the database and the file containing the information. When consulting the database, we suggest to associate a viewer for each type of non-usual object, which allows multi-platform use and independance of the information.

Our interface is also based on a network layer, which allows multiple users working at the same time on the database. The integrity problems have been solved by using pre-build products to manage the system (ILOG systems: Views & Server).

To structure geographical information, we have chosen to use a map as the background. This way, we get information on a region by clicking on it. We have at the same time the ability to access any slide presented on the map. When entering new information, we process it in the same way: firstly, we delimit with the mouse the region of interest, and then we have access to the entry interface. This mecanism is of benefit when considering interaction between different cases, because it facilitates the retrieval of geographical information.

LINKS WITH COMPUTATIONAL CODES

Computational codes are an indispensable component of the software tools required by the engineer. Such codes can perform quickly (using numerical tools such as the finite element method) otherwise laborious scientific calculations, thus furnishing the engineer with important information such as safety factors, pore pressures, etc.. In terms of risk management, they must be linked with the other tools in order to provide maximum efficiency. That is, they must be accessible via a user friendly interface that, in addition, allows access to the necessary data from a GIS, or an expert system, etc..

Even though computational codes are, in general, written to solve very specific problems, (e.g. the safety factor of a sliding slope or the pore pressures resulting from unsaturated flow as cited above) their output may be required as input by various other programs that perform completely different tasks. Consequently, the results can often be so important they can modify an entire project. Thus, it is necessary for every computational code to be validated by an expert in the field. Once the validation is performed, any change must be checked. Therefore, care must be taken to ensure that the results can be used by very different programs without having to adapt the code to individual needs. (Adaptation in this context means to change in a program that has been validated; a change may bring into question the integrity of the results.) Thus, once a computational code has been validated it must be separated from the rest of the programs, treating it as a completed module or "black box" and ensuring that it is not being changed constantly.

The modular approach to developing calculation codes also has the practical advantage that it facilitates development by a team. Each module is written independently, and consequently, where a team member disappears (as in colleges, when a student has finished his studies) the work is not lost, as all that is required by subsequent developers is the input and output format.

If the computational code is a separate module, the only point where attention should be focused is the interface: what to put in and how to retrieve results. From the design standpoint, the problem is to know how to transmit information. The answer depends on the software environment. Since that environment is evolving rapidly, the method has to change with it.

The solution that was decided upon at ENTPE was dialogue using files. Any program that needed a computational code created a file with the information the code needed (a slope, boundary conditions, etc.) then it launched the code. The code read data from the file (whose name was standard), computed and created a result file. At the end of the execution, the main program read the result file (with a standard name again). This solution worked perfectly well until Windows was used.

The exchange of file works if the main program knows when the code has finished. With Windows, this it is difficult to know because Windows is a pseudo-multitasking environment. Because we had decided to use programs that worked with files, we were forced to find a solution. However, currently, this approach within the Windows environment requires an advanced knowledge in Windows programming and is

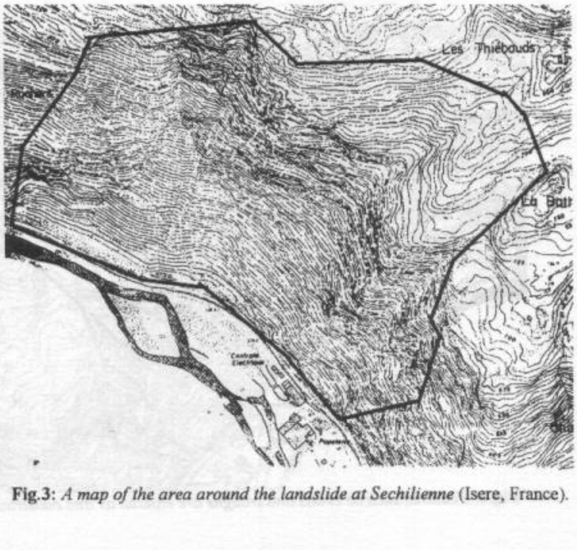

Fig.3: *A map of the area around the landslide at Sechilienne* (Isere, France).

consequently a stumbling block to heavy development.

In the future, the problem will be solved thanks the object technology. An object being an independent module, it satisfies all requirements. This is one of the reasons why all development at ENTPE is now done in C++. This object technology has been used in the design of Nixes & Trolls Windows, a program that computes the safety factor of slopes. All information is provided to the code by a GUI via objects. The same method has been used with PIR3D, a rockfall simulator (Faure *et al* 1995b).

RAPID GROUND MODELLING FOR SLOPE STABILITY ANALYSIS.

Like all information pertinent to an impending landslide, the geographical and topographical data is vast (Buisson *et al* 1993 & 1994). In the context of an object oriented expert system for landslide management such information is stored as a set of objects (e.g. buildings, roads and zones) placed punctually in space (i.e. **x,y,z** co-ordinates) but the actual surface may not be stored. Where it is stored it is either stored as:

1. A set of points on a regular grid,
2. A set of contours as read from a map.

The former, though possibly detailed (France has been covered by a 5m grid) reveals little about more complex features such as ridges, hill crests and troughs. The latter, though visually meaningful, is difficult to interpret automatically in computer codes. Added to this is the fact that both types of representation involve a lot of data.

For the engineer in the field, who wishes to perform a quick slope stability or groundwater flow calculation for a given site, the step from grid or map to the slope profile necessary for such calculations is vast. In the case of a crisis, the necessity for a quick method to achieve this step is of paramount importance. In this section we propose a quick method that avails of the facility of computer graphics interfaces and a new constrained mesh generation algorithm (Robinson 1995).

The rapid ground modelling (R.G.M.) method presented here is based on the use of line contoured maps. These maps can be stored in the computer as bitmaps (in the case of an object oriented database) or can be easily scanned into the system in the case of a crisis. Fig.3 shows such a map of the area around the landslide at Sechilienne (Isere, France) after scanning. It is used as the background for the graphics interface proposed here.

The thick black line represents the perimeter of a hypothetical area of interest. The important point to note at this stage is that the polygon has been entered with the mouse using visual criteria. (Although we sometimes tend to develop fully automatic systems, the human brain is sometimes the most efficient processor.) The **x,y** co-ordinates are generated automatically from the mouse cursor position and the elevation can be simultaneously typed in.

The next step is the innovation proposed here. The user visually discerns important features that describe the surface such as hill crests, ridges and troughs as shown in Fig.4. The idea here is that this process results in an adequate description (for slope stability calculations, etc.) of the surface with a minimum number of points. If these points were entered punctually there would be no guarantee that they would result in the representation of the ridges, etc., after standard triangulation. However, the use of a constrained mesh generation algorithm (Robinson 1995) ensures that the mesh conforms with the features. Fig.5 shows the mesh generated over the 81 points entered.

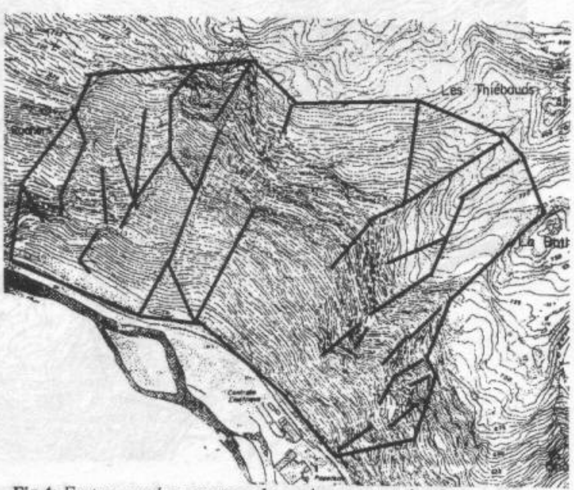

Fig.4: *Features such as crests and troughs are entered as line constraints.*

At this stage a ground model for the zone of interest has been generated. Fig.6 shows the **3D** representation of the surface (with shading from a vertical light source). The surface can now be "sliced" to provide a vertical profile in any direction as seen in Fig.7 & 8. Thus, the profiles needed as the geometrical data for many slope calculations are quickly and easily generated.

This system also facilitates the generation of surface parameter data necessary for 3D programs such as PIR3D or flood simulation tools. A zone defining a forest, grass or soft soil area can be entered using the same graphics interface as for the ground model polygon. Subsequently, any triangle in that zone is allocated the corresponding surface parameter value.

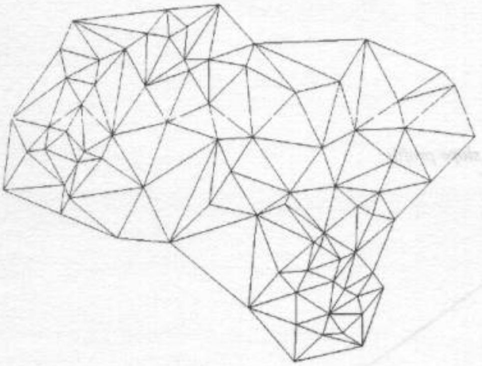

Fig.5: *A mesh that conforms to the features is easily generated.*

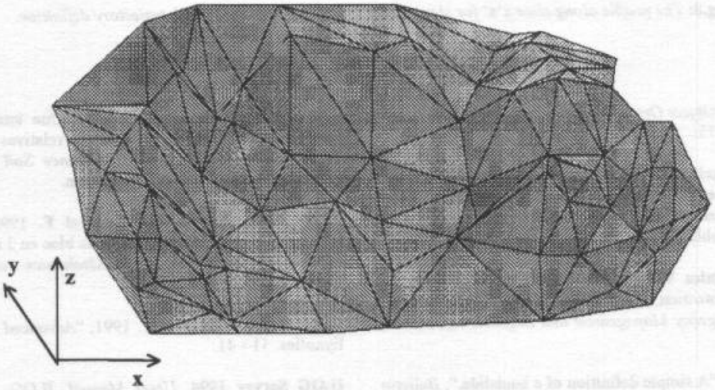

Fig.6: *A 3D image of the ground model.*

CONCLUSION

The real benefit of all these new risk management tools is clear if they are quickly and easily available to all. The solution to this is the use of computer networks. In the WASSS project (Faure *et al* 1995a) we are developing, using Internet, a wide array of such tools that will soon be used by other universities.

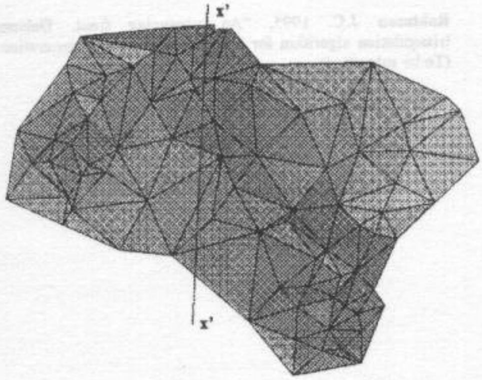

Fig.7: *The ground model can be sliced to produce a profile.*

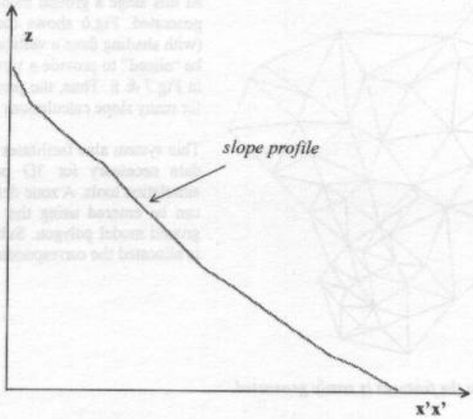

Fig.8: *The profile along slice* **x'x'** *for slope stability calclations or 2D block trajectory definition.*

REFERENCES

Booch G. 1992, "*Object Oriented design and Application*", Ed. Ad. Wesley: 22 - 75.

Buisson L., **Charlier C.** 1993, "Avalanche modelling and integration of expert knowledge in the ELSA system.", *Pierre Beghin International Workshop on rapid Gravitational Mass Movements*, Grenoble, France.

Buisson L., **Cligniez V.** 1994, "Spatial knowledge base for natural hazard protection, the ARSEN project." *International Congress of Emergency Management and Engineering*, Miami, USA.

Cruden D. 1991, "A simple definition of a landslide.", *Bulletin of the International Association of Engineering Geology*: 27 - 29.

Faure R-M., **Mascarelli D.** 1993, "XPENT, Slope stability expert system for managing the risk.", TIEMEC '93, Arlington, Virginia.

Faure R-M., **Pairault T.** 1995a, "Une interface conviviale pour le recueil de toute les données relatives à un glissement du terrain.", *XI European Conference Soil Mechanics and Foundation Engineering*, Copenhagen.

Faure R-M., **Fayolle G.**, **Tartivel F.** 1995b, "PIR3D, un logiciel de modelisation de chute de bloc en 3 dimensions.", *XI European Conference Soil Mechanics and Foundation Engineering, Copenhagen.*

Gardarin G., **Valduriez P**. 1991, "*Advanced Data Base*", Ed. Eynolles: 11 - 41.

ILOG Server 1994, *Users Manual*, ILOG: 2, av. Galliéni, BP85, 94253 GENTILLY FRANCE.

Robinson J.C. 1995, "An advancing front, Delaunay triangulation algorithm for constrained 2D mesh generation.", (To be submitted).