AGENCY FOR INTERNATIONAL DEVELOPMENT
WASHINGTON, D. C. 20523
BIBLIOGRAPHIC INPUT SHEET

FOR AID USE ONLY
Batch 62

1. SUBJECT CLASSIFICATION
A. PRIMARY: Food production and nutrition — AE10-0000-0000
B. SECONDARY: Agricultural economics

2. TITLE AND SUBTITLE
Computer library for agricultural systems simulation,a progress report

3. AUTHOR(S)
Adkin,M.H.; Carroll,T.W.

4. DOCUMENT DATE	5. NUMBER OF PAGES	6. ARC NUMBER
1976	53p.	ARC

7. REFERENCE ORGANIZATION NAME AND ADDRESS
Mich.State

8. SUPPLEMENTARY NOTES (Sponsoring Organization, Publishers, Availability)
(In Computer Library for Agr.Systems Simulation. CLASS doc.no.1)
(Final rpt.of commputer program: CSD-2975 Res.)

9. ABSTRACT
A PROJECT: To develop and operate a Computer Library for Agricultural Systems Simulation (CLASS) in one or more of the standard computer languages indexed and made available to borrowers on a national and international basis.
DATES: October 1972-July 1976.
DEVELOPMENTS: Approaches are examined for institutionalizing an analytical capacity for planning, policy formulation, program development, and project implementation for agricultural sector development within the public decision making structure of developing countries. A major component of the analytical capacity is a series of system simulation models tailored to the needs of the individual country. The library acquires, catalogs, maintains and distributes computer programs and associated documentation. The library sets standards of admissibility for programs and documentation; catalogs and indexes programs and documentation so as to facilitate their retrieval by users seeking a set of programs to be used in a specific problem analysis; and distributes programs and documentation to users. Progress during the development and implementation stage discussed in this report includes: design of an idealized scope and structure for the library, specification of standards for library elements and their documentations and computer programs, acquisition of the components, routines and language, investigation of alternative institutional arrangements, and utilization of CLASS elements. The report concludes with a discussion of activities yet to be undertaken.

10. CONTROL NUMBER
PN-AAD-980

11. PRICE OF DOCUMENT

12. DESCRIPTORS
Computer programs
Libraries
Sector analysis
Simulation

13. PROJECT NUMBER

14. CONTRACT NUMBER
CSD-2975 Res.

15. TYPE OF DOCUMENT

AID 590-1 (4-74)

COMPUTER LIBRARY FOR AGRICULTURAL SYSTEMS SIMULATION: A PROGRESS REPORT

Michael H. Abkin

Tom W. Carroll

COMPUTER LIBRARY FOR AGRICULTURAL SYSTEMS SIMULATION

CLASS Document No. 1

Agricultural Sector Analysis and Simulation Projects
Supported under Contract AID/csd-2975
U. S. Agency for International Development

Department of Agricultural Economics
Michigan State University
East Lansing, Michigan

July 1976

CONTENTS

Page

INTRODUCTION 1

BACKGROUND AND RATIONALE 2

DEFINITION AND SCOPE 5

Scope of CLASS Contents 5

Generality 6
Components and Models 7
Technical Routines 8
Simulation Language 9

Scope of CLASS Activities 9

CLASS STANDARDS 11

Software Selection Standards 12

General Applicability 12
Theoretical and Empirical Validity 13
Compatibility 13

Documentation Standards 13

General 14
Announcement 14
Abstract 15
Documentation 15

Programming Standards 16

PROCEDURES AND ORGANIZATION 17

Procedures 17

Acquisition and Development 17
Storage and Maintenance 18
User Services and Communications 19

Organization 20

Page

PRESENT STATUS AND UTILIZATION OF CLASS 24

FUTURE ACTIVITIES 26

Announcement 26

Acquisition 26

Advisory Boards 27

Operation of CLASS 27

Communications 27

Consulting 28

Cooperation with Other Institutions 28

Developmental Research 28

REFERENCES 29

APPENDIX: Index to CLASS Software 31

COMPUTER LIBRARY FOR AGRICULTURAL SYSTEMS SIMULATION

The Computer Library for Agricultural Systems Simulation (CLASS) is one of the four major activities of the Agricultural Sector Analysis and Simulation Projects at Michigan State Unviersity under U. S. Agency for International Development Contract AID/csd-2975. The other three major interrelated project activities include theoretical and methodological research, the Development Analysis Study Program, and field activities, primarily in the Republic of Korea.

The project objective is to develop an approach to institutionalizing an analytical capacity for planning, policy formulation, program development, and project implementation for agricultural sector development within the public decision making structure of developing countries. A major component of the analytical capacity is a series of system simulation models tailored to the needs of the individual country. Much of the experience gained from the field activity and the knowledge gained from the theoretical and methodological research added to the present stock of knowledge about building and maintaining analytical capacities for agricultural sector development can be preserved and extended in the training provided through the Development Analysis Study Program and in the stock of model, component, and utility routine computer software documented in the Computer Library for Agricultural Systems Simulation.

In full operation, the Computer Library for Agricultural Systems Simulation (CLASS) acquires, catalogs, maintains and distributes computer programs and associated documentation. These computer programs are of generalized simulation models, components, and routines designed specifically for the analysis of agricultural development problems and processes. In particular, the library sets standards of admissibility for programs and documentation; catalogs and indexes programs and documentation so as to facilitate their retrieval by users seeking a set of programs to be used in a specific problem analysis; and distributes programs and documentation to users.

To enhance the effectiveness of the library, its functions also include identifying and soliciting needed models; actively bringing programs and documentation up to the library's standards; and providing limited consultation in identifying and implementing appropriate library programs for a particular application. A subsidiary function of the library in conjunction with the identification and solicitation of models is to survey and catalog ongoing research in agricultural systems modeling and simulation.

The CLASS document publication series is the main vehicle for informing potential users of the substance of CLASS holdings and activities.

July, 1976

George E. Rossmiller
Director
Agricultural Sector Analysis and
Simulation Projects

THE COMPUTER LIBRARY FOR AGRICULTURAL SYSTEMS SIMULATION: A PROGRESS REPORT

Introduction

The scope of work of Michigan State University's contract AID/csd-2975 with the Agency for International Development includes the following paragraph:

> Establish a software library, initially at MSU, but perhaps to be transferred later to the proposed IDI of A.I.D. The various mathematical models and computational routines referred to here as "software" should be placed in one or more of the standard computer languages, indexed and made available to "borrowers" with appropriate explanations. This will involve generalization of existing components and the provision of a higher-order simulation language for the integration and combination of components into configurations required in various institutional and geographic contexts. Such an arrangement is needed to make the model components available on a national and international basis to potential users in donor, lender, and host country agencies.

A proposal [8] for the establishment of a software library was drafted in October 1972. The present report discusses the progress made to date--at the end of the above AID contract--in the development and operationalization of the Computer Library for Agricultural Systems Simulation, CLASS.

Although for purposes of Contract 2975 this is a final report on CLASS, from the CLASS perspective we must consider this a progress report. Michigan State University has submitted a proposal to the Technical Assistance Bureau of AID for funding to continue CLASS activities another two years. In addition, we are exploring [9] the possibility of transferring CLASS to the International Institute for Applied Systems Analysis (IIASA) in Austria.

Progress during this preliminary development and implementation stage under the AID contract has included:

1. design of an idealized scope and structure for the library
2. specification of standards for library elements and their documentations and computer programs
3. acquisition--i.e., development, programming, and documentation--of the components, routines, and language indexed in Appendix A
4. constitution and meeting of the Policy Advisory Board
5. investigation of alternative institutional arrangements for continued development, maintenance, and utilization of CLASS (e.g., [9])
6. utilization of CLASS elements

We discuss each of these accomplishments in this report, and we conclude with a discussion of activities which could be undertaken in the next couple of years to further develop, expand, utilize, and institutionalize the library.

Background and Rationale

Under contracts AID/csd-1557 and AID/csd-2975, in Nigeria and in Korea [1,2], the Michigan State University (MSU) agricultural sector simulation team has developed and applied the generalized system simulation approach to agricultural development analysis and policy making. As part of that effort, models were developed, for example, of production and processing processes; of demand and consumption; of demographic processes as applied to human, cattle, and tree populations; of commodity, regional, and national accounting; of resource allocation decisions; and of the diffusion of innovations process.

Experience has demonstrated that these and other processes are of general interest in the developing (and developed) world and that generalized models of them could be built and then adapted and assembled in various configurations to help analyze particular problems in particular countries.

For example, the cattle component of the Nigeria model was adapted and used to analyze cattle industry policies in Venezuela [3].

In addition, in building these models, functional components have been developed which have shown promise of being generalizable for application in a broad range of situations. For example, demographic components individually designed to model specific populations of people, trees, and cattle may be generalized so that a single component would model any of these types of populations as well as others that may be of interest, such as tractors, houses, cars, bacteria, insects, etc.

A software library of such generalized models can have many advantages for its potential users. First, where models or components are needed for a particular application, a central location would simplify the task of finding out whether appropriate models are available or whether it would be necessary to either build them on an ad hoc basis or to do without. In this regard, where there is a central repository, duplications of effort in model building can be avoided or at least reduced. Perhaps the largest advantage would derive from a standardization of models and organization. Specifically, models and components can be assured to be linkable with one another as necessary for a particular application, and they will certainly be programmed in the same style and computer language. In addition, the style, format, and content of documentation will be standardized, as will communication with and distribution to actual and potential users.

Finally, it is well worth mentioning that, while performing such services, a software library will also serve as a vehicle for preserving and using, as capital, models and components developed by researchers around the globe. While such libraries are in operation for other application areas, there has been no comparable attempt to establish one for the agricultural development field.

Users serviced by such a library will include individuals as well as public and private organizations and will fall into two classes. First, individuals and organizations doing teaching or research can use appropriate models and/or components from the library as teaching aids or research tools, both as a part of regular university courses and curricula and in high-level professional training programs. The other class of library users includes public and private organizations serving as consultants or contractors to national or local governments (or government agencies themselves) and charged to do policy, program, and project analyses.

With this rationale, the development of such a software library--CLASS, the Computer Library for Agricultural Systems Simulation--was initiated, and a Simulation Library Working Conference was held in March 1973 with representatives from national and international development assistance agencies, LDCs and relevant disciplines to consider the potential demand for such a library and how best to organize it. The conferees concluded, briefly, that [4]:

1) there is clearly a need for developing analytical capacity, particularly in LDCs, for policy and other decision-making in development planning;

2) CLASS can contribute to the satisfaction of that need;

3) the library should be open to any analytical techniques which prove themselves useful;

4) a passive clearinghouse would not be sufficient, but rather complementary inputs, particularly expertise, need to be actively provided to successfully implement library components; and

5) such expertise is a major limiting resource and, therefore, training is a high priority, complementary activity.

The remainder of this report describes the preliminary definition and organization of CLASS as it has been initiated at MSU. Succeeding sections

discuss definition and scope, standards, and procedures and organization. We conclude with a summary of the present status of CLASS and a discussion of future activities.

Definition and Scope

Consistent with the recommendations of the Simulation Library Working Conference discussed above, the library is defined as follows:

> The Computer Library for Agricultural Systems Simulation (CLASS) is viewed as a unit which acquires, catalogs, maintains and distributes computer programs and associated documentation. These computer programs are of generalized simulation models and routines designed specifically for the analysis of agricultural development problems and processes. In particular, the library sets standards of admissibility for programs and documentation; catalogs and indexes programs and documentation so as to facilitate their retrieval by users seeking a set of programs to be used in a specific problem analysis; and distributes programs and documentation to users.
>
> To enhance the effectiveness of the library, its functions also include identifying and soliciting needed models; actively bringing programs and documentation up to the library's standards; and providing limited consultation in identifying and implementing appropriate library programs for a particular application. A subsidiary function of the library in conjunction with the identification and solicitation of models is to survey and catalog ongoing research in agricultural systems modeling and simulation.

The scope of CLASS lies in two dimensions: the kinds of models, components and routines it inventories; and the activities and services it performs or offers.

Scope of CLASS Contents

The library stocks three categories of computer programs with associated documentations: generalized <u>function-oriented components</u>, generalized <u>problem-oriented models</u>, and <u>technical programs</u> to assist in building, testing, and validating and using models and components.

Generality

The quality of "generality" is actually an open-ended continuum wherein movement in one direction is towards the more specific and in the other towards the more general. In this sense, "complete" generality is impossible. Indeed, extreme generality in the model is as undesirable for the library as is extreme specificity. While a model of the latter type may be so pragmatic as to have no application but one, one of the former type may be so abstract and unrealistic as to have no practical application at all. The implication is that the library must aim for a middle ground, and, hence, a set of models for each process (including different levels of aggregation) so that within the set can be found a model which meets the requirements of a given application. Of particular relevance for the kinds of models to be included in CLASS is the fact that, as models become larger and more problem-oriented, they of necessity lose generality. Thus, there will be trade-offs to consider.

Models of socioeconomic processes may be general in two dimensions. First, a model can be of a generalized process. For example, a model of a demographic process may be used for human, tree, livestock, or capital equipment populations; or a processing model may be suitable for the processing of cocoa, oil palm products, rubber or tobacco (as in the Nigerian simulation model [1]). The other dimension of model generality is with respect to applications. For instance, a population model, a demand model or a production model may be applicable to analyses of food production problems in Tanzania, cattle industry problems in Venezuela or Colombia, or agricultural sector problems in Nigeria or Korea.

Components and Models

Generalized function-oriented components (hereinafter called components) can be defined and developed which would serve in any number of applications. For example, a production component which was essentially an input/output routine could be used for the production of fish, livestock, crops, industrial goods, and public and private services. A single demographic component could model populations of people, cattle, trees, automobiles, and capital goods, such as housing, tractors, and factories. An accounting component could be designed which would perform industry, sector, and national accounting functions.

In keeping with the building block approach, such generalized components can be combined to form any number of larger models for use in the investigation of a wide spectrum of specific problems. For example, to look at the problem of low productivity of agricultural resources, we might build a model using a production component, a resource allocation decision component, an accounting component, and a demand component, all linked together with an overall executive program which would generate policy inputs and display results. In this way, problem-oriented models (hereinafter called models) can be built to examine a wide range of problems in the agricultural sector or any other sector.

Problem areas of concern to agricultural sector development, for example, may be distinguished in three fields: economic--including resource allocation, resource productivity, inflation, income distribution, etc.; social--including health care, population, education, housing, law enforcement, etc.: and physical infrastructure--including power systems, communication systems, public works, etc. These fields are related and, indeed, over-

lapping; and analysis of a particular problem may require components and models from all of them.

So far, then, we have defined two types of CLASS elements: generalized function-oriented components and generalized problem-oriented models. Of course, as new and improved components and models become available, they will join the library, perhaps replacing units already there. Furthermore, as old problems cease to be of concern, certain problem-oriented models may fall into disuse and be retired from the active library, to be replaced by new models designed for new problems.

Technical Routines

There is one more type of library element contained in CLASS: technical programs and routines (hereinafter called routines). In building, modifying, testing, validating and using models and components, there are a number of necessary activities which could be simplified with the aid of computerized routines. For example, programs and routines can be used to conduct sensitivity tests and/or Monte Carlo analyses on model parameters; perform spectral analyses of simulation-generated data; track the model against recorded time series; estimate model parameters with regression analyses on time series data, or with the use of a parameter optimization package [5]; and facilitate the use of models for policy analysis with an interactive, conversational language system. Such routines will be useful to users in modifying and specializing library components and models for particular applications and to model builders in developing new models or updating and testing old ones.

Simulation Language

To facilitate user access to models constructed from library components, CLASS has developed a FORTRAN-based simulation language [6,7] as part of developmental research in support of the library concept. The language enables modelers to write a program which interfaces a model with users through an interactive "conversation," thus allowing the policy analyst to sit down at a computer terminal and enter parameter values to "tell" the model what he wants to do with it, without the aid of a computer programmer. For example, he can specify whether he wants to perform a policy experiment or sensitivity tests; if a policy experiment, what policy combinations and levels (e.g., budget levels, tax rates, etc.); what endogenous variables to display at what intervals and in what format (e.g., graphs, histograms, etc.); and the time period to simulate and whether he wants to interact and possibly change policies in the course of simulation or to see results only at the end.

A second phase of the simulation language development will focus on facilitating the job of the model builder/programmer in linking up library components into the particular configuration desired as a model for the specific application at hand.

Scope of CLASS Activities

The primary charge of the software library is the stocking, maintenance, and distribution of the libraries of components, models, and routines. In

performing these functions, CLASS is responsible for acquiring and testing (discussed below), potential library software, insuring that documentations meet specified standards and distributing documentations and software upon request. Distribution procedures and the formats of various kinds of documentations will be discussed later in this report.

In acquiring components, models, and routines (hereinafter collectively called software) for a particular slot in the library--e.g., a resource allocation component or a validation routine--it will be desirable to actively search out and test alternative candidates. One or more will then be chosen to become part of the library, the appropriate set of documentations will then be provided, and users will be notified of the changes in, deletions from, or additions to the library. Of course, software entering the library must meet the CLASS criteria for inclusion, discussed in the next section.

The other side of CLASS activities is servicing users with library software as requested. As a bare minimum, CLASS publicizes and distributes software and associated documentations and keeps a file of user's feedback; i.e., reports, results, and specialized models and components.

More broadly, however, CLASS could offer consultancy and training services both in support of and supported by its distribution activities. Consultants could be supplied--from CLASS itself or by referral to outside expertise--in conjunction with the use of library software. For example, if FAO were investigating agricultural marketing in India, say, a CLASS consultant (or a team) would participate on site in performing, first a system and problem definition study. He would then identify components

and/or models available from the library which would apply, as well as necessary components the library might not have and which would have to be built. The CLASS consultant would assist in modifying, specializing, and merging the generalized software to operate in the particular context of agricultural marketing in India. This would involve model testing and validation using routines from the library, where available. Finally, he would help define and run policy alternatives and interpret the simulated results. Any given application would require any, all, or none of the above consultancy services, depending on the available manpower within FAO (in this case) itself.

If FAO, AID, or national or local governments wished to develop that manpower, CLASS could perform or participate in a training role. For example, if AID wishes to train either its own personnel or, as part of a high-level extension program, foreign nationals, CLASS could supply its models, components, and routines as instructional materials. In return, the library could benefit from class projects arising out of the training program. Further discussion of the consultancy and training functions will be given as part of a later section on the organization and procedures of CLASS.

CLASS Standards

If CLASS is to operate effectively and efficiently in carrying out its obligations of maintenance and distribution; and if the software is to be suitable to the successful building of models, then it is essential that a set of standards be defined and maintained. These standards pertain to the admissibility and continued inclusion of library software. The software includes models and components designed specifically to simulate

processes and/or address problems related to the library's area of application and utility routines which perform generally useful mathematical or logical operations. Each software element is basically an abstract concept which is operationally realized in the library as computer programs and written documentations. Standards are set separately for the selection of software, for the documentations and for the computer programs.

Software Selection Standards

Software are evaluated for admission to, and continued inclusion in, the library in three broad respects: general applicability, theoretical and empirical validity, and compatibility with other library software. Since quantitative or objective criteria cannot be set for some of these concepts, subjective evaluations often have to be made.

General Applicability

We discussed above the problems of defining and measuring "generality." While understanding this difficulty, library components are expected to be of general interest and applicability in the library's areas ofapplication, i.e., agricultural development planning and policy analysis. That is, specialized models designed to simulate processes and/or address problems unique to a particular country or region within a country will not be included in the library. Models built for specific countries or regions will be eligible for admission to the library after reworking to generalize their structure and programming and otherwise be made to meet the library's standards and if the processes modeled or problems addressed are judged to be of broader interest than that for which the models were originally specifically written.

Similarly, utility routines will be included if they are judged to be

of general usefulness. These may include matrix operations, simplex algorithms, gradient search procedures, distributed and discrete delays, table look-up functions, plot routines, etc.

Theoretical and Empirical Validity

Generalized models cannot be validated to the extent of faithfully representing a particular real-world system. Such a validation can only be made once the models have been adapted in the context of a specific application with its particular data. Rather, they are idealized representations of reality whose validity can only be determined with respect to internal logical consistency and conformity with accepted social and economic theory.

Compatibility

Since components and routines will be combined in a variety of configurations, as required for specific applications, library software are expected to be compatible with one another. Compatibility is insured and facilitated by the use of a common programming language, a standard programming style, clearly specified and compatible (as to definitions and formats) inputs and outputs, and documentation illustrating how software combinations may be made. Programming and documentation standards are discussed in more detail below.

Documentation Standards

CLASS maintains three types of documents for each software element it inventories: announcement, abstract, and documentation. Format and content standards for each of these will be discussed separately following some general considerations.

General

Standard identifying information appears on the first page of each of the three document types for an element as in the sample in Appendix C. In addition, the first page of the announcement and of the abstract contains the body of the text. The first page of the documentation contains a table of contents following the identifying information, with the body of the documentation text beginning on the second page.

In order to assure completeness, accuracy and clarity, at least two reviewers check each software element's set of documents. A computer-oriented reviewer evaluates primarily the documentation and the computer program itself in terms of:

(a) the accuracy and efficiency of the program as a computer model of the mathematical model described in the documentation; and

(b) the sufficiency of the documentation for implementing the computer program.

The other reviewer, with expertise in the subject areas (disciplines) covering the problem or process addressed by the software, deals with:

(a) the theoretical and empirical validity of the mathematical model described in the documentation and the suitability of the component as a whole, in terms of its intended applications; and

(b) the sufficiency of the documentation for understanding the software and its intended applications (e.g., the completeness of the discussion of implicit and explicit assumptions of the model) and as a basis for deciding the suitability of the element for particular applications.

Announcement

Announcements are published and distributed--for example in a newsletter--to actual and potential library users as a means of publicizing what CLASS has to offer. Thus, in most cases it will be on the basis of announcements that a user will decide, at least initially, whether certain software may

be useful to him and, hence, whether to seek further information by requesting the abstract, documentation and/or consultation with the library staff.

An announcement, therefore, while it must be brief (a short paragraph), must include enough information so the above decision may be made. As a minimum, an announcement must give a verbal, non-technical statement of what the software does, i.e., the problem or process simulated by the model or the utility function or algorithm performed by the routine, including any special features of the software which may distinguish it from other possible models of the same process or similar processes. In addition, examples may be given of possible uses of the software, e.g., larger models into which it may be embedded.

Abstract

Abstracts are intended to provide technical experts, analysts and advisors with brief technical descriptions of CLASS software for use in preliminary evaluation and selection. In addition to a more technical presentation, where appropriate, of the information contained in the announcement, an abstract states the inputs required by the software, the outputs it supplies, and the core storage and sample run execution time for the computer on which it was tested.

Documentation

There are actually two audiences for which documentations are intended: consultants and analysts, and programmers. One effective way of meeting the needs of these distinct audiences is to compartmentalize the

documentation in three clearly identifiable sections. The first section includes much explanatory material which would be required in order to understand what the software does and where it may properly be used; the second goes deeper into the theoretical and mathematical detail required by the professional analyst; and the third provides the information a computer programmer will need in order to successfully implement the program at the local computer installation. The documentation will be essential in (1) selecting appropriate software elements for a particular application, (2) validating models composed of a number of components, (3) analyzing the results of simulation experiments, (4) operationalizing software on a local computer in the context of a larger program comprising a model, and (5) making modifications to software as necessary to suit a specific need.

The contents and formats of each of the three sections of the documentation and an illustrative set of documents for one of the software elements of CLASS, a demography component, are included in the Software Standards Manual (CLASS Document No. 2, July, 1976).

Programming Standards

The objectives of programming standards are to

(1) maintain a consistent programming style
(2) maintain a compatibility among computer programs
(3) ensure and facilitate adequate error checking
(4) facilitate further developments
(5) enhance readability
(6) ensure as much machine independence as possible

These objectives are set with the belief that their attainment will resolve a large number of problems which might otherwise be encountered by users of CLASS software elements.

In general, CLASS standards are compatible with ANSI standards. Detailed programming standards appear in the Software Standards Manual.

Procedures and Organization

CLASS functions can be summarized in three areas: 1) the acquisition and development of components, models and routines with associated software and documentations; 2) storage and maintenance of these software elements; and 3) user-related services such as software retrieval, consulting and training. In this section, we discuss procedures and an organizational structure for carrying out these functions.

Procedures

Acquisition and Development

Models, components, and routines are acquired by the library in either of two ways--passively or actively.

Passively, elements may be acquired from other researchers and modelers who submit their work to CLASS for possible inclusion and distribution. In this case, CLASS reviews the software first with respect to its potential applicability and usefulness for agricultural modeling and policy analysis. If it is found to have a place in the library, CLASS makes the necessary modifications in the model, the program and the documentation to bring them up to the CLASS standards (discussed earlier), and then the software is admitted to the library and publicized for distribution to users.

This process may cause a substantial time lag between the initial submission of a software element to the library and its final acceptance and publication. In such cases, CLASS will consider it for conditional distribution pending completion of the process once it has been judged of sufficient

applicability and usefulness for CLASS purposes to warrant distribution before being brought up to CLASS standards.

There are two modes in which CLASS actively acquires software. Both begin with a determination of a need for a certain kind of problem-oriented model, function-oriented component or technical routine in the CLASS repertoire. In one mode, CLASS actively seeks out or solicits candidates from the literature, from ongoing research projects, and from public and private institutions which use models. If potentially suitable candidates are found, one or more is selected for review and standardization prior to final acceptance and release for distribution.

Alternatively, or if the above active search is unsuccessful, CLASS may develop the needed software internally. In this case, model development, programming and documentation will conform with CLASS standards from the beginning.

Storage and Maintenance

Between the acquisition and utilization of software elements lies the functions of storage and maintenance.

Library software--models, components, and routines--exist in three forms: an abstract conceptualization (e.g., the mathematical model), a computer program, and a set of documents. (As we saw earlier, CLASS maintains standards for each form.) Physically, the software elements exist in the latter two forms, and these are the concerns of the storage function. The media for storage are magnetic tapes and listings for the programs and duplicated typescript for documents.

Beyond the passive storage of software, the library must actively maintain each of the three forms of a software element's existence--model, program and documentation. Here, maintenance refers to making corrections

and improvements in programs and documents as indicated by users. In addition, in order to maintain CLASS at a high level of usefulness, each software element is reviewed from time to time and, if necessary, is revised or removed from active service. By this process, the abstract models, programs and documentations are kept up to date with regard to both user needs and technical knowledge.

User Services and Communications

As we have discussed earlier, CLASS user services range from merely announcing the availability of software and distributing them upon request to full consulting with users on the selection and use of library software appropriate for their particular needs to training user personnel in the building and use of analytical models.

Effective communication with users is essential to the success of these activities. Three types of communication are necessary: primary communication to disseminate information on the library's offerings of software and services; secondary communication to enable users to select and implement library software and services; and tertiary communication as feedback to CLASS on user experiences and results, including reports and specialized models and components.

Primary communication consists of a descriptive index (for example, see Appendix A) of library models, components, routines and services, periodically updated and sent to past, present, and potential users. The index could also be published from time to time in appropriate professional journals and newsletters. Normally included with the index is a standardized order form for ordering the various kinds of secondary communication on items described in the index.

Secondary communication consists of the documentation on each of the software elements in the library. These are necessary for evaluation and selection of software for a particular application and as reference material for modifying and linking software elements as required for the application.

Finally, tertiary communication involves feedback to CLASS from users. CLASS solicits comments and suggestions from users concerning its own operation as well as the usefulness of the models, components, and routines themselves. In addition, a file of reports will be kept and a library of specialized models and components will be created from user-modified versions of its own generalized software. In this way, deficiencies in current software, as well as the need for new software, can be identified; and later applications may use specialized models and components, where appropriate, which have already been built rather than start from scratch or from generalized models and components.

Organization

The organizational structure of CLASS which carries out the above functions and procedures is shown in Figure 1.

The clerical staff communicates with and sends distribution packages to users, keeps files of such communications, and prepares and keeps files of the library documentations. It is essential, if CLASS is to remain effective and relevant to users, to keep records of software which users have received from the library, applications and modifications of that software and reports of user experiences. In addition, as new and improved versions

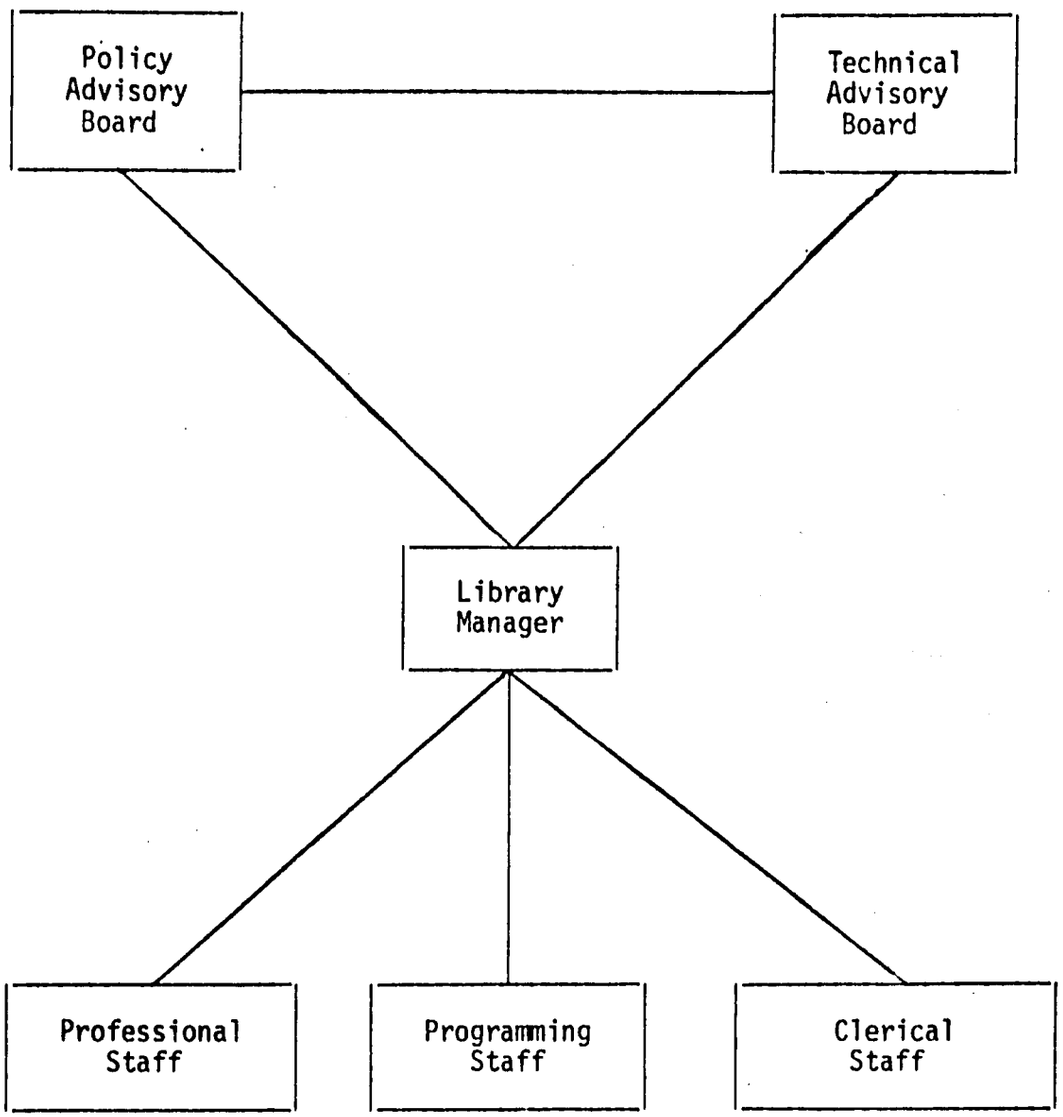

FIGURE 1

Organizational Structure of CLASS

of library software become available, such records will facilitate sending the new versions to users who have the old ones.

The CLASS professional staff includes both temporary, part-time consultants, e.g., economists, sociologists, ecologists, systems scientists, computer scientists, etc.; and permanent, full-time computer programmers. Disciplinary professionals are used on a part-time basis as needed to meet consultancy and training demands and to participate in the researching and testing of software to be acquired by the library. They contribute important to the documentation of acquired or updated software which they have researcl and tested. A computer programmer is necessary to reprogram acquired software in the library's standard format, to develop and operate the computerized system of library maintenance and information retrieval, and to prepare distribution packages for users. He is also responsible for the programmer-oriented documentation of each software element. Part-time editorial services may also be desired to review the professionally written documentations and to assure that each documentation is comprehensible to the level of expertise of its intended audience.

The library manager is responsible for coordinating the library's activities and maintaining quality control over the models, components and routines in the library and their associated documentations. He identifies needed software or modifications of existing software and oversees (and perhaps participates in) the researching, testing, reprogramming, and documenting of them. He communicates with users and negotiates with them as to the library services they will use, e.g., consulting or training as well as the use of library software.

In view of the fact that the library's objective is to provide a unique service for the agricultural development field, it is essential that potential users and other experienced professionals have an input into its development and operationalization. Formally, this input is channeled through policy and technical advisory boards whose functions, in general, are to review, evaluate, advise, and recommend.

The specific charge given the Policy Advisory Board is to

1. Evaluate the quality and coverage of library models, components, and routines, and recommend where existing software should be revised and where further modeling is needed.

2. Review and evaluate the library's scope and operations as to the level of usage and effectiveness of its software and services and recommend modifications of services and operating procedures as necessary for improvement.

3. Recommend and evaluate alternative means of disseminating information to users and potential users on CLASS software and services.

4. Based on the performance of CLASS while at MSU, consider and advise on the library's ultimate geographic and institutional location and means of financial support.

The CLASS Policy Advisory Board is constituted and, to date, has held one meeting at MSU. Its membership includes representatives of FAO, AID, USDA, and the international academic community.

A technical advisory board has not yet been formed. Its functions would be to review, evaluate, and make recommendations concerning the technical aspects of CLASS activities--mathematical modeling; documentation; computer operations such as programming, storage and retrieval, and perhaps computerized text writing for documentation; and training and consulting services.

Present Status and Utilization of CLASS

The present status of CLASS can be discussed in terms of the software it contains, its organization, and utilization of its software and services. Its current inventory of software is summarized in Appendix A. Organizationally, CLASS has been a part of the Agricultural Sector Analysis and Simulation Projects at MSU, with a part-time manager, a part-time secretary, and one part-time and one full-time programmer. Contacts have been made with potential users in AID, USDA, and elsewhere; and a proposal has been submitted to IIASA for establishing a similar library there [9]. In the remainder of this section we discuss utilization of CLASS to date.*

A primary consideration guiding acquisition of software elements for CLASS has been their potential usefulness in modeling work. The delay, table functions, demographic components, and Policy Analysis Language (PAL) have been used in several research projects at MSU, and CLASS software has been sent to other locations. Following is a description of their uses in research at MSU.

Posada [10] used delay, table, and demography routines that were very similar to those in the Korean agricultural sector model and the Nigerian simulation model. This fact contributed to the decision to include these components in the library in addition to providing inputs to their design specifications. In addition, the PAL package was later applied to the Posada model.

The development of a Michigan Agricultural Sector Simulation model provided a preliminary test of seven CLASS software elements: both demographic components, two table routines, two delay routines, the accounting component, and the Policy Analysis Language. This model was demonstrated at the Policy Advisory Board meeting in May, 1974. This test

*The summary of utilization to date was compiled by David L. Watt.

guided additional revisions to the various components, models and the Policy Analysis Language.

Components and routines of CLASS have been used in several doctoral research projects. Lee [11] projected technological change in Korean agriculture. In this project delay routines were used extensively for lags in the acceptance of innovation, and table functions were used for the allocation of resources to education and extension work for the diffusion of innovations. Nweke [12] in his model of Nigerian forestry demand, used distributed delay routines to model the replacement needs for wooden structures, table functions for tracing projections of economic variables not amenable to simple algebraic equations, and demography components for population modeling. Additionally, table routines were used by Watt in his dissertation research [13] developing a production component for the Michigan Agricultural Sector Study. Delay routines, table functions, and demographic components were used in Jaske's dissertation work [14] on livestock enterprise decision making.

PAL has been used in conjunction with two national agricultural models of the USDA Economic Projections group. The first was a national framework model of the agricultural sector; the second is an aggregate farm production model.

Research projects currently underway using CLASS library components include a fisheries model for commercial fisheries in Michigan and a rubber industry model. Several other dissertation projects in process have familiarity with the computer library and a strong potential for using them. These include the research on land use policy for flood plains and on projecting financial input demands in U. S. agriculture.

Future Activities

Resource permitting, the following activities will be pursued in order to enhance the utility and effectiveness of CLASS by its potential users.

Announcement

CLASS has not yet been formally announced. The first step is to prepare and distribute announcements of the existence of CLASS to journals, departments of agricultural economics, research institutes, and others who might be interested. It may also be useful to prepare an audio-visual demonstration of the use of the library. This demonstration would illustrate how the components of the library could be combined into a model under the control of the Policy Analysis Language (PAL) in order to address specific problem issues. The user-oriented features of PAL would be stressed.

Acquisition

In order for the library to serve as a useful clearinghouse for agricultural simulation modeling work being done around the world, it will be necessary to identify and actively solicit components from known models, such as the Korean agricultural sector model of MSU, the Thailand LP model being developed under a USAID contract with Iowa State University, the AGRIMOD model of the U. S. agricultural sector developed under an NSF contract to System Control, Inc., of Palo Alto, the NIRAP model of U. S. agriculture being developed by USDA, and others. A considerable effort will be required to identify, acquire, and evaluate the general usefulness of various models and components for inclusion in CLASS. Once acquired, models must then be tested and perhaps the programs

and documentation reworked to meet CLASS standards.

Advisory Boards

Further meetings of the Policy Advisory Board will be held to review the status and operation of CLASS, to set priorities for the acquisition and reworking of models and components to meet CLASS standards, and to determine future directions. Also, the Technical Advisory Board will be constituted.

Operation of CLASS

The types of procedures required to operate the library have been outlined above. However, these procedures must be put into actual operation. Therefore, some minimum programming effort must be spent in further developing storage and retrieval programs and procedures to handle the library operation. In addition, a rate schedule must be established for charging users for services provided by the library in order eventually to make the library a self-sustaining operation.

Communications

Other software libraries have found it useful to establish regular communications with potential contributors and users of the library by means of a newsletter. CLASS will plan to do the same, probably on a semi-annual basis to begin with. The newsletter will carry information on new acquisitions, state-of-the-art summaries, technical tips on how to utilize components from the library to address specific problem issues, etc. In addition, an updated index to the library holdings will be distributed with each edition of the newsletter.

Consulting

The staff of the library will provide, as requested, technical assistance on utilizing the resources of the library and will also be available to participate in special training seminars on the application of the system simulation approach to agricultural analysis. Users would pay travel and salary costs if full-time consulting were required for a period of time.

Cooperation with Other Institutions

CLASS will cooperate with other institutions interested in setting up their own versions of CLASS or in becoming a more permanent home for CLASS itself. Contact has already been made with AID and IIASA in this regard.

Developmental Research

As resources allow, further research will be conducted in what is required to make the library software really useful as policy analysis tools to governmental and donor agency analytical units. Possible areas for further development include improved output via CRT terminals, plotters, etc.; the linkage of models to data base management systems; and the use of structured programming, top-down program design, and other improved programming and documentation techniques. The Policy and Technical Advisory Boards will provide guidance in establishing priorities and procedures for developmental research.

References

[1] Manetsch, Thomas J., et al., A Generalized Simulation Approach to Agricultural Sector Analysis With Special Reference to Nigeria. Final Report to USAID (East Lansing: Michigan State University, 1971).

[2] Rossmiller, George E., et al., Korean Agricultural Sector Analysis and Recommended Development Strategies, 1971-1985. AERI-MAF, Seoul, Korea and Department of Agricultural Economics, Michigan State University, East Lansing, Michigan, 1972.

[3] Miller, S. F. and A. N. Halter. "Systems-Simulation in a Practical Policy-Making Setting: The Venezuelan Cattle Industry." American Journal of Agricultural Economics. Vol. 55, No. 3, August, 1973, pp. 420-432.

[4] Petit, Michel. "Summary of the Simulation Library Working Conference, March 29-30, 1973." East Lansing: Michigan State University, March 1973.

[5] Buchner, Marcus Robert. "An Interactive Optimization Component for Solving Parameter Estimation and Policy Decision Problems in Large Dynamic Models." Unpublished Ph.D. dissertation, Michigan State University, 1975.

[6] Manetsch, T. J. "A Simulation Language to Facilitate the Development and Use of Simulation Models in Policy, Program and Project Analysis--Preliminary Outlines." Project Working Paper 73-1. March 1973.

[7] Winer, Claudia, and Wolf, Chris, "Technical Documentation for the Policy Analysis Language," Agricultural Sector Analysis and Simulation Project, June 1976.

[8] Abkin, M. H. and D. W. Pervis. "Draft Proposal for an Agricultural Sector Simulation Library." Project Paper. October 1972.

[9] Abkin, Michael H. "Draft Proposal for an International Software Library for Applied Systems Analysis." East Lansing: Michigan State University, October 1975.

[10] Posada, Alvaro. "A Simulation Analysis of Policies for the Northern Colombia Beef Cattle Industry." Unpublished Ph.D. Dissertation, Department of Agricultural Economics, Michigan State University, 1974.

[11] Lee, Jeung-Han. "Projections of Product Supply and Factor Demand under Structural Change for Korean Agriculture: A Systems Simulation Approach." Unpublished Ph.D. dissertation, Michigan State University, 1974.

[12] Nweke, Felix Izu. "A Systems Analysis and Simulation Study of Nigerian Forestry Sector: Wood Consumption Component." Unpublished Ph. D. dissertation, Michigan State University, 1975.

[13] Watt, David L. "Michigan's Agricultural Production." Unpublished Ph. D. dissertation, Michigan State University, 1976.

[14] Jaske, MIchael R. "System Simulation Modeling of a Beef Cattle Enterprise to Investigate Management Decision Making Strategies." Unpublished Ph. D. dissertation, Michigan State University, 1976.

APPENDIX

Index to CLASS Software

INDEX OF SOFTWARE

COMPUTER LIBRARY FOR AGRICULTURAL SYSTEMS SIMULATION

Agricultural Sector Analysis and Simulation Projects
Supported under Contract No. AID/csd-2975
U. S. Agency for International Development

Department of Agricultural Economics
Michigan State University
East Lansing, Michigan

July, 1976

TABLE OF CONTENTS

	Page
INTRODUCTION	1
MODELS OF AGRICULTURAL SYSTEMS	2
COMPONENTS FOR AGRICULTURAL SYSTEMS ANALYSIS	8
DEMOGRAPHIC COMPONENTS	9
SPECIALIZED TECHNIQUES	10
SIMULATION LANGUAGES	14
UTILITY ROUTINES	15

INTRODUCTION

This document contains a index to computer program components currently contained in the Computer Library for Agricultural Systems Simulation (CLASS).*

For each software item inventoried, this index gives a title, authors, a brief description, the compilers used and an approximate card count, the computer on which the software has been run, documentation references, and in some cases special comments.

Documentation and source code for the programs listed in this index may be obtained from:

Computer Library for Agricultural Systems Simulation
Department of Agricultural Economics
Michigan State University
East Lansing, Michigan 48824 USA

Requests for source code should include format specifications for punched cards or magnetic tapes.

---

*Work under Contract No. AID/csd-2975 also included development of the Korean Agricultural Sector Model (KASM) and the Korean Grain Management Program Model (GMP). The components of the Korean Agricultural Sector Model (KASM) include: technology change component, resource allocation and production component, demand/price/trade component, national economy component, population component. When the documentations for these models and components are completed, they will be inventoried in a future index.

MODELS OF AGRICULTURAL SYSTEMS

AGM-1. Northern Nigeria Beef Industry Model

Glenn L. Johnson et al, Michigan State University, East Lansing

In the model, the livestock population of Northern Nigeria is disaggregated into two populations--one traditional and one managed using modern techniques. The "traditional" cattle population (belonging to nomadic Fulanis) is assumed to subsist on the tsetse-free grazing land of Northern Nigeria during the wet season. During the dry season, crop residues and additional grazing land (which becomes available as the habitat of the fly recedes southward during dry months) also add to the nutrient supply. Grazing land and, hence, total digestible nutrients (TDN) for the traditional population are endogenous model variables which are related to any food crop acreage expansion to feed a growing human population, to expansion of cash and animal feed crop acreages, and to increased tsetse-free grazing land areas as a result of fly eradication and grazing reserve programs. Expenditures on fly eradication and grazing reserve programs are policy variables which can be varied during model tests.

The "condition" of the traditional grazing land, which influences grass yields per acre, is computed as a function of the number of animals grazing on an area relative to an equilibrium number. Male and female livestock populations and sales are computed as functions of calving rates, death rates and marketing strategies, the latter being policy variables which may be varied during model tests. Calving rates and death rates are endogenous model variables determined by the per-animal digestible nutrients supplied by grazing land and crop residues. Animals in the "modern" sector are assumed to be situated on grazing reserves where adequate nutrition is available from properly managed grassland (range productivity in the modern sector is not deteriorating) and supplemental feed obtained from cropland is devoted specifically to animal feed production. Land allocated to animal feed crops is a policy variable which can be substituted for land allocated to cash crops (another policy variable). Range land available for "modern" grazing is determined by expenditures on grazing reserve programs. Male and female population sizes and sales, calving rates and death rates are computed using virtually the same functional relationships but different input levels compared to the traditional sector.

Several alternative criteria functions which might influence a policy-maker's choice of development programs are calculated at both interim and final stages of a simulation experiment. Farm level incomes derived from meat, milk and cash crops are computed, as are capital investment and operating costs incurred through implementing various modernization policies being explored in successive simulation runs. Thus,

several relative benefit/cost relationships for various modernization policies are summarized by computed criterion function which include discounted net cash flow, foreign exchange balance, farm incomes, net beef imports and domestically produced nutrient outputs. The ability of the industry to meet the (forecasted) increase in demand for beef is determined by computing imports necessary to satisfy the projected demand.

Compiler: FORTRAN

Computer: CDC 3600

Documentation: G. L. Johnson et al. "A Simulation Model of the Nigerian Agricultural Economy: Phase I--The Northern Nigerian Beef Industry." A progress report to the Agency for International Development submitted under the auspices of Contract No. AID/csd-1557, Agricultural Sector Models, by Michigan State University, April 28, 1968.

Comment: The programming and documentation do not meet CLASS standards as described in CLASS Document No. 2. The model is included because it is historically important. It was the precursor to the Venzuelan Beef Cattle model which was developed by S. F. Miller and A. N. Halter ["System in a Practical Policy-Making Setting: The Venezuelan Cattle Industry," American Journal of Agricultural Economics, Vol. 55, No. 3, August, 1973, pp. 420-422] and utilized for policy analysis in Venzuela, but which is not yet inventoried in CLASS.

AGM-2. Nigerian Agricultural Sector Simulation Model

Thomas J. Manetsch *et al.*, Michigan State University, East Lansing

Since agriculture contains most of the productive resources in Nigeria (contributing 65 percent of the gross domestic product (GDP) and 66 percent of Nigerian exports in 1962-63) and in most less-developed countries, its role in future growth will be very important. In specifying the model, concepts of relevant policy-making clientele and their problems determined which sectors and interrelationships needed particular attention within the model and the level of aggregation required.

Many planners in the less-developed countries are interested in evaluating alternative projects, programs and policies affecting regional specialization of production and trade. These typically involve farmer responses to various economic incentives or government assistance projects, etc. The model has a commodity orientation, emphasizing export crops. To permit considerations of simple questions related to regional specialization and interregional trade, a two region (North and South) model was conceived. However, several ecological zones within each region were also differentiated to permit more detailed consideration of problems encountered within the two regions. Although the model is based on the Nigerian environment, its orientation toward both annual and perennial commodities with distinct ecological zones and regions makes its components adaptable and applicable to a broad range of countries in accordance with the objectives of the AID contract under which the work was done.

The Northern, Southern and nonagricultural submodels are each shown as two interacting parts: (1) demography and (2) production, marketing and consumption. These parts are linked together by income flows, labor supply and the demands and supplies of subsistence foods or, in the case of the nonagricultural submodel, labor supply and flows of consumer goods. The Northern and Southern submodels are linked directly with the non-agricultural submodel through the flow of consumer goods, raw materials for manufacturing and agricultural producer's inputs supplied by the nonagricultural submodel. Labor migration can take place between the agricultural and nonagricultural submodels. The major interaction among the three submodels takes place through interregional trade in food which is simulated by a market and interregional trade component.

Submodel of Northern Nigeria

The Northern model consists of six interacting components. The cattle production component simulates the meat and milk production process in traditional and modern herd management situations, using inputs of total digestible nutrients (TDN) from the production of various forage and grain crops. The main interaction between the cattle and annual crop

components in the Northern model is the land allocation component where the acreage in various crops partially determines the quantities of total digestible nutrients available in the cattle component.

The agricultural production and marketing component simulates the production and marketing activities for groundnuts, cotton and food. Land use is determined by the land allocation component, prices for food by the market component and yields by the modernization component. In turn, the production and marketing component computes average returns to land and labor, which affect farmer's land allocation decisions. The value added at each stage of production and distribution is also computed in the production and marketing component and utilized in calculating the national accounts in the nonagricultural component.

In addition, the market component of the total model simulates the price mechanism of the cash food market in each of the regional models. This component traces the shifts in the demand for food and utilizes the calculated food supply to determine the price of food.

The consumption and budget component computes a number of agricultural sector variables needed by the nonagricultural model. There are expenditures on chemical inputs, capital goods and consumer goods derived from the calculated income and population levels in the North. The component also computes disposable incomes from production and marketing, for use in the nonagricultural model, and per capita income by region in the North.

Submodel of Southern Nigeria

The Southern model is composed of five interacting components. The agricultural production, marketing and processing component computes the production from acreages of traditional cocoa, modern cocoa, traditional palm, modern palm, traditional rubber, modern rubber, food and tobacco by simulating commodity yields and food subsistence levels of the agricultural population. Marketing and processing the modeled, using accounting equations based upon fixed relationships related to the product. Input demands--labor, capital, chemical and biological materials--are calculated for the functions of production, processing and marketing. These variables provide the main interaction with the nonagricultural component of the national model.

The land allocation and modernization component of the Southern model simulates farmers' allocation of land to the traditional or modern production of cocoa, palm, rubber, tobacco and food based upon economic, physical and cultural factors.

The third component of the Southern model generates world, market processor and producer prices for the five commodities considered in the previous two components. Two additional components provide further interaction with other components of the national model. The allocation

of the agricultural modernization budget (from the national budget) is made by the modernization program and policies component. The accounting component of the Southern model computes values for the performance variables for the Southern model as a whole, and provides inputs to the national account section of the nonagricultural model.

Submodel of the Nonagricultural Sector

The nonagricultural model has a dual purpose within the national model. In a broad, rudimentary way it models the nonagricultural components of the economy, enabling key interactions between agriculture and nonagriculture to be studied. The nonagricultural model simulates the demand for food by the nonagricultural population and manufacturing demand for agricultural raw materials. Likewise the model simulates the supply of agricultural inputs, such as chemical materials and fertilizer, and the supply of consumer goods and services to the agricultural population.

The nonagricultural model also summarizes the accounting variables of both the agricultural models and the nonagricultural sector, and constructs national accounts and balance of trade tables. These include measures of gross national (domestic) product by branch of activity and category of expenditure, government revenues, and import-export balances.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 16,000 symbolic cards

Computer: CDC 6000

Documentation: Thomas J. Manetsch et al. A Generalized Simulation Approach to Agricultural Sector Analysis (With Special Reference to Nigeria), Michigan State University, East Lansing, November 30, 1971.

Gloria Page, Mark Turnquist, and T. W. Carroll. "Technical Reference Manual for the Nigerian Simulation Model, Version 7," Agricultural Sector Analysis and Simulation Projects, Michigan State University, East Lansing, November 1, 1971.

Comment: The programming and documentation of this model do not meet CLASS standards as described in CLASS Document No. 2. The model is included in CLASS because it has been used for policy analysis and is historically important.

AGM-3. Beef Cattle Enterprise Simulation Model

Michael R. Jaske, Michigan State University, East Lansing

The model simulates the behavior of the major physical and financial variables of a general beef cattle enterprise through time. Although the model is intended for general use, attention has been concentrated on modeling of the land extensive cow/calf range operation. The model consists of five major components and several secondary components. Major components are (1) cattle demography, (2) forage growth, (3) feed stock accounting, (4) nutrient impacts on growth and reproduction, and (5) management decision making. The financial component is the most important component of the secondary group. Land allocation among alternative uses and crop production are not modeled and are arbitarily specified by the user.

A highly disaggregated herd population model allows full exploration of population dynamics. Using TDN at the basic feed quality descriptor, and net energy for growth and maintenance where possible, the nutrient impact component predicts rates of weight gain and reproductive dynamics as a result of feed intake. Forage growth is modeled as a function of the exogenous weather variables of solar radiation, temperature, and rainfall. The financial component determines revenue and cost as a result of physical events and these determine the effect on profit, cash flow, depreciation, taxation, and debt levels. The management component has been developed to provide for the needs of managers is ongoing operating decisions as well as investment planning. Detailed control variables allow realistic simulation of events of interest to managers. The interactive structure of the component, with straight forward decisions made endogenously following standard economic criteria and more complex decisions made exogenously by the model user, is implemented to allow user control at appropriate decision points.

Three examples of management strategies are evaluated (in the thesis) to demonstrate the capability of the model to assist decision makers. These are (1) early versus late calf weaning, (2) the rate of cattle herd development in investment projects, and (3) general profit maximization. These examples are discussed in terms of financial criteria.

Compiler: CDC FORTRAN Extended, 6000 Version 4, 7,500 symbolic cards

Computer: CDC 6000

Documentation: Michael R. Jaske, "Systems Simulation Modeling of a Beef Cattle Enterprise to Investigate Management Decision Making Strategies." Unpublished Ph.D. dissertation, Michigan State Unviersity, 1976.

Michael R. Jaske, "User's Guide for the Beef Cattle Enterprise Simulation Model," CLASS Document No. 11, July, 1976.

COMPONENTS FOR AGRICULTURAL SYSTEMS ANALYSIS

AGC-1. AGACC: Accounting Routine for the Agricultural Sector

D. Pervis and C. Wolf, Michigan State University, East Lansing

The purpose of AGACC is to perform the many computations involved in the aggregation and accounting functions necessary in most models. AGACC is a subroutine subprogram to be invoked at the end of each time iteration during the execution of a model. No integration over time is done in AGACC. The data requirements of AGACC include model size parameters, policy-determined coefficients, technology-determined coefficients and model variables. The output consists of a set of enterprise, regional and national data.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 160 source statements (ANSI subset).

Computer: CDC 6000

Documentation: CLASS Document No. 10, July, 1976.

DEMOGRAPHIC COMPONENTS

DEM-1. DEMOGC: Demography with Distributed Age Cohorts

M. H. Abkin and C. Wolf, Michigan State University, East Lansing

DEMOGC models the demographic cohort-survival processes of birth, death, migration and aging for a population disaggregated by age and sex. The length of a cohort is an average of the time individuals take to pass through the cohort; i.e., cohort staying times vary among individuals in the population. DEMOGC may be used, for example, for populations of trees, capital equipment, livestock, insects and humans--where "age" cohorts may be defined by such age-related characteristics as yield, reliability, fertility, and use of social services, respectively.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 90 source statements (ANSI subset)

Computer: CDC 6000

Documentation: CLASS Document No. 6, July, 1976.

DEM-2. DEMOGD: Demography with Discrete Age Cohorts

M. H. Abkin and C. Wolf, Michigan State University, East Lansing

DEMOGD models the demographic cohort-survival processes of birth, death, migration and aging for a population disaggregated by age and sex. Each cohort is of fixed length so that all individuals in the population take the same time to pass through it; all cohorts need not be of the same length, however, DEMOGD may be used for example, for populations of humans, capital equipment and students--where "age" cohorts may be defined by chronological age, vintage, and class year, respectively.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 80 source statements (ANSI subset)

Computer: CDC 6000

Documentation: CLASS Document No. 7, July, 1976.

SPECIALIZED TECHNIQUES

SPT-1. PERT: Program Evaluation and Review Technique

Adapted by Martin Hanratty, Michigan State University, East Lansing

The Program Evaluation and Review Technique (PERT) is a method of planning, replanning, and evaluating project implementation. The technique provides project administrators and planners with information which can be used to monitor and adjust a planned or ongoing project. It is helpful in insuring the attainment of project goals within a minimum time period. The technique was developed during the late 50's at the Navy's Special Projects Office by a team studying the application of statistical and mathematical methods to the planning, evaluation and control of the research and development of new defense systems, the first one being the Polaris missile. Since its inception PERT has undergone continuous development and been applied to a wide variety of projects, both public and private.

The technique provides a number of positive benefits to the project analyst by:

(1) forcing the analyst to develop a detailed outline of the project which might otherwise be done poorly or not at all,
(2) requiring the construction of a network diagram where each project is depicted and its relationship to all others is expressed,
(3) characterizing projects as "critical" or "non-critical" in terms of the overall project phasing,
(4) allowing managers to simulate the effects of alternative decisions under consideration and to study their effects upon project deadlines prior to implementation,
(5) providing important timing information which can assist management in conducting further evaluation, as for example, cost-benefit analysis,
(6) by providing relative benefits which increase proportionally with project scale and complexity.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 750 symbolic cards

Computer: CDC 6000

Documentation: Agricultural Economics Report 290, Michigan State University, East Lansing, Michigan, May, 1975.

SPT-2. DULPDX/DULPLX: Linear Programming Subroutines

Adapted from program developed by the University of Wisconsin Computing Center, Madison

DULPDX and DULPLX are subroutines used to solve linear programming problems. DULPDX is the DOUBLE PRECISION version of DULPLX. Both subroutines use the dual simplex algorithm with inverse explicit form; i.e., the inverse is stored explicitly in core. Since the inverse is stored explicitly, problems which may be solved by the subroutines are restricted by the core memory capacity and the length of the calling program. Calling sequences of DULPLX and DULPDX are identical but some DULPDX parameters must be declared DOUBLE PRECISION.

The main features of DULPDX and DULPLX are:

--A constraint or group of constraints can be easily added or dropped (when the problem id dual feasible).
--Special provision is available to handle the Gomory cutting planes for integer programming problems.
--Inequality and equality constraint types are easily specified via the type vector T.
--Restart without reinversion permits changing the cost and right hand sides without recalculating a new inverse.
--Nonzero costs may be assigned to slack variables.
--Full or partial basis may be specified by the user.
--The system automatically reinverts the basis when either an infinite or an infeasible solution is encountered.
--Reduced costs, inverse, and shadow prices are easily obtained.
--An intermediate output or a final output report option may be used.
--Many algorithm controls and tolerance settings are available.
--An option permits reinverting the basis whenever an optimum is reached.

However, provisions for cost ranging, right hand side ranging, etc. are not included.

Compilers: FORTRAN IV; CDC FORTRAN Extended, 1800 symbolic cards

Computer: Univac 1108; CDC 6000

Documentation: "DULPDX and DULPLX Dual Simplex Algorithm, Linear Programming Subroutines, Reference Manual for the 1108," The University of Wisconsin Computer Center, Madison, Wisconsin, May, 1971.

Comment: All inquiries regarding the acquisition of DULPDX/DULPLX should be sent directly to the Madison Academic Computer Center, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706, Phone: 608-262-2105, which maintains and distributes this program. The cost for documentation and source code on magnetic tape or cards is $50.

SPT-3. Interactive Optimization Component

Marcus Buchner, Michigan State University, East Lansing

SYSOPT utilizes numerical optimization techniques to find approximate solutions of parameter estimation or policy decision problems resulting from the use of complex simulation models. SYSOPT, an interactive computer program component, contains a "pattern recognition" subcomponent and an "optimization" subcomponent to assist in the efficient examination of either parameter or policy variable spaces.

In particular, the pattern recognition subcomponent allows an examination of model behavior in either a printed or graphical form. In addition, comparisons between model outputs, real world data, and smoothed real world data can be made with the component. Therefore this section of SYSOPT will assist in the solution of parameter estimation and policy decision problems in three principle ways: (1) through rapid isolation of regions in the input variable space representing invalid or insensitive behavior, (2) through rapid isolation of regions in the variable space needing detailed investigation, and (3) through assistance in the determination of the validity of the model.

The optimization subcomponent contains an efficient four-level interactive optimization algorithm. In the algorithm, the Complex method and Powell's method, both direct search numerical optimization methods, are integrated into a search technique that is effective both far from and in the vicinity of a local optimum. In addition, methods have been incorporated into the algorithm to perform global optimum searches and to insure compatibility between the Complex and Powell's method. Thus information gained from using one level of the algorithm is used to initialize another level; thus the convergence rate of the entire component is increased.

Particular emphasis in the development of SYSOPT was placed upon the problem solution accuracy, the number of simulation runs necessary to achieve convergence to the solutions, the interactive capability to change search structure and parameters during the component execution, and computer memory and central processor use.

In summary, SYSOPT can assist in the solution of problems often encountered in a systems approach such as model validation, model tuning, and model control. Although the straightforward application of the automatic optimization search techniques in the component can lead, to incorrect or illogical solutions to parameter estimation or policy decision problems, the use of the component together with proper human interactions can greatly assist in obtaining useful models and useful results from models.

Compiler: CDC FORTRAN Extended, CDC 6000 Version 3, 3000 symbolic cards

Computer: CDC 6000

Documentation: Marcus Buchner, "An Interactive Optimization Component for Solving Parameter Estimation and Policy Decision Problems in Large Dynamic Models," unpublished Ph.D. dissertation, Michigan State University, 1975.

Marcus Buchner, "A User's and Programmer's Guide to SYSOPT: An Interactive System Optimization Computer Program," CLASS Document No. 12, August, 1976.

SIMULATION LANGUAGES

SML-1. PAL: Policy Analysis Language

C. Winer and C. Wolf, Michigan State University, East Lansing

The Policy Analysis Language (PAL) was developed in order to make computer simulation models easily accessible to nontechnical users. PAL provides the analyst who designed the model with an easy-to-use "author" language for formating an interactive session with the user. During the session, the user may be presented with explanations of the model and its characteristics, given choices as to what areas he would like to explore, asked for input values, and shown the results of a model run. Although developed for use with simulation models, PAL is suitable for almost any FORTRAN program. An experienced FORTRAN programmer can learn PAL in an hour or two.

A PAL program may be constructed to offer the user two alternative sets of messages--typically a set of detailed explanations for the novice user and a set of brief prompts for the experienced user. In addition, whenever the user is expected to enter input, he may choose to enter commands for changing and printing data values, running the model, or performing various other actions. If desired, all functions may be performed by commands bypassing the conversation entirely. This capability is particularly useful to the programmer when checking out the model.

The PAL software consists of a translator and library routines written in FORTRAN. ANSI standards are followed whenever possible. The PAL translator translates a PAL program into FORTRAN to be compiled. The model and the PAL library routines are then loaded along with the resulting object code.

Compiler: CDC FORTRAN Extended, 6000 Version 4, 7200 symbolic cards

Computer: CDC 6000

Documentation: "Policy Analysis Language, Version 2.3, Programmer's Guide for CDC Cyber Computers," Claudia S. Winer and Chris Wolf, CLASS Document No. 3, July 1976.

"Policy Analysis Language, Version 2.3, Reference Manual," Claudia S. Winer and Chris Wolf, CLASS Document No. 4, July 1976.

"Policy Analysis Language, Version 2.3,Guide to Using a PAL Program," Claudia S. Winer and Chris Wolf, CLASS Document No. 5, July 1976.

UTILITY ROUTINES

UTR-1. Distributed Delay Routines: DEL, DELS, DELF, DELLF, DELVF, DELLVF

M. H. Abkin and C. Wolf, Michigan State University, East Lansing

All six routines in this set simulate distributed time delay processes which are useful for modeling information lags and flow delays inherent in such aggregated processes as investment, production, transportation, information dissemination, and demographic processes. DEL is the simplest form of the routines; the other routines have additional features. Five routines compute storage in the delay by integrating the net flow into the delay. Four routines insure simulation stability by automatically decreasing the integration step size when the time constant is small relative to the order of the delay. Finally, two routines allow the mean lag time of the delay to be varied over the simulation run.

Compiler: CDC FORTRAN Extended, 6000 Version 3, average of 18 source statements per routine (ANSI subset)

Computer: CDC 6000

Documentation: CLASS Document No. 8, July, 1976.

UTR-2. Table Functions: TABEL, TABEX, TABUL, TABUX

M. H. Abkin, C. Wolf, and T. W. Carroll, Michigan State University, East Lansing

All four functions in this set use straight line interpolation between known points of a dependent variable as a function of an independent variable. TABEL and TABEX handle cases in which there are equal intervals between the known values of the independent variables; TABUL and TABUX handle cases in which there are unequal intervals. Beyond the specified range of TABEX and TABUX, straight line extrapolation is used, where the slope of the line is the same as that of the last segment within the specified range; beyond the specified range of TABEL and TABUL, extrapolation is limited to end point values.

Compiler: CDC FORTRAN Extended, 6000 Version 3, 10 source statements per routine (ANSI subset)

Computer: CDC 6000

Documentation: CLASS Document No. 9, July, 1976.

CLASS Documents

Number	Title
1	"Computer Library for Agricultural Systems Simulation: A Progress Report," Michael H. Abkin and Tom W. Carroll. (July 1976)
2	"Software Standards Manual." (July 1976)
3	"Policy Analysis Language, Version 2.3, Programmer's Guide for CDC Cyber Computers," Claudia S. Winer and Chris Wolf. (July 1976)
4	"Policy Analysis Language, Version 2.3, Reference Manual," Claudia S. Winer anc Chris Wolf. (July 1976).
5	"Policy Analysis Language, Version 2.3, Guide to Using a PAL Program," Claudia S. Winer and Chris Wolf. (July 1976)
6	"DEMOGC: Demography with Distributed Age Cohorts," Michael H. Abkin and Chris Wolf. (July 1976)
7	"DEMOGD: Demography with Discrete Age Cohorts," Michael H. Abkin and Chris Wolf. (July 1976)
8	"Distributed Delay Routines: DEL, DELS, DELF, DELLF, DELVF, DELLVF," Michael H. Abkin and Chris Wolf. (July 1976)
9	"Table Functions: TABEL, TABEX, TABUL, TABUX," Michael H. Abkin, Chris Wolf, and Tom W. Carroll. (July 1976)
10	"AGACC: Accounting Routine for the Agricultural Sector," Dennis Pervis and Chris Wolf. (July 1976)
11	"User's Guide for the Beef Cattle Enterprise Simulation Model," Michael R. Jaske. (July 1976)
12	"User's Guide tc SYSOPT: An Interactive System Optimization Computer Program," Marcus Buchner. (August 1976)