COMMUNICATION EMERGENCY DECISION SUPPORT SYSTEMS INTEGRATION*

Vladimir B. Britkov

Institute for Systems Analysis, Russian Academy of Sciences
9, prospect 60-let Octyabria, 117312, Moscow, Russia,
Voice: 7-095-1355541
Fax: 7-095+9382209
E-mail: britkov@wlab.msk.su

ABSTRACT

This paper presents a framework for systems integration of communication systems in emergency decision support activity. There are some achievments in emergency management engineering. International society has developed many high level emergency decision support systems. An analysis of those systems clearly demonstrates that all of them have some advanteges and disadvanteges. The task is to integrate all advanteges of every system. In this case we have necessary tools of making qualitative decisions during emergency situations, when dealing withcomplicated problems with large amounts of information and many active agents, who influence on the consiqency of events.

KEYWORDS: systems analysis, emergency situations, system integration, communications systems, networks, decision support systems

---

*The research of this paper was carried out under support of the Russian Foundation of Basic Researches (Grants N 96-01-00740 and N 97-01-00168)

1. INTRODUCTION

There are some achievments in emergency management engineering now. International society has developed many high level emergency decision support systems.

Some of them were developed by international teams: Muster [2], MEMbrain [11], RODOS [13]. There are Russian systems: SPRINT [19], Archimedes [9], Typhoon, etc. An analysis of those systems clearly demonstrates, that all of them have some advantages and disadvantages. It is not possible to develop the best system, which can solve all problems the best way. As a rule emergency management engineering is a multi-disciplinary task. The problem is to carry out the approach of communication emergency engineering systems. In this case we can use the best feature of every heterogeneous decision support system (Figure 1).

2. COMMUNICATION EMERGENCY ENGINEERING SYSTEMS

There are some attempts to develop the general approach for systems integration. It is very interest problem, but it is very difficult. We think that it is impossible to create to universal decision of this problem in a short time. We try to achieve some results in more narrow field - Emergency Management Systems. There is currently a lot of activity in the area of integration frameworks and system architectures. Key issue of this direction is the date interchange protocol and systems communication technique. There are some achievements in this field in computer communication networks [7-9].

The systems approach to decision-making process is the necessary condition for successful decision in emergency situation [4,6]. To work out decision support systems (DSS) effectively we should develop communication approach to decision making technology.

System integration methods and communication approach allow to solve the knowledge base decision support system development, but system integration is a complex problem-area where solution approaches often need to make compromises for conflicting requirements [2]:

heterogeneity versus homogeneity;
autonomy versus common policy;
distribution versus centralization;
efficiency versus flexibility and extensibility;
simplicity versus support a large variety of requirements;
variety of components versus functional usability;
syntactical basis versus semantical integration.

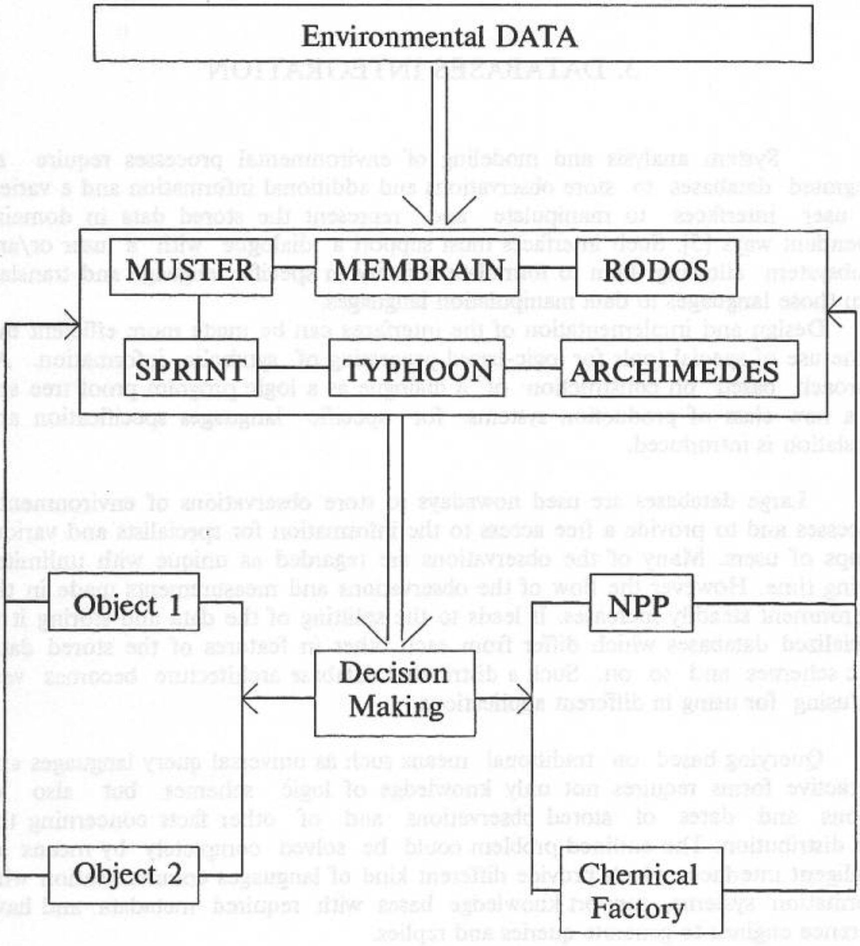

Figure 1. Emergency Management Communication Systems Archetecture

We can underline the main problems in system integration:

1) Data integration:
2) Control integration

On the first stage of this problem it is possible to consider only information part of such task - databases integration.

3. DATABASES INTEGRATION

System analysis and modeling of environmental processes require an integrated databases to store observations and additional information and a variety of user interfaces to manipulate and represent the stored data in domain-dependent ways [5]. Such interfaces must support a dialogue with a user or/and a subsystem allowing them to formulate a query in specific language and translate from those languages to data manipulation languages.

Design and implementation of the interfaces can be made more efficient due to the use of special tools for logic-based processing of symbolic information. An approach based on construction of a dialogue as a logic program proof tree and on a new class of production systems for specific languages specification and translation is introduced.

Large databases are used nowadays to store observations of environmental processes and to provide a free access to the information for specialists and various groups of users. Many of the observations are regarded as unique with unlimited storing time. However the flow of the observations and measurements made in the environment steadily increases. It leads to the splitting of the data and storing it in specialized databases which differ from each other in features of the stored data, logic schemes and so on. Such a distributed database architecture becomes very confusing for using in different applications.

Querying based on traditional means such as universal query languages and interactive forms requires not only knowledge of logic schemes but also of regions and dates of stored observations and of other facts concerning the data distribution. The outlined problem could be solved completely by means of intelligent interfaces which provide different kind of languages communication with information systems, support knowledge bases with required metadata and have inference engines to generate queries and replies.

However the latest researches in Artificial Intelligence applications show that natural language understanding involves a complex of problems for which satisfactory solutions are not yet found [15]. A full-scale implementation of an intelligent interface at a single-processor computer, now in common use, is high-cost and resource consuming. The future progress in AI applications and computer technique will change the situation, but today a designer has to look for a fair balance of intelligent features and productivity. Assuming this approach we face a major problem of interface design for information systems. Construction of procedures which translate the formal queries into data manipulation languages. The translation is determined by database architecture and semantics of specific professional language. To solve these problem we propose an approach based on logic programming and a special class of production systems. Tools being developed within the approach provide on-line modification of contents and construction of effective procedures for translation of queries into data manipulation languages.

Queries are represented as frames [14]. Their representation serves in the form of an AND/OR-tree. The root of the tree is an OR-node and it has AND-nodes as descendants. Lower levels consisting of OR-nodes and AND-nodes alternate. A descendant of an OR-node can be only an AND-node. A descendant of an AND-node can be an OR-node or a terminal node. Values entered at the terminal nodes are translated from specific language into strings in a formal query language. At the AND-nodes the generated strings are concatenated and passed to the parent OR-nodes. At these OR-nodes the text is translated into the formal language and then is passed to the ancestors. Finally a formal query is generated at the root of the dialogue tree. The text of the query is determined by a conceptual data model chosen by the designer of the information system, and does not depend on user's views and the database logic scheme. The formal query is then translated into a sequence of commands in a data manipulation language, for example, in SQL.

Such PROLOG programs contains a dialogue tree description which is a set of named nodes with lists of their descendants. Each node is linked to a text displayed when a user reaches the node during the dialogue. If it is an OR-node the text is a list of descendants with explanations about alternatives. If it is an AND-node the text includes lines which are the names of the descendants. Later when a descendant passes a string to the parent its name is replaced with the string. A terminal node is linked to a help message or to an instruction saying how to enter a corresponding value. The predicates defining actions at the nodes form a nucleus module. The actions are described above and do not depend on a dialogue tree. Passing from a node to its descendant is implemented as a subgoal selection. passing in the reversed direction is made on backtracking. The unification mechanism passes data from nodes to their ancestors.

The dialogue logic programs call procedures which translate input into a designer chosen formal language. At the end of the procedure is called which translates formal queries into a data manipulation language. Effective tools were developed to create the translation procedures. The tools are based on a new class of production systems for symbolic data processing according to logic rules [1]. The production systems are built of so called L-productions which generalize Post canonical systems [16]. A premises of an L-production may include typed variables. The conclusion of such a production may include functions over symbolic strings.

Types of variables are defined as recursive unary predicates over strings and functions are defined as strings transformation procedures. In the definitions L-production systems can be used recursively. L-productions are used in calculeses and algorithmic systems which describe string sets transformations. A PROLOG version of L-production systems interpreter has been developed and implemented [5]. It enriches significantly the means for creation of logic programs for texts analysis and generation and permits to use joint methods of logic and production programming in development of intelligent front-ends for information systems.

4. SYSTEMS INTEGRATION

Systems integration is more complicated task that data integration. There are some approaches and methodologies in this area. First of all we need develop task model, which is system independent. A task model is system independent if it can be transported from system to system while remaining sufficiently accurate. This property of a task model is particularly important in heterogeneous computer systems where it is likely that software components are moved from one type of hardware to another. The parameters of task models are usually system-dependent as well. Separation of concerns is necessary in order to isolate the system-dependent features from the system-independent features. A task may be modeled at various levels corresponding to the levels at which a computer system may be described [12]:

Physical level. The task model is based on the consumptions and consumption rates of the system's hardware devices and software resources.

Virtual level. The task model is based on consumptions and consumption rates of the system's logical resources such as high-level language statements, logical disk accesses, and demand for virtual memory.

Functional level. The task model is based on rates of invocation of functions at the end-user level. This level of task modeling requires consideration of operations provided by the software.

Task models at the physical level are normally system-dependent whereas models at the virtual or functional level potentially are system-independent. To obtain task characterizations that are independent of the speed of hardware devices, the hardware demands should be found in terms of number of invocations of hardware device operations. Then, hardware demands can easily be calculated for similar hardware devices with different speeds.

System independence may be achieved by means of a layered model of execution of software. The software as described in terms of programming language statements and programming language statements are given hardware-dependent characterizations in terms of machine instructions. This division provides flexibility in work modeling when the software is re-compiled by using a different compiler, when the software is moved to a faster CPU of the same type as before, or when the software is moved to another kind of CPU.

The suggested approach is based on the principle like client-server one. It is built of subsystems, which are connected via a communication interface. Each of these subsystems can either be a server, which provides special procedure to other subsystem, or a client, which requests services from other subsystem, or both.

5. CONCLUSION

In this paper we have presented an new approach, which united different areas of modern information technology: systems analysis, DSS, artificial intelligence, network communications. There are great success in these areas separately. But there are a few examples of achievements in the implementation all of them simultaneously. We are making a first steps in this multidimensional space.

REFERENCES

1. Aleyev V.R. Construction of Linguistic Models. in: "System modeling and informatics". Moscow, Institute for Systems Studies of Academy of Sciences, 1986.

2. Andersen V., Andersen Henning B. Multi-User System for Training and Evaluation of Environmental Emergency Management Response - MUSTER. In.: The International Emergency Management and Engineering Conference, 1995. Nice, France.

3. Berre, A.J. Requirements to ESF Software Bus. Technical report, SI, ESF-Eureka Software Factory, 1989.

4. Britkov, V.B., Gelovani, V.A. and Yurchenko V.V. An Interactive Modelling System for Analysis of Alternative Decisions.- In: Decision Support Systems: Issues and Challenges. IIASA Proceedings, no. 11, 1980.

5. Britkov V.B., Aleev V.R. An intelligent user interface for an enviromental information system. In: Scandinavian Conference on Artifical Intelligence. Proceedings of the SCAI'88. IOS Amsterdam. Springfield, VA, 1988.

6. Britkov, V.B. Decision support information system development. In: Multiobjective Problems of Mathematical Programming. Lecture Notes in Economics and Mathematical Systems. Vol. 351. 1991, Springer-Verlag.

7. Britkov, V.B. and Vyazilov, E.D. Expert Systems Implementation for Oceanographic Data Using During Decision Making. IX International Conference Automation of the World Ocean Research and Exploitation Technical Means Control Processes. 1994, St.Petersburg.

8. Britkov, V.B. International Knowledge Based System for Emergency Decision Support. In.: The International Emergency Management and Engineering Conference 1995. Nice, France.

9. Britkov, V.B. Distributed System for Emergency Decision Support with Systems Analysis and Modern Information Tecnnology Implementation. In.: The International Emergency Management and Engineering Conference. Ed. Jean Luc Wybo, Marie-Christine Therrien, Frank Guanieri, Montreal, Canada, 1996.

10. Chen, P.P. The Entity-Relationship Model - Toward a Unified View of Data, ACM Transactions on Database Systems no.1, 1976.

11. Drager K.H. MEMBRAIN - A Modular Emergency Management System. In.: The International Emergency Management and Engineering Conference. Ed. Jean Luc Wybo, Marie-Christine Therrien, Frank Guanieri, Montreal, 1996, Canada.

12. Ferrari, D., Serrazi, G., and Zeigner, A. Measurement and Tuning of Computer Systems. Prentice-Hall, Englewood Cliffs, N.J. 1983.

13. Gutzwiller, Th.A., Lehmann-Kahler, M.G.C. and Oesterle, H. computer Based Strategic Information Planning Architecture. In.: Information Technology for Organizational Systems. H.J.Bullinger et al. (Editors). Elsevier Science Publishers B.V. (North-Holland), 1988.

14. Minsky M. A Framework for Representing Knowledge. MIT, Cambridge, 1974.

15. Natural Language Understanding and Logic Programming./ edts. V.Dahl, P.Saint-Dizier. North-Holland, 1985.

16. Post E. Formal reduction of the general combinatorial decision problem. "American Journal of Mathematics", 1943, v.65, No. 2.

17. Sauer-Pasler J. et al. "RODOS, a Real-Time On-Line Decision Support System for Nuclear Emergency Management in Europe". In.: Oslo Conference on International Aspects of Emergency Management and Environmental Technology, Oslo, Norway, 1995, Ed.: K.Harald Drager.

18. Sernadas A. Information Systems: The THIRD WAVE. In: Information Processing-86. Proceedings of the IFIP 10th World Computer Congress. H.-J.Kugler(ed). North Holland, IFIP, 1986.

19. Vyazilov, E.D. Britkov, V.B. Loading the Decision Support System with Knowledge for Emergency Management under Natural Disasters. In.: The International Emergency Management and Engineering Conference 1995. Nice, France.