FUNCTIONAL APPROACH TO EMERGENCY RISK MINIMIZATION

Oleksandr Kuzomin, Illya Klymov

Abstract: An alternate approach to emergency risk minimization, based on lazy calculations and high-level functional interpretation is presented in this article. This approach is compared with classical imperative approach, based on statistical analysis on real data in order to prove its effectiveness.

Keywords: risk minimization, emergency, functional programming, lazy calculations

ACM Classification Keywords: H.1.1 Systems and Information Theory

Conference topic: Information Systems.

Introduction

Emergency risk minimization is a common task for geo-informational systems, which are implemented for everyday use in governments of many countries. A great number of different approaches were developed in order to provide an effective model for describing current situation, forecasting of possible emergencies and ways of reducing losses.

However, there are a number of challenges for such systems, which include, but not limited to:

- *Time factor* – usually an estimation of current state is needed urgently. That means that system is strictly limited in amount of possible calculation. The main challenge is to find optimal balance between complexity of developed system and precision of forecast.
- *Information factor* – in emergency it is a common situation, when obtained information could be incomplete or even contradictory. An intellectual approach is required for solving such conflicts.
- *Human factor* – experts situation estimation is an important information which is difficult to express in mathematical form in order to include into system estimation

A way to solve this problem greatly depends on type of emergency and can vary, depending on characteristics of geo-informational system, where such decision system is implemented. A challenging task is development of generalized self-adopting system, which can automatically configure its internal structure to fit current situation and requirements.

In other words the following ideas should be implemented in such system:

- Modular system of mathematical models – a framework, designed for describing peculiar properties of emergencies in region, their kind, type, and risk factors
- Map server – a common component of any geo-informational system to provide actual data of current relief, state of different objects on map (geo-information stations, rivers, dams, etc. – any object which can dramatically affect current state of situation)
- Decision adaptation system – tool, allowing quick adaptation of previous solutions to current situations, estimation of its effectiveness and quality.
- User interface subsystem – a way of visualization and obtaining information

Task formalization

Risk minimization system task can be expressed in a way, described in (1).

$$L(Sit, t) = \min F(X, R, A) \quad (1)$$

In this formula we introduce "losses function", which depends on current situation Sit, and period of time t for taking decision. Amount of losses depends on our "actions function" F describing our actions depending on current state of nature environment X, available resources R, and database of previous situations A.

Current state of nature environment is presented as superposition of quantitative parameters (such as speed of the wind, angle of slope, etc.) and qualitative (weather state, type of surface, etc.). Available resources are expressed in form of triplets *(Type, Amount, Availability)* – which accordingly describe type of resource, its amount. Third parameter (availability) is intended to describe a risk of resource loss during emergency. For example after snow avalanche an automobile roads could be destroyed rendering safety cars useless, and making helicopters the only way to access avalanche site.

Database of previous situations in this formula should be interpreted not only as storage of data, but also a system for access and providing compare function to search for nearest to current situation in database. That could be implemented as ontology of situations or knowledge base, gathering required information through data mining process.

In order to decrease complexity of solving task, following assumptions are made to minimization function:

1. Each situation in database could be compared with current situation, and a numeric value – "distance" between situation could be calculated. There are no situations of different types in database (in other words databases of solutions for different kinds of situations are separated);
2. Qualitative parameters of situation could be expresses as a value from limited subset. No human language descriptions are supported;
3. Available resources are sufficient for handling any kind of emergency – there are always at least one solution of function which leads to decreased losses, comparable to other variants, and so called "zero variant" – amount of losses taken in case of no action;
4. Human losses coefficient is supposed to have a priority over material losses. No further comparison between two solutions is performer if one leads to increased human losses;
5. No pre-calculations are performed. System is not using stored data in order for synthetic decrease of function calculation time;
6. Amount of time after taking decision is sufficient for any of actions, proposed by system;
7. Recommended actions will be performed according to system recommendation. Effectiveness of such actions will be estimated by experts in order to obtain quality coefficient of current solution (learning procedure).

Such assumptions allows decreasing available space of possible solutions almost two times, leading to no significant reduce in risk estimation effectiveness. The function result is a numerical value describing losses factor and actions which should be taken to obtain this result.

The quality coefficient is added to a system when action was taken and it is an integral value, describing experts' estimation of system effectiveness.

Functional Approach to Risk Calculation

In computer science, functional programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids state and mutable data. It emphasizes the application of functions, in contrast to the imperative programming style, which emphasizes changes in state. Functional programming has its roots in the lambda calculus, a formal system developed in the 1930s to investigate function definition, function application, and recursion. Many functional programming languages can be viewed as embellishments to the lambda calculus.

Functional approach in any situation handling interprets change of nature environment state (for example avalanche fall) as a result of function, depending on current parameters, and former value of this function [Hudak, 2006]. We interpret change of natural environment state as a result of low-level processes, which could be accumulated over time (2.)

$$R(fSit, t) = r(Sit) * R(fSit_{t-1}, t-1) \quad (2)$$

It is important to realize that $*$ in this formula should be interpreted not as common multiplication, but as an additional function taking two parameters, and producing a set of numerical value, which is expressed as risk of emergency, and internal data not visible to user describing previous states of function.

$fSit$ - is a high-level function, describing current situation change. This function is a key element of risk estimation and it is composed under following rules:

1. Each parameter of nature environment is integrated in function one or more times;
2. Each qualitative parameter is integrated into function only combined with other qualitative parameters or as an argument in predicate form;
3. Each part of function could be evaluated independently and do not affect evaluation of other parts.

A source of creating such function could be different. In this paper for testing purposes a simple predicate system was created, based on automated analysis of statistical data. Any kind of low-level functions can be used, including data, obtained from regression and correlation analysis, experts' opinion, etc. This data are integrated into high-level function in such way they form a weighted graph for risk estimation. The functioning of this high-level function greatly depends on weight of this graph. In order to improve performance of system different parts of it are evaluated simultaneously, with no guarantee of computational order (this is safe because of rule 3). Irrelevant points of graph would not be evaluated; also in case of conflicting data a point with fewer sums of weights attached to it is eliminated.

Lazy calculations allow an effective usage of conflicting data. For example if we have a conflicting node in our graph, which describes a snowfall of grade 3, which is marked in conflict with node "snowfall – grade 5", and an evaluation functions takes a predicate "if it was snowing?" as argument – weights of this nodes will be summarized in final calculations since there is no conflict in appearance of snow and conflict evaluation would not be performed.

Additional data and functions could easily expand working model on-the-fly without any requirements of previous data recalculation.

Testing technique

In order to test effectiveness of proposed system a subset of mudflows in Carpathian region in period from June 2006 up to July 2008 was analyzed. Nature environment data are obtained both from geo-informational stations logs from that period, official meteorological data and records of commissions worked on that mudflows elimination. Due to lack of information about available resources, the primary aim of a check was to compare existing methods of mudflows analysis (implemented in GIS system of Ukraine Ministry of emergency situations) and proposed functional method in terms of time, required for prognosis and its effectiveness.

A limited set of previous situations (about 1000) were passed to a system in order to obtain elementary predicate rules for possible mudflow. These rules were integrated into system with equal coefficients and a learning process was performed. A learning was performed on different sets of previous situations (from 100 up to 1000) in order to show the effectiveness of self-refining functional system.

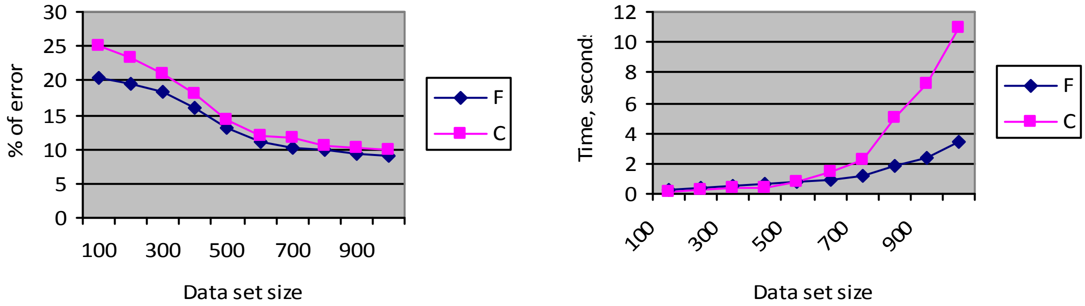

Fig. 1. Test results

On the first graph (see Fig. 1) we can see a percentage of error prognosis for implemented in governmental GIS system algorithm (marked “C” in graph) and our functional system (marked “F”). As it seen from the graph, the precision of prognosis on big sets (more than 500 situations) are almost equal. The situation with time of prognosis is significantly different. We can see almost a clear exponential grow for “C” algorithm, when grow of proposed functional algorithm after initial complexity change is almost linear, as expected.

According to statistical data [Ishchuk, 2002] an average amount of situations taken into account is about 5000 and process of taking decision usually takes up to 10-12 minutes, depending on amount of available data. Implementation of functional approach allows to significantly decreasing this value without loss of prognosis precision.

Modeling of decision taken was performed on snow avalanche data in Kirgizia during the period 2003-2009. Due to small amount of real avalanches in this period it is not possible to build a mathematical model to check system effectiveness. The main task of the system was proposal of resources and actions for avalanche controlling. A system solutions were effective (have an error less than 10%, compared with real situation) in 89% of situations.

However, testing revealed several problems, which should be solved before real system implementation:

1. A problem converting data from different sensors to one numerical format, suitable for processing with high-level functions;
2. Problem of building an effective high-level functions, describing complex emergency processes, for example earthquakes and water flows – system is quite effective analyzing big sets of similar data, but it’s speed decrease greatly on growing complexity of mathematical model
3. An effective algorithm for checking high-level functional models should be developed in order to test different function variations against real data.

Further improvements of an algorithm should allow benefiting from already clustered data about situations and providing a way of obtaining generalized rules for similar emergency in different regions all over the world.

Conclusion

The experimental check demonstrates that functional approach to emergency risk minimization is an effective solution for taking decision in limited time. The speed of proposed algorithm combined with lazy calculations, which allows greatly decrease time for making model less complex, allows performing just-in-time estimations of current nature state in order to perform a correct prognosis. However, the main disadvantages of this approach are problems of implementation and creation of basis functions in order to fully describe situation development. Creation of such mathematical tool should allow rapid increase in complexity of modern risk estimation systems.

Acknowledgements

The paper is partially financed by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA and the Consortium FOI Bulgaria (www.ithea.org, www.foibg.com).

Bibliography

[Hudak, 2006] Conception, evolution, and application of functional programming languages. ACM Computing Surveys 21 (3): 359–411, 2006

[Ishchuk, 2002] Іщук О.О., Ободовський О.Г., Коноваленко О.С. Взаємодія ГІС та проблемно-орієнтованих моделюючих комплексів в системах прогнозування та оцінки наслідків надзвичайних ситуацій, пов'язаних з паводками // Науковий збірник КГУ «Гідрологія, гідрохімія, гідроекологія», 3т, 2002 р. –С.53- 59.

Authors' Information

Kuzomin Oleksandr *– Professor, Ph.D., Chief of information marketing department. Kharkiv National University of Radio Electronics, Ukraine, Kharkiv, Lenina pr. 14, e-mail: kuzy@kture.kharkov.ua*

Illya Klymov *– Postgraduate (informatics chair), Kharkiv National University of Radio Electronics, Ukraine, Kharkiv, s. Mir 118, ap. 55, e-mail: illya.klymov@gmail.com*