SCHOOL FINANCE SIMULATOR

John J. Willson
Nichols Applied Management

Graham Macmillan
Peat, Marwick and Partners

Edmonton, Alberta, Canada

ABSTRACT

Keywords: Decisions Support Systems Applications, Computer Modelling, Education Financing.

The school finance simulator is a Decision Support System (DSS) built to simulate effects of funding changes at the Departmental level upon school jurisdictions within the Province of Alberta. The approach taken to create the "simulator" incorporated the following data processing related issues:

- the use of an established *System Development Life Cycle* methodology
- the adoption of a *prototype* approach to flesh out the user simulation needs
- the creation of the DSS as *inter-active* so that a simulation session would successively approach an acceptable result
- the interfacing of the user through *decision system support* English language like software
- the use of software techniques such as *structured analysis,* design, and programming

The school finance simulator project resulted in the creation of a powerful tool for educational policy planning and research which is used and accessed by the policy makers themselves.

I. BACKGROUND AND OBJECTIVES

Alberta funds its 132 school jurisdictions via a foundation type program fund. Originally four grants were allocated to each jurisdiction to cover instruction, transportation, administration and debt service costs. However, due to surpluses in the foundation fund, additional grants have been added to meet the specialized needs (e.g., handicapped students) of some individual students within the province. In 1981 Alberta school boards qualified for over 85 individual grants and received over 890 million dollars in funding. (See exhibit 4 for the conceptual diagram of the Alberta grant allocation system.)

With every new grant, and with every modification to an existing grant, it became more difficult for provincial policy makers to predict the likely impact of any proposed policy changes. The difficulty of this task was exacerbated by the fact that virtually all school finance analysis was conducted manually. The net result of manual analysis is its inevitable partial nature due to the computational complexity of altering more than one grant or parameter.

In order to produce a new and better analytic technique for exploring the consequences of modifying grant provisions and/or alternative school finance plans on the total system, Alberta Education initiated this study to develop a school finance simulator.

The DSS was designed to provide the following:

1) *An aid allocation simulation and optimization capability* detailing the impact of various aid programs, district by district, by school system type, and for the Province as a whole. 'Impact" is defined in terms of:

- the final, aid distribution pattern;
- local requisitions and requisition rates;
- local expenditures;
- aid parameter values for a given program or specific grant,
- the total dollar cost implications for the Province
- the necessary grant parameter values for a total dollar appropriation, parameter values.

2) *An executive reporting capability.* A standard set of summary reports incorporating trend analyses and comparative statistics by school district, pre-specified groups of districts or for the Province as a whole. Subsequently this analysis was output in graphic form.

SPONSORING AGENCY ALBERTA EDUCATION.
HONORABLE DAVID KING, MINISTER OF EDUCATION.

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors

In addition the DSS was to be able to be both maintained and used by minimally experienced data processing professionals and management personnel.

II SELECTED CONCEPTUAL DESIGN

Our selected approach was based on a detailed interview program and literature search. The following four design alternatives were investigated with respect to operationalizing the school finance computer model:

1. Using the existing Alberta Education MIS database and supplementing it with an inquiry type language.
2. Acquiring a software package that had a pre-built model DSS contained within it.
3. Customize/build a DSS.
4. Use an approach which combines the first three.

The inquiry language approach was rejected because the complexity of these languages necessitates the use of sophisticated computing science people. This requirement ran contrary to the strong user preference that the model be readily accessible and useable by senior management.

The acquisition option was rejected due to the lack of relevant pre-built software packages on the market at that time. The closest model was the SFEMS (5,9) (School Finance Equalization Management System), however, it required extensive modification to meet Alberta Education needs. SFEM's strength is its ability to analyze changes in granting philosophy and because of this ability, the DSS which was constructed did have the flexibility to add-on the inexpensive SFEMS.

Having narrowed the alternatives to the customize/build approach we then looked at

- the traditional construction approach and
- the prototype approach

A prototype approach was chosen as it lends itself to the continuous evolution of end-user requirements. A successful working DSS was created by the system designer through the construction of different prototype versions which are modified to satisfy end-user demands. The prototype approach lends itself to designing a system that handles planning and strategy changes as opposed to freezing and capturing operational data. Therefore, design changes can be accommodated before large resource committments have been made. In addition, the approach encouraged the end-user to work down the model's learning curve before the final version was operational. This prototype approach is called an evolutionary strategy for implementing a DSS. (28)

Customize/build options were examined in terms of minimizing start-up and turnaround time and maximizing model life expectancy and maintainability. Meeting these objectives suggested the use of one of several widely used high level languages that have a broad user base. Incorporation of a high level language would also allow the DSS (or parts of it) to become a standard in-Department MIS responsibility during the model's life-time. The high level language also lends itself to ease of modelling and a high degree of interactiveness. The interactive feature was deemed to be very desirable because it met the identified need to respond quickly to the likely question sets posed by personnel internal and external to the Department. Our choice of a high level language was APL. It was chosen because it is:

- flexible as a programming language
- maintainable
- able to meet response time demands

We also considered acquiring a software language such as SIMPLOT, TAB, SPSS under the customize/build approach. This sub-approach was rejected for two major reasons: maintainability and compatibility with the (AGS) system. In terms of maintainability the fact that these languages are only vendor supplied means that it is very difficult to acquire programmer expertise. In terms of compatibility, many simulation languages are not supported by the existing AGS computer configuration.

III. SYSTEMS PLANNING OF THE MODEL

The systems planning of the model will be discussed in terms of the phases of the system *d*evelopment *l*ife *c*ycle (SDLC) methodology chosen and in terms of the techniques employed to make the methodology work.

There was more than one SDLC employed through the project; the elapsed time spanned 6 months. The methodologies used were primarily:

- **A**lberta **G**overnment **S**ervices Traditional Method

and

- **S**ystem **D**evelopment **M**ethod (SDM) of Peat, Marwick and Mitchell

The first phase of the project of building the model followed the AGS Traditional approach.

This phase was entitled the Initiation phase. During the Initiation phase the feasibility of and approach to the DSS were presented to an Alberta Education steering committee for acceptance. However, it was found that although the project was conceived as large in terms of expenditure and effort by Alberta Education, the building of the model could not exactly follow the traditional build approach. The AGS Traditional approach did not lend itself as a methodology for model or DSS building but more as a method for building production oriented data processing systems. Subsequently, the SDLC of SDM was chosen and followed.

The phases of the system development method were:

- a Requirements Definition phase
- a Conceptual Design phase
- a Detail Design phase
- a Development phase
- an Implementation phase
- a Post Implementation Review phase

These phases had numerous tasks associated with their work programs and documentation of the deliverables of each task and phase. It is important to note that by the prototype approach several reiterations of the detail design, development, and implementation phases were performed.

Although most SDLC's have strengths and weaknesses that can be categorized in many ways (17), we found the techniques of SDM to be particularly effective in building a DSS. The techniques we made major use of in this methodology included:

- structured system analysis tools including: data flow diagrams and data dictionary
- structured system design tools such as process structure charts
- English language walk-throughs of the paper model before computer program construction
- human engineering concepts of the Decision System Support software, a close to English-menu driven interface between the, end-user or policy maker and the computer system
- incremental testing, including top-down testing, through the use of a preconstructed prototypes
- change control subsequent to the prototype sign-off and detail design phase acceptance
- data base design both traditional with a traditional language (Fortran) and free form design with a mathematical modelling language (APL).

These techniques in themselves constituted documentation standards that were easy to follow, appropriate, and heuristically valuable to Alberta Education and the construction team.

V. MODEL DEVELOPMENT AND IMPLEMENTATION

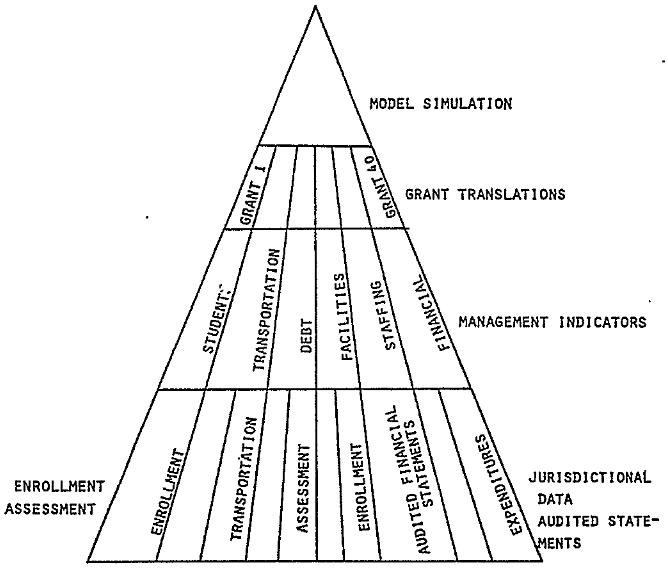
Model development will be described in terms of the level of DSS detail, the computing systems used to create, load, and run the DSS, and in terms of some implementation issues.

Level of Detail

The level of detail of the DSS is shown in the chart exhibit #1 (with apologies to Richard Nolan (18)). The maximal level of accurate, acceptable detail statistics included:

- projected and actual enrollment
- projected and actual assessment
- jurisdictional auditted financial statements (1976-1979)

EXHIBIT 1 — DATA HIERARCHY



The detail was condensed at times to management indicators (e.g., student-teacher ratios), that showed trends or reasonableness of the detailed data. However, *all* numbers used in the simulations were at the detail level and considerable effort was expended locating the statistical values to be put in the model. The existing granting policies and legislation was translated into mathematical symbolism, and often user sign-off translated into APL.

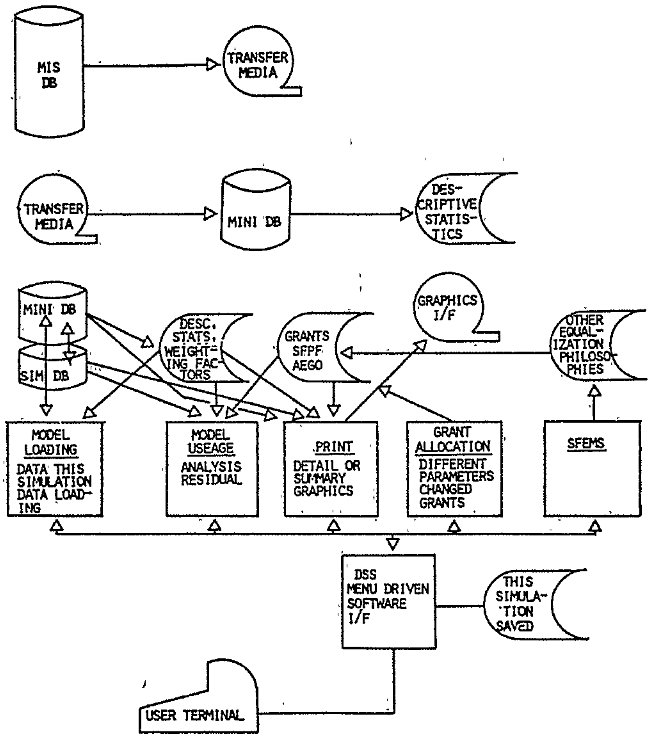
The model created a provincial pro-forma statement based on model general or verified pro-forma statements for individual jurisdictions within the province. The model also incorporated the capability to estimate pro-forma statements for jurisdictions given a pro-forma statement for the province and certain assumptions such as save-harmless.

Note should be made by the reader of the effect of swings in the data hierarchy pyramid. A change at the bottom of the pyramid does not substantially affect the model or policy options given that the model has some base line values. However, a small change in policy in the model at the top of the pyramid could eliminate the need for much of the bottom pyramid detail or create a need for considerably more detail. The dramatic effects of caused changes at the top of the pyramid underlined the need for a high degree of end-user input during construction. The prototype or process approach ensures user participation.

Computing Systems

The model spanned several computer systems, several languages, and many data structures. The diagram exhibit

EXHIBIT 2 – A SYSTEM FLOW CHARTS



#2 shows the hardware oriented approach to building the model. The existing Alberta Education Management Information System database was passed using COBOL and MARK IV to create a magnetic tape of detail statistical data. This was necessary since the existing government service bureau did not support APL. The monthly updated tape became the transfer media that carried many of the statistics required to load the data collection aspects of the model and was an end-user requirement deliverable in itself. Many additional data elements were loaded or changed by the end-user when running the model. The existing client data processing personnel did this as part of their monthly production tasks.

The model then used a different operating system, MTS, (Michican Terminal System), that was more user oriented and forgiving. The Decision System Support Software interface was written in Fortran to accommodate interfacing file handling with additional software packages and APL file types. At that time the targetted external service bureau did not have an operating system file to workspace handler. The model itself was written in APL to assist in ease of change. The ability to consider a jurisdiction as a collection (a multi-dimensional matrix) of data was considered important. The ability to, with one command, reshape that matrix was considered very important. The legislation was translated into APL functions and it also could be changed quite quickly by the end-user. This created a need for the user to become quite creative in his examination of policy questions as opposed to passive during traditional information requests. This provided both decision support and "creativity enhancement" (32) to the policy makers.

Indeed, the man-machine boundary requires the end-policy-user to set up a simulation session in his head or on paper before sitting down at the terminal in order to maximize the utility of the output. However, the user can easily change or fine tune the policy question interactively and see the resultant effect. This can be seen in the diagram exhibit #3 as many simulation snap-shots can be taken during a simulation run.

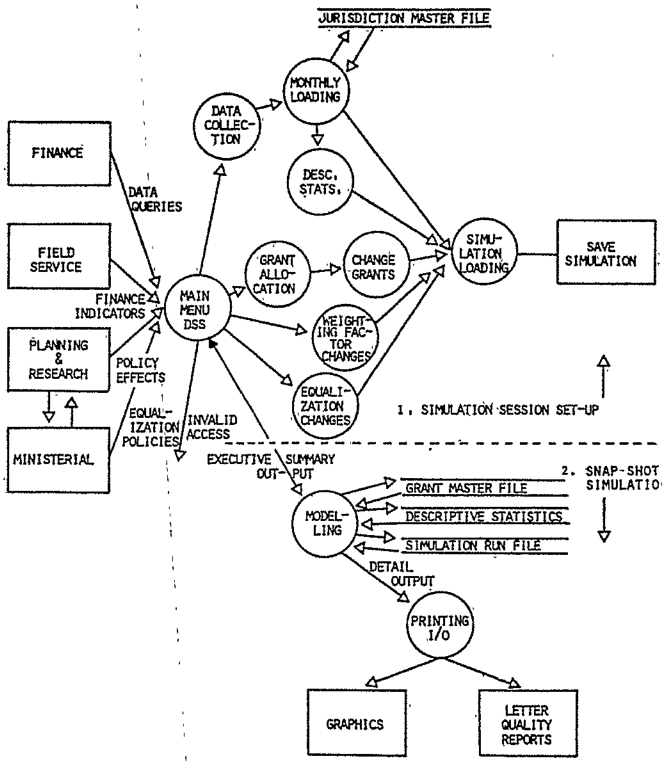
Implementation

The implementation of the total DSS commenced with the demonstration of the paper model and was further presented with the prototype – a very restrictive working version of the model. The DSS has been fully signed-off at the time of writing this paper. The goal of implementation of any computer system is end-user satisfaction and the primary user, the associate director of planning, has described the DSS as "similar to the hippopotamus in a tutu from Fantasia. The DSS is big but performs the most intricate ballet steps well"! All potential end-users of this model have been directly involved through all phases of the SDLC and a smooth transition to user personnel has been accomplished.

V. OPERATION OF THE SIMULATOR MODEL

The best example of the operation of the DSS is given through an English-language walk-through that was prepared during the first phase and subsequently became a design idea.

EXHIBIT 3 – CONCEPTUAL DATA FLOW DIAGRAM – PHASE I



An English Language Walk-Through

The policy question posed is; what would be the dollar figure in a dollar per pupil grant which would replace SREG and EOF, (two AEGO grants), while not reducing total board revenue. At any time the use could use the full grants name i.e. AEGO is Alberta Education Grants Order. (See exhibit #4)

Resolution

1. Sign on and direct the model to the grant matrix file where each SEPF and AEGO grant is represented in arithmetic form.
2. The user would then communicate with the "decision system support software" or English language interface which precludes the need for the user to possess computer language skills. (See exhibit #5)

The English language interface would ask questions such as:

- which grant(s) do you wish to modify?
- do you wish to set SREG = 0?
- do you wish to add a new grant?

And then translate the answers to these questions into the program language of the model.

3. After the user has set SREG and EOF = 0 and specified the parameters of the new grant he would then ask for a simulation.
4. During the simulation run, the model would calculate revenue generated from the new grant and revenue lost by eliminating SREG and EOF, by referring to descriptive statistics such as jurisdictional enrollment which have been generated from a mini data base. The use of the descriptive statistics file negates the process of passing a large data base thus reducing run costs. If new descriptive statistics are needed to run the simulation, the mini data base would be passed and the statistics generated and stored for future use.

EXHIBIT 4
CONCEPTUAL REVENUE FLOW MODEL

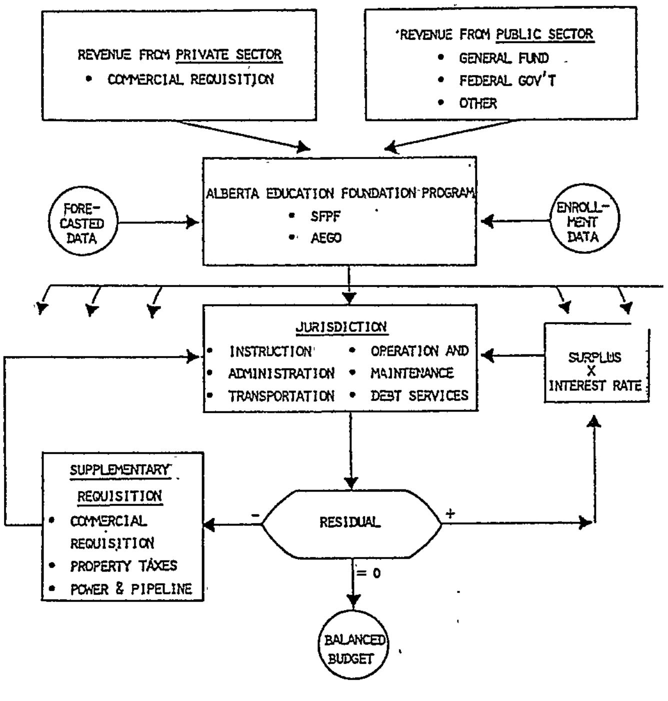
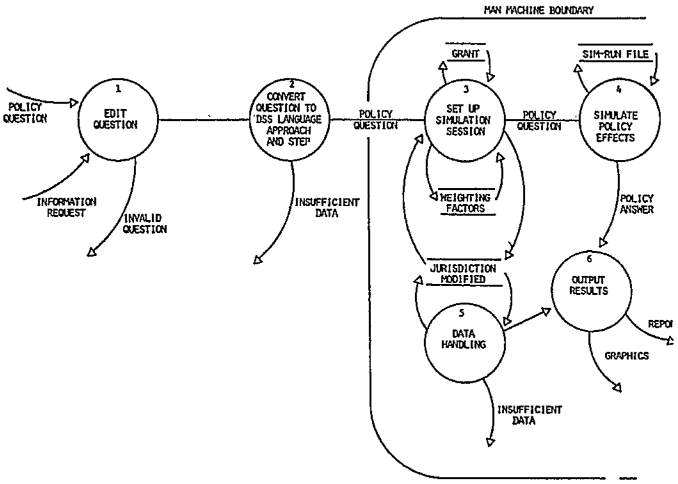


EXHIBIT 5 — CONCEPTUAL DFD — PHASE II



5. The output of the simulation would be in executive summary form with only a few results and/or graphics being reported. In our example, the first simulation run reported that there was a large shortfall in total board revenue under the assumption of the initial specifications.
6. The user would then re-enter the grant matrix file, increase the per pupil grant, and re-run the simulation. The user would analyze the results and refine the per pupil grant until he was satisfied with the overall effects of the grant changes.
7. After the best simulation had been chosen, the user would then ask for a detailed run which would illustrate the effects of the grant changes on every, or a selected group, of jurisdictions in the province.
8. If certain key jurisdictions are adversely affected, detailed runs or other promising simulations would also be asked for.
9. When sufficient information was received, the user would sign-off after about ½ hour elapsed time.

VI. MAINTENANCE

Maintenance of the system can be described in terms of monitoring of the software and the software-like requirements, and monitoring the data. From the Nolan pyramid, exhibit #1, we see the data is not critical if we define default data. This was done as the benchmark data with base year 1978. Should the model be unable to use a given year's data actual or projected then the model uses 1978. House-keeping of the data should be minimal with much of it done by the existing MIS department on an ongoing basis.

Maintenance of the software can be complicated. Changes in base year legislation, 1978, would be done on a yearly basis. The user can do much of this himself through the use of the decision system support software. However, changes to the software interface requires the use of direct data processing personnel. Their response turn-around time has been minimal given:

- the modern documentation standards employed

and

- the change control mechanism of the System Development Method.

VIII. MANAGEMENT AUTOPSY

The post-implementation review has not been done on this project at the time of writing of this paper, hence the management autopsy cannot be comprehensively written now. However, the review of the system will evolve over the model's life expectancy of 3 to 5 years. As the end-users travel along the DSS learning curve, model useage and ease of enhancements will decide whether the final delivered system meets the objectives of Alberta Education.

BIBLIOGRAPHY

List of Models Reviewed

1. Inman, Robert P. and Douglas Wolf, University of Pennsylvania, Philadelphia, PA 19174 U.S.A. "A Simulation Program for Predicting and Evaluating The Policy Effects of Grants in Aid". **Socio-Econ Plan Sci Vol #10** pp 77-88, Pergamon Press 1976.
2. Stern, David, "Effects of Alternative State Aid Formulas on the Distribution of Public School Expenditures in Massachusetts", **The Review of Economics and Statistics,** Feb. 1973. Vol #55, pp 91-97.
3. Lindman, Erick L., University of California, Los Angeles, "Foundation Programs and Percentage Grants", **Journal of Education Finance,** Fall 1975, pp 270-273.
4. Nickaus, John M., University of Florida, Gainesville, "Budget Simulator: Computerized Program Cost Analysis and Program Budget Preparation", **Journal of Education Finance,** Spring 1977, pp 430-443.
5. **School Finance Equalization Management System,** Educational Policy Research Institute, Princeton, N.J.
6. Sklar, Sigmund L. and William E. Loup, "A Prototype National Educational Finance Planning Model", Dec. 1971, **Presidents Commission on School Finance.**
7. "The Oregon School Finance Simulation" in **State School Finance Alternatives: Strategies for Reform,** Pierce et al, Oregon University, ED105571.
8. Steffensen, Robert, G., The Utah State **Public School Finance Simulator,** Educational Planning 2, 4, 8-9, Jun 73.

List of Books and Articles Reviewed

9. **School Finance Equalization Management System: An Overview.** Educational Testing Service: Princeton, N.J. Education Policy Research Institute. Apr 77, 27p;
10. Jargowsky, Peter P.; Moskowitz, Jay. **Systems Division Report. The New Jersey Education Data System.** SDR.76.45. Educational Testing Service, Princeton, N.J.
11. Pierce, Lawrence C.; and others. **State School Finance Alternatives: Strategies for Reform** Oregon University, Eugene. Center for Educational Policy and Management. May 75 182p. Sponsoring Agency: National Inst. of Education (DHEW), Washington, D.C.
12. Calahan, John J.; Goettel, Robert J. **Revising State Education Finance Systems: Simulating Policy Alternative.** April 72 American Educational Research Association Annual Meeting
13. Pernal, Michael. "Efficiency and Accountability: A computer Assisted Financial Aid Operation for the Small College". **Journal of Student Financial Aid,** 7, 3, 47-55 Nov 77
14. Jargonwsky, Peter; and others. School Finance Reform: Decoding the Simulation Maze **Journal of Education Finance,** 3,2, 199-213.
15. Norris, Graeme, "Justifying Research and Teaching Objectives in Universities" ED183093 226p. Mar 80, **Renouf, U.S.A. Inc.,**
16. Linthicum, Dorothy S., **Economic Impact Manual for Maryland Community Colleges.** Maryland State Board for Community colleges, Annapolis.
17. "An Experimental Model for Analyzing Strategies for Financing Higher Education in New York State.' New York State Education Dept., Albany. Office of Post-Secondary Research, Information Systems, and Institutional Aid. Dec 77.
18. "Proposed State Aid Formula for South Dakota". ED139094 (842 Assisted) School Finance Equalization Study Workshop (Denver, Colorado, March 3-4, 1977. South Dakota State Div. of Elementary and Secondary Education.
19. Weldon, Kent; Huckfeldt, Vaughn. "Prototype Software for a Federal Planning Model for Analysis of Accessibility to Higher Education". ED098905 Western Interstate Commission for Higher Education, Boulder, Colo. National Center for Higher Education Management Systems. 98.; Related Documents are HE 066 086 and 088 73.
20. Chick, Charles E.; and others. "School Construction Programming in Florida" K-12. ED084705 96p.; Paper presented at National Symposium on State School Finance Reform (Washington, D.C. November 26 and 27, 1973.); Related Documents.
21. Matzke, Orville R.; "A Linear Programming Model to Optimize Various Objective Functions of a Foundation Type State Support Program". ED064331 45p.; Paper presented at the Annual Meeting of the American Educational Research Association (Chicago, Illinois, April 1972.

22. Sklar, Sigmund L.; Ioup, William E. **A Prototype National Education Finance Planning Model.** "Projections of Educational Needs, Resources and Disparaties Under Various Forecasting and Policy Assumptions".

23. **Temporary Special Levy Study Commission.** "Volume II: Research Reports. Final Report". Washington State Temporary Special Levy Study Commission. Mar 71.

24. Bright, R. Louis; Gidenonse, Hendrik D. **Education Research and Its Relation to Policy,** "An Analysis Based on the Experience of the United States. Oct. 67

25. Atkinson, R. Wayne, "Structured Approach Aids Implementation of Business Systems" **Canadian Datasystems,** May 1979

26. Nolan, R.L., "Managing the Crises in Data Processing" **Harvard Business Review,** March-April 1976

27. Tsichutzis, Dionysios, "The Architects of System Design" **Datamation,** September 1980

28. Alavi, M. and J.C. Henderson, "An Evolutionary Strategy for Implementing a Decision Support System", **Management Science,** Nov 1981.

29. Bonczels, R.H., C.W. Hosapple, and A.B. Winston, "A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management", **Operations Research,** Mar 81.

30. Ize, P. and A. Sen, "Logical Data Base Design in Decision Support Systems", **Journal of Systems Management,** May 1981.

31. Briggs, W.G., "An Evaluation of DSS Packages", **Computerworld,** March 1982.

32. Wagner, G.R., "Decision Support Systems: The Real Substance", **Interfaces,** Apr 1981.

33. Hansen, J.V., L.J. McKell, and L.E. Heitzer, "ISMS: Computer-Aided Analysis for Design of Decision-Support Systems", **Management Science,** Nov 1979.

34. Prastacos, G.P. and E. Brodheim, "PBDS: A Decision Support System for Regional Blood Management", **Management Science,** May 1980.