Market Oriented Integration of MS-Windows-Based Tools for Distributed Decision Support*

M. Biró† P. Danyi‡ P. Gelléri‡

Abstract

In this paper, we discuss the meaningfulness of value added systems integration for distributed decision support from a market oriented primary perspective. The issues to be analysed are derived from all pairwise interrelationships of the entities involved in a decision situation. These are the task logic, the decision culture, and the decision environment. Keeping these considerations in focus, we summarize experiments with commercially available products for the Microsoft Windows environment which is undisputably the most popular operating environment for personal computers.

Keywords: Systems integration, Decision support systems, Model design, Distributed decisions.

1 Introduction

The purpose of this paper is to give a structured guide to the design of distributed decision support systems. Since our primary objective is the supply of the market, we are concentrating on Microsoft Windows based tools which can be used on the most popular type of personal computers worldwide.

Our approach is derived from practical experiences in building and installing decision support systems to orders. One of our recent observations is that users are less keen on accepting a clever but custom made software tool than well established commercial products. We also see however, that commercial products alone are most of the time inappropriate for the support of specific decision circumstances. Our answer is *value added systems integration*.

The universal validity of our conclusion is supported by the report of a coloquium held by the U.S. Computer Science and Telecommunications Board, the Commission on Physical Sciences, Mathematics, and Applications, and the National Research Council in 1991. There, "systems integration was identified as a

*Supported by OTKA grants No. 2571 and 2575.

†Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Kende u. 13-17. H-1111 Hungary

‡Dept. of Information Engineering, Technical University of Budapest, Budapest, Műegyetem rkp. 3. H-1111 Hungary

large and rapidly growing market in which the United States was a clear leader" [1] [Keeping the U.S. Computer Industry Competitive... 1992].

In this paper, we are not going into the details of systems integration issues in general. We are rather concentrating on the structuring of ideas based on our practical experiences in building and installing decision support systems and our pioneering role in introducing object- oriented windows based software and decision support technology in Hungary [2], [3].

2 A Model for Mapping Decision Situations

A DSS must always refer to the particular decision situation. However, decision situations are not only determined by the decision problem itself, but also by the problem owner and the available decision techniques. Let us formulate a model which, according to our experiences, provides an appropriate guidance for our analysis (Figure 1.). A DSS stands in the intersection of the entities of the basic model which means that a DSS can only be built if we bring together the contexts of these entities.

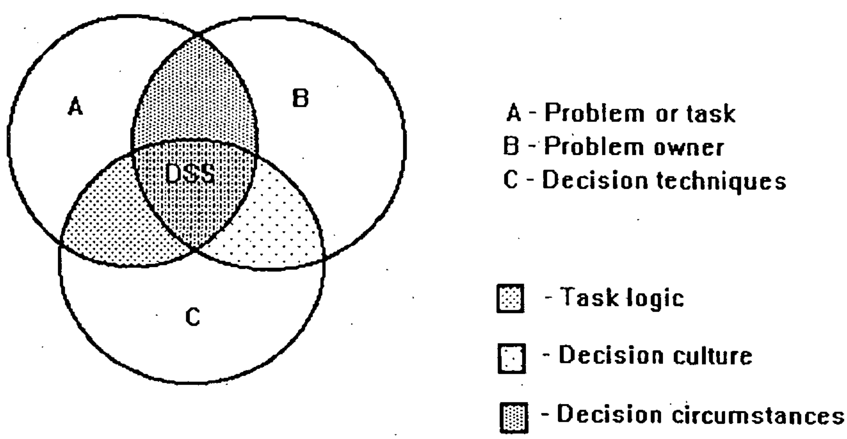

Figure 1.

Examples could be brought from an infinitely wide range of areas including money allocation, tender evaluation, personnel selection. Let us consider the following specific example. A DSS is being designed for managing *catastrophe situations* in a power plant. The system must not only contain decision techniques in themselves, as e.g. MCDM, fuzzy logic or AHP. Decision models should be set up concerning

- different kinds of problem (or task) situations, e.g. earthquake, computer virus, etc...
- problem owners with different levels of decision authority ranging from a guard to the president.

The entities do never occur apart but in a colorful amalgamation which we are interested in. Let us consider the intersections of all pairs of entities:

Task logic. The intersection of decision technique and problem (or task) is related to the abstract *types of decision problems* which reflects different decision models and have logically different solution algorithms. The most typical task logics are as follows: selection among discrete (well defined) alternatives, task monitoring, resource allocation, etc. It is obvious that any DSS supports some of the possible task logics but not all of the logics. Different DSSs must be built for the catastrophe example in the different warning phases with dissimilar levels of danger.

Decision culture. The intersection of problem owner and decision technique is related to the decision culture. This means the problem owner's experience in solving decision problems that is capability of using various kinds of decision methods and tools, and his skill level at using them. More specifically, e.g. some people prefer using probabilities, others do odds or utilities. On the other hand, some people are risk-averse, some are risk-prone. Japanese and American managers hardly ever look similarly at the very same problem. What kind of presumptions can we have about cultures? First, some people may be homogeneous as far as their decision thinking is concerned. Second, if they think differently, classes must be defined. Our goal is to help the problem owner in finding his real role i.e. his class.

Decision circumstances. Finally, let us consider the concept of decision circumstances, which is related to the intersection of problem and problem owner in the model. First, decision circumstances include the constraints and goals of the problem owner together with his or her attitude to the task. This also means time and resource constraints, and considerations coming from personal interests on the other hand. Second, the environment of the given problem has a huge influence on the design of the DSS. Some of the important issues are the individual or group nature of the decision environment, the chance for a compromise in the group case, the equality or inequality of voting powers, etc.

3 The MS-WINDOWS Based Toolkit Approach

In our opinion the most effective way of building a DSS satisfying particular requirements is using a toolkit. When we build a DSS from parts, we can excellently track the needs of the user, the environment, etc. In addition, the toolkit approach provides some technical advantages:

- modularity
- all of the pieces are exchangeable
- interfaces between units must be precisely elaborated.

The tools that we shall inspect are commercially available products for the Microsoft Windows operating environment. The interface between the units is naturally provided by DDE (Dynamic Data Exchange) and OLE (Object Linking and Embedding) which are defined in general within the environment.

However, while the above features significantly facilitate systems integration, we have to extend the commercial tools with new capabilities in order to supporting specific decision circumstances. These extensions will be highlighted below as well.

Tools covering significant task logics

The tools that can be mentioned here must include at least group scheduling capabilities which are necessary for monitoring the group decision making process and for allocating the necessary resources. There are many Windows based products belonging to this category. One of them is Schedule+ included with Windows for Workgroups and the future Windows NT as well.

Windows for workgroups has another important characteristic from the task logic point of view, which differentiates it from other groupware tools like Lotus Notes available today. It supports peer-to-peer networking with network dynamic data exchange facility as opposed to the client- server paradigm inherent to other tools. This feature opens new possibilities for distributed decision support where each personal computer on the network can operate both as a client and a server, obviating the need for a dedicated server. These networks are not only inexpensive but also easy to set up. A useful exploitation of this technology for distributed negotiation support (DINE) is described in [4]. This application was based on a prototype network dynamic data exchange facility developed with the participation of one of the authors one year before the release of the commercial Microsoft tool.

Tools covering significant decision cultures

Experts participating in a distributed decision making process may have different professional backgrounds which basically determine their decision culture. Different professional backgrounds implie that their professional cognitive patterns are different as well. A tool supporting distributed decision making must provide support for each individual expert and for the group as a whole. Thus, the model representations offerred by the system must be appealing to all of the participants, which implies that they must be as close as possible to everyday cognitive patterns. Tabular (relational) representations in *spreadsheets* for example satisfy this requirement, since tables are incorporated among our cognitive patterns at the elementary school level. This is in fact the fundamental reason of their general success [5].

Spreadsheet products for Windows are numerous again. They include Lotus 1-2-3, Borland Quattro, and Microsoft Excel.

The already mentioned application (DINE) [4], [6] is based on Microsoft Excel, which was extended with several features in order to accomodating experts from various professional backgrounds and still providing a high level of decision support. These features include optimisation and multiple criteria decision making capabilities in an environment where dynamically changing data originating from shared data bases or other members of the decision making group are permanently taken into account.

Tools covering significant decision circumstances

Groupsystems and Lotus Notes are commercial tools that are relevent to different decision circumstances. Groupsystems provides anonymous, real-time interaction with the help of a facilitator in an electronic meeting room. Lotus Notes provides workgroup electronic mail, distributed databases, bulletin boards, document management, etc... in an environment distributed in time and space.

It is a characteristic property of DINE that it provides integrated support for both the group as a whole and the individual user while privately evaluating the positions of other group members. This support is independent on the cooperative or competitive nature of the decision circumstances.

4 DINE

The DINE model supports simultaneous, multiple issue, independent peer-to-peer negotiations. It allows the integration of existing negotiation support techniques which, as opposed to DINE, mostly focus on scenarios where the negotiation issues are shared by all negotiators. The latter techniques are used to support the independent peer- to-peer negotiations in DINE. Negotiators may in fact use any tool even without DINE, as long as it supports the same peer-to-peer information sharing protocol. At the same time, DINE is a generalized multiple criteria decision making model where the alternatives to be ranked are compound subsets of negotiated offers. DINE naturally integrates asynchronous and synchronous communication requirements, intuitive judgement and deep knowledge based techniques. The implementation is based on the Microsoft Windows environment and some of its value added features have already been mentioned.

Our objective here is the critical description of the value- added features related to model-based deep knowledge generation wich bring the Microsoft Excel commercial tool closer to a wide range of task logics, decision cultures and decision circumstances.

The cunstruction of models in general is well supported in spreadsheet environments. There is even integrated support for the specification and solution of optimization models within a spreadsheet (What's Best, IFPS/Optimum, Microsoft Excel Solver). The advantages of such systems over algebraic languages have been analysed in detail [8], [18], we will not go into these issues here.

What difficulties do arise however with existing tools and what kind of further support can be provided for optimization modeling and model experiments in spreadsheets which can improve their scope of usability? Let us list some of these below.

1. The first problem is that while changing most parameters of the model is natural and easy, *changing the size of the model* involves spreadsheet manipulations which are error prone and external to the world of the model itself.

2. The second problem is also related to the size of the model. There are two major reasons why *large models* are increasingly difficult to handle with spreadsheets. The first reason is *memory limitation* which is a question of money and technology scaling only. The other reason is our *cognitive limitation*. The power of the spreadsheet in visualizing data relationships may decrease with larger models unless appropriate data are stored in relational databases and the display structures of the model are carefully chosen in the beginning.

3. The third problem is that existing spreadsheet model building schemes are essentially algebraic which means that a transformation of real world objects and relationships into *algebraic entities and expressions* is necessary. A remarkable possibility for integrating iconic and other representation schemes including spreadsheets is described in [14]. This issue is not discussed any further in this paper, it will be the subject of a further study.

The purpose of the meta-model building tool in DINE is the provision of relief to the first two difficulties above. The solutions provided by DINE are best illustrated in the light of an example.

An example

The example is a simple multiperiod investment problem similar to the one provided as a sample application for Microsoft Excel Solver. The point is not on the validity of the assumptions, but on the new spreadsheet representation and underlying meta-model building tool which solves the first two problems above.

Determine how to invest cash into certificates of deposit (CD) with fixed interest rate and fixed term, so as to maximize interest income while meeting given periodical cash requirements (plus a safety margin). The algebraic formulation of this problem is a typical textbook exercise. The spreadsheet formulation provided as a sample application for Microsoft Excel Solver has its advantages, however it strongly suffers from the above listed difficulties. The DINE approach will preserve the advantages, while resolving the problems.

The *primary concepts* that appear to be necessary for the formulation of the model are the following:

- Date
- Cash requirement
- CD
- Interest
- Term
- Investment

These concepts will be extended during meta-model building with a few secondary quantities which contribute to a better visualization of the data relationships.

The meta-model building tool

The quantities in our example which are appropriate for database storage are the cash requirements with the corresponding dates (a private database) and the CD's with their interest rates and terms (public database). The decision variables are clearly the amounts invested into different CD's at the specified dates (Investment).

CD	Interest rate	Term

Date	Cash requirement

The meta-model of the problem is placed into the first line of a table whose field headings are the primary concepts and some further interesting secondary quantities. On request, our macros interpret the meta-model and replace the line with a table which is then the final model still hot linked to the underlying databases and automatically responding to any intuitive or optimization based changes.

The purpose of the meta-model is the definition of the way the actual model will be automatically built as soon as the underlying databases are available and the user requests it. The meta-model by consequent is independent on the sizes of any databases which determine the size of the model itself, it depends however on the fields of those databases.

The functional decomposition of the model into databases and meta-model provides a solution to the first problem above. It allows an easy reconstruction of the model any time the size of any database changes. The use of the relational database paradigm means a solution to the second problem (keeping a clear view of relationships) from the side of the primary data the model refers to. The databases may even reside on remote servers.

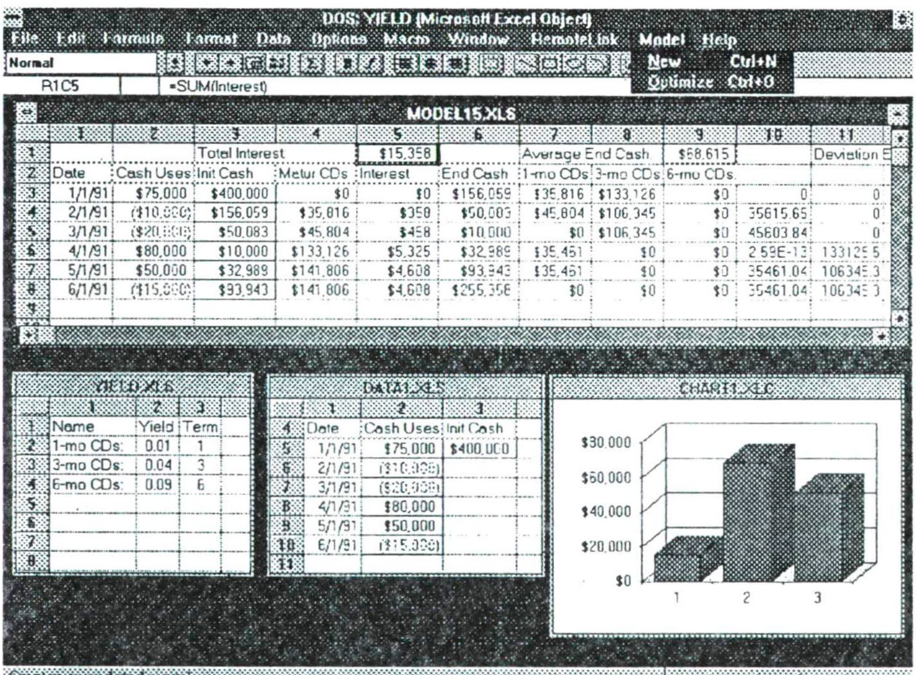

Figure 2. Model, underlying databases, and chart showing model characteristics.

The solution to the second problem from the side of the model, that is keeping a clear view of relationships within the model, is a question of careful design of the model structure in the spreadsheet, and of the most useful decomposition of calculations into secondary result tables. The secondary result tables should in particular include quantities which will serve as constraints to the optimization problem, and should at the same time be useful for the evaluation of the effect of intuitive changes made with the decision variables. From the technical point of view, the primary and secondary result tables have to be defined in such a way that the same spreadsheet formula can provide all required quantities in any given column of the table when the meta-model is expanded into the final model.

5 CONCLUSION

In this paper, we gave a structured guide to the design of distributed decision support systems from a market oriented perspective. We concentrated on Microsoft Windows based tools which can be used on the most popular type of personal computers worldwide.

We illustrated the power of value added systems integration with new features incorporated into a prototype distributed negotiation support application exploiting the advanced capabilities of the Microsoft Excel commercial spreadsheet environment.

References

[1] L.A.Belady: Software Engineering: Beyond Software and Beyond Engineering. In: Shifting Paradigms in Software Engineering (ed. R. Mittermeir). (Springer-Verlag, Wien, New York, 1992) pp.2.

[2] Hernádi,A; Biró,M.; Horváth,T.; Hutter,O.; Király.L.; Knuth,E.; Remzső,T. Window Systems. (Typotex Kft, Budapest, 1990). (in Hungarian)

[3] Biró,M.; Csáki,P.; Vermes,M. WINGDSS Group Decision Support System under MS-Windows. In: Proceedings of the Second Conference on Artificial Intelligence (ed. by I.Fekete and P.Koch). (John von Neumann Society for Computer Sciences, Budapest, Hungary, 1991) pp.263-274.

[4] M. Biró, E. Bodroghy, A. Bor, E. Knuth, L. Kovács, The Design of DINE: a DIstributed NEgotiation Support Shell. In: Decision Support Systems: Experiences and Expectations, (ed. by T. Jelassi, M.R. Klein, W.M. Mayon-White). IFIP Transactions A-9 (North- Holland, 1992) pp.103-114.

[5] M. Biró and I. Maros, The Use of Deep Knowledge from the Perspectives of Cooperative Problem Solving, Systems Modeling, and Cognitive Psychology. In: Shifting Paradigms in Software Engineering (ed. R. Mittermeir). (Springer-Verlag, Wien, New York, 1992) pp.56- 67.

[6] M. Biró, A. Bor, E. Knuth, T. Remzso, A. Szilléry: Spreadsheet-Based Model Building and Multiple Criteria Group Decision Support, MTA SzTAKI Working Paper, 1993.

[7] J. Bisschop and A. Meeraus, On the Development of a General Algebraic Modeling System in a Strategic Planning Environment, Mathematical Programming Study, 20(1982)1-20.

[8] S.E. Bodily, Spreadsheet Modeling as a Stepping Stone, Interfaces, 16, no.5(1986)34-52.

[9] J.M. Bowers and S.D. Benford, Studies in Computer Supported Cooperative Work, North- Holland, 1991.

[10] Brookes, C.H.P. (1986) Requirements Elicitation for Knowledge Based Decision Support Systems, in:McLean, E.R. and Sol, H.G.(eds): Decision Support Systems: A Decade in Perspective, Elsevier Science Publ. B.V. (North-Holland)

[11] T.X. Bui, Co-oP - A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making. Lecture Notes in Computer Science 290. Springer-Verlag, 1987.

[12] S.I. Gass, Model World: Danger, Beware the User as Modeler, Interfaces, 20, no.3(1990)60-64.

[13] Gelleri, P. and Martinez, F. (1988): How to Handle Differences in Importance among Participants with GDSS, in: Lee, R.M., McCosh, A.M. and Migliarese, P.: Organizational Decision Support Systems, North-Holland.

[14] J. Gerlach and F. Kuo, An Approach to Dialog Management for Presentation and Manipulation of Composite Models in Decision Support Systems, Decision Support Systems, 6(1990)227-242.

[15] Holt, C.C. Conceptual Environment for Organization Support Systems: Design and Use of Information Technology in Organization, in:Conf. 'Environments for Supporting Decision Processes', Budapest, Hungary, 18-21 lune, 1990.

[16] Huber, G.P. (1984): Issues in the Design of Group Decision Support Systems, MIS Quarterly, 8.pp.195- 204.

[17] T. Hürlimann, LPL, A Structured Language for Modeling Linear Programs (Verlag Peter Lang AG, Bern, 1987). Keeping the U.S. Computer Industry Competitive: Systems Integration. (National Academy Press, Washington, D.C., 1992).

[18] A. Roy, L. Lasdon and D. Plane, End-user optimization with spreadsheet models, European Journal of Operational Research, 39(1989)131-137.

[19] Sol, H.G. Information Systems to Support Decision Processes, in:Conf. 'Environments for Supporting Decision Processes', Budapest, Hungary, 18-21 lune, 1990.

[20] Yazdani M. Shells Versus Toolkits. Expert System User, January 1989, pp.13.

Received June 28, 1993