December 14, 1982

Almond Board Project No. 82-H4: Modelling Population Dynamics
Project Leaders: J.K. Oddson and S. Aggarwal, Department of Mathematics, UCR

Model Status: Programming has now been completed and verified for all of the biological components, management options, and interactive features which were originally planned for the navel orangeworm computer simulation model. The program is currently available on the campus computers at UCR and on a research account on the statewide IPM computer network. A few revisions will be made before we release a finished version. Documentation and a user's manual are now in preparation. Implementation of the model on a microcomputer system such as the Apple III or the IBM PC will be attempted later.

Validation Studies: Recent simulation experiments, in which the model was first initialized with an orangeworm population taken from orchard mummy samples and then run forward through the year, have predicted spring oviposition patterns in excellent agreement with egg trapping data collected from the field. Additional trials must still be made but these indicate that we may have some confidence in the validity of the model. Other results shown by these simulations are:

(i) More information on navel orangeworm biology may be needed. Larval development parameters are somewhat uncertain; there is some indication, from both simulations and field observations, that an upper temperature threshold to development may be operating during the summer.

(ii) The period of spring oviposition seems to be relatively independent of the overwintering population profile; the pattern of spring temperatures is a more important factor.

(iii) Emergence of F_1 generation adults and subsequent oviposition are generally well-tuned to the period of hull-crack, but often a substantial portion of the eggs have been laid (back on mummies) before it begins.

(iv) Larval crowding in the mummies in the spring can result in significant mortality which may over-shadow or alter the effects of grower pest management actions. Further field work will be necessary to determine to what extent this factor is present.

(v) In the model an almond is judged "infested" if it shows any evidence of past or present larval activity in the nut or on its hull. There is a need to relate this measure of crop infestation to some more standard industry measure such as percentage of rejects at the huller.

Pest Management Simulations: A preliminary attempt has been made to use the model to evaluate management options of spraying and orchard sanitation. All simulations were run for the same typical almond orchard, subjected either to a single larvicidal spray at a selection of different dates or to various levels of clean-up (or dirtiness) on January 26. In each case the model was run forward into summer to determine the percent infestation in the Nonpareil crop at the time of the Nonpareil harvest knock (August 16), which was then compared with what the value would have been for the original, unsprayed field.

(a) Spraying results are shown on p. 3. The main point to note is that spraying at the "wrong" time may increase crop infestation. The explanation (in the model) for this phenomenon is that such sprays kill off eggs and larvae which would produce descendants which would appear too early to infest the new crop of nuts. Thus they would not contribute to the new crop infestation anyway. However, if they are killed off, the mummy nutritional resource that they would have eaten is still available for the later larvae which will now make it through the system in larger numbers to infest the new crop of nuts. i.e. we have here a strong interaction between the effects of the spray and a density-dependent larval mortality.

Best spraying times seem to be during the latter half of the spring oviposition period or during the early part of hullcrack. Even so, the best results seem to be unreasonably low (less than 15% improvement) which could indicate that the parameters of the spraying submodel are not correct.

We stress that these results are very preliminary and that no recommendation is being made.

(b) Sanitation results are shown on p. 4. The upper figure indicates how the relative crop infestation at Nonpareil harvest would vary with January mummy densities in the orchard. Results are shown relative to the infestation in our typical orchard, which had slightly more than 10 mummies per tree on January 26. Note that the graph differs significantly from a straight line only for winter mummy densities which would not be found in commercial groves. The assumption in these simulations was that all mummies knocked to the ground were then destroyed. The lower figure on p. 4 shows a blow-up of this graph for mummy densities less than 10. It also includes, for comparison, the simulated results of what would happen if the orchard were cleaned to various levels (from slightly more than 10 nuts/tree) but if the mummies that were knocked off were then left undestroyed on the ground. The model indicates that slightly better infestation control is achieved in this case. Again the explanation (in the model) involves the density-dependent larval mortality. Larvae and pupae continue to develop in the ground nuts (although with increased mortality) and emerge in the spring to join the other moths in laying eggs on the remaining nuts in the trees. (No oviposition occurs on ground nuts.) This creates a higher larval density per nut and thus a higher larval mortality, especially for the later arrivals, due to a reduced nutritional base upon which they can feed.

Whether this explanation is valid or not, it does seem clear that winter clean-up can have a dramatic effect in reducing crop infestation while the case for spraying is still unclear. Note also that no economic costing has been included yet in our simulations.

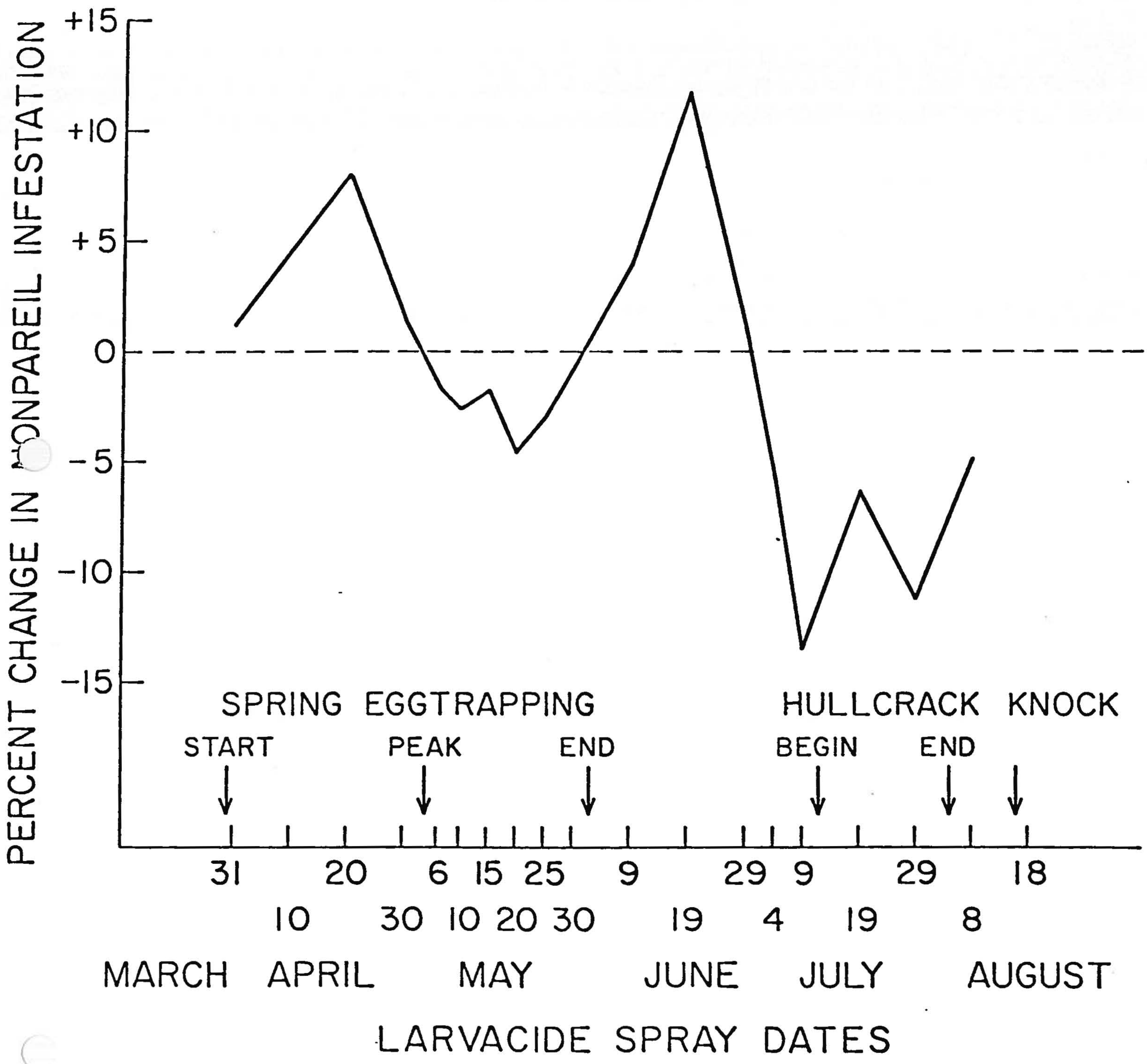
General Remarks on Validation Trials

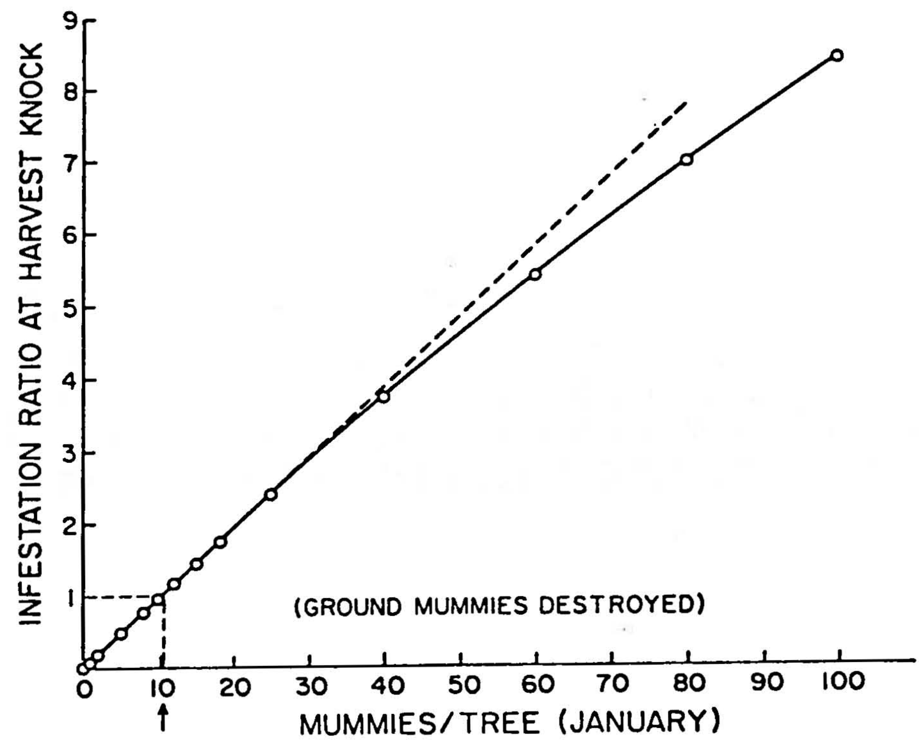
1. The period of spring oviposition seems to be relatively independent of the over-wintering population; the spring temperature pattern is a more important factor.

2. Emergence of F_1 generation adults and subsequent oviposition are generally well-tuned to hull-crack, but often a substantial portion of the eggs have been laid (back on mummies) before it begins.

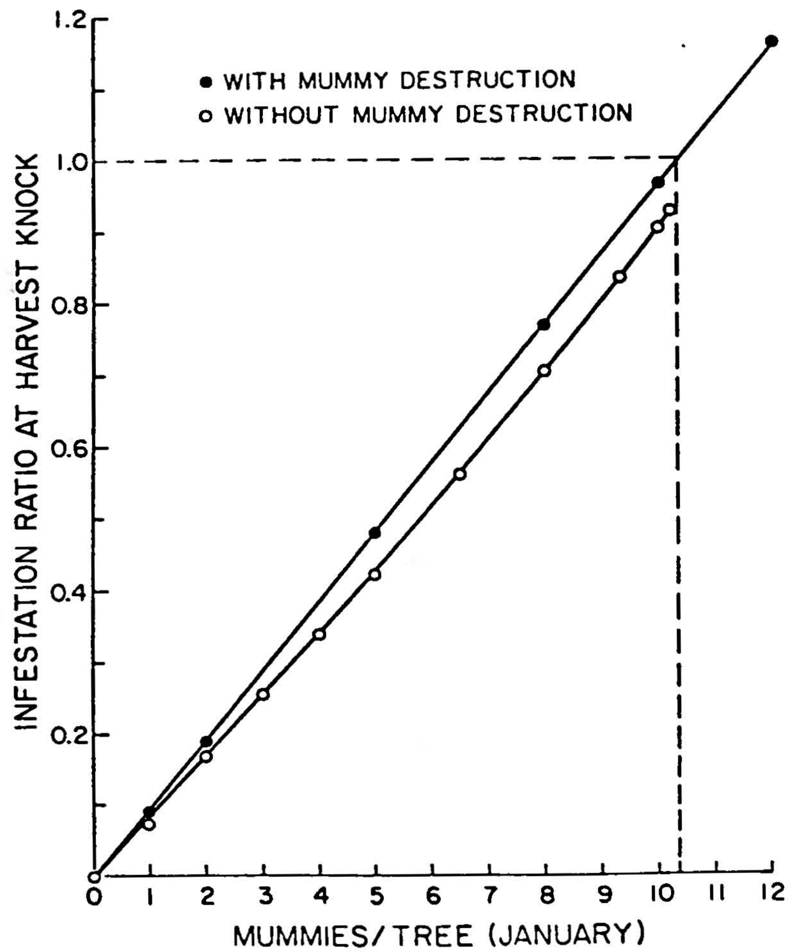
3. Larval crowding in the mummies in spring can result in significant mortality which may over-shadow or alter the effects of grower pest management actions.

4. Crop infestation values must be interpreted in a relative, not absolute, sense until further comparative field work can be done.


PERCENT CHANGE IN NONPAREIL INFESTATION
+15
+10
+5
0
-5
-10
-15
SPRING EGGTRAPPING
START
PEAK
END
HULLCRACK
BEGIN
END
KNOCK
31
10
20
30
6
10
15
20
25
30
9
19
29
4
9
19
29
8
18
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
LARVACIDE SPRAY DATES



INFESTATION RATIO AT HARVEST KNOCK
(GROUND MUMMIES DESTROYED)
MUMMIES/TREE (JANUARY)



WITH MUMMY DESTRUCTION
WITHOUT MUMMY DESTRUCTION
INFESTATION RATIO AT HARVEST KNOCK
MUMMIES/TREE (JANUARY)

Almond Board of California

Project No. 83-H5
(Continuation of Project No. 82-H4)

Cooperator:
University of California
Department of Mathematics
Riverside CA 92521

Project Leaders: Dr. J. K. Oddson (714) 787-5001 or 787-3113
Dr. Sudhir Aggarwal (714) 787-3627 or 787-3113

Personnel: John Sanderson collaborating with Dr. M. M. Barnes

Project: Navel Orangeworm, Mite and Insect Research
Modelling Population Dynamics of the Navel Orangeworm in Almonds

Objectives: (1) To field test and refine the completed navel orangeworm computer simulation model. (2) To evaluate management options of orchard sanitation, spring and hullcrack spraying, and timing of the harvest by means of simulation experiments with the model.

Progress: The NOW model has been completed and is presently available on the campus computers at UCR and on a research account on the statewide IPM computer network, although some revisions will be made before a final version is released. Validation experiments to date agree well with field observations. Initial pest management simulations indicate that a density-dependent larval mortality in mummies may play a significant role in the selection of effective control strategies.

Plans: In close collaboration with John Sanderson and Dr. M. M. Barnes, (1) To test and refine the current model by following navel orangeworm development during the year in one or two selected orchards by means of model simulations initialized and corrected with successive field observations; (2) To run further laboratory, field and simulation studies of some aspects of navel orangeworm biology such as an upper temperature development threshold, larval developmental parameters, and the density-dependent larval mortality on mummies; (3) To use simulations to evaluate the relative effectiveness of different pest management control strategies in reducing NOW infestation at harvest.

PROGRESS REPORT: UC/IPM PROJECT

December 15, 1982

Title:

NAVEL ORANGEWORM COMPUTER SIMULATION MODEL

Principal Investigators:

J. K. Oddson and S. Aggarwal (787-3113)
Associate Professors, Department of Mathematics
University of California, Riverside

Collaborating with M. M. Barnes (Professor, Dept. of Entomology, UCR);
P. Wilkinson (Graduate Student in Computer Science, Dept. of Mathematics, UCR);
J. Sanderson (Graduate Student, Dept. of Entomology, UCR).

Objectives:

(i) To make the programming modifications to our n.o.w. microcomputer simulation model that will convert it to a form which can run on the Prime computers of the IPM computer network;

(ii) To transfer the converted program to a Research account on the IPM network and to create the files of sample data which will be necessary for its operation;

(iii) To conduct trial simulation runs on the IPM network in order to check the operation of the program and its interactive I/O features.

Timetable:

(1) June 15 - July 31: Program modification; transfer to IPM network.

(2) Aug. 1 - Aug. 15: Creation and/or transfer of input data files; explore linkages with other weather data bases.

(3) Aug. 16 - Sept. 14: Testing of program on the IPM network; consultation on interactive I/O features.

Summary:

The navel orangeworm computer simulation program, developed on UCR campus TERAK microcomputers and a DEC VAX 11/750 TERAK emulator under grants from the Almond Board of California, has been successfully adapted to the Prime computers of the statewide IPM computer network and is now running on a Research account on the

Riverside IPM facility. A demonstration of its operation was presented at the 1982 Annual Research Conference of the Almond Board, in Sacramento, using a telephone hook-up to the IPM network through the UC Davis IPM facility.

The current version of the program includes verified code for all of the biological components, management options, and interactive features which were originally planned for the navel orangeworm model. In this sense it is now complete, although a few revisions will be made before it is considered ready for release. Such changes will be made simultaneously in both the campus and the IPM network versions, in order to maintain these two implementations of the model at the same level of development. Technical documentation of the program and a users' manual for its operation are to be prepared before it is presented to the Almond Commodity Group for IPM Project evaluation.

Linkage of the IPM network version to the statewide weather data base that is now available on that network has been determined to be both feasible and desirable, and will be attempted in the coming months.

Budget:	Requested	Awarded	Expended
Programmer, Setp I, 3 months, full time	$6,420	$5,131	$5,131
Benefits	124		
Total	$6,544		

Accomplishments for 1982-83:

The three objectives for this project will be discussed here in one section, since they are closely related and combine to form the single overall objective of preparing the n.o.w. computer simulation program for eventual operation on the IPM network.

(a) Justification for the project

The n.o.w. model, with its potential for simulating the population dynamics of a navel orangeworm infestation in an almond orchard under the influence of weather and grower control practices, should provide a better quantitative understanding of the life cycle and crop interactions of this pest and lead to improved measures for its control. Installation on the statewide IPM network would make it more readily available to researchers and pest management advisors, first for testing and validation and later for field implementation.

(b) Methods and Results

Computer programs seldom are directly transportable from one computer system to another. Changes must be made to take into account differences in the operating system environments of the machines and in the programming languages that they support.

In our transfer of the n.o.w. simulation program to the IPM network these changes included the creation of additional language constructs, revision of some language syntax, creation of new I/O intrinsic functions, and a restructuring of the program for independent module compilation. Although extensive re-coding of the program was required, comparison testing of the completed IPM version indicates that a faithful conversion has been achieved. A listing† of the program and sample output from a typical simulation run are appended to this report.

Details of the work on this project can best be discussed in terms of the stated project objectives:

Objective (1) - "To make the programming modifications to our n.o.w. microcomputer simulation model that will convert it to a form which can run on the Prime computers of the IPM computer network."

The n.o.w. model was designed to use a processing algorithm based on "discrete event simulation", which is best implemented in PASCAL to take advantage of the advanced record, pointer-variable, and linked-list data structures of that programming language. The n.o.w. program was written in an enhanced version of "standard" PASCAL, known as UCSD PASCAL, which is available on many mini and microcomputer systems. The Prime 250 at the Riverside IPM facility supports PASCAL, but in a "standard" version that lacks the language enhancements and system intrinsics of UCSD PASCAL. To use Prime PASCAL required a number of coding additions and modifications:

---

† This is included only as formal evidence of the conversion work and is not, nor was it intended to be, a very enlightening document. The structure of the navel orangeworm simulation model can be understood much more easily from a listing of the original form of the program, which is not encumbered by the additional (and extraneous, from the point of view of the model) language and system support coding that had to be included in the IPM version.

(i) The n.o.w. program makes extensive use of a variable type called 'string' - an array of characters which may be manipulated as a block - particularly in the interactive I/O components, which provide the main user-convenience features of the program. UCSD PASCAL has 'string' variable and 'string-handling' intrinsics built into the language; Prime PASCAL does not.

A variable type STRING had to be built, as an 80-position array of characters, and then filled with the desired elements of the 'string', character by character. Constant 'strings' had to be filled and declared as global variables. This added an extra 60 items to the variable list for the program. A number of 'string-handling' routines also had to be created. These will be discussed below.

Much of the additional code that had to be inserted into the IPM version of the n.o.w. program involved the creation and manipulation of such 'string' variables, which in a less primitive implementation of the PASCAL language would have been handled at the system level.

(ii) UCSD PASCAL supplies an intrinsic routine EXIT, which provides for a premature exit from a procedure when a specified condition has been met. It is not available in Prime PASCAL. Most uses of EXIT in the n.o.w. program could be handled by rewriting the code with negative logic, i.e., the remaining statements in the procedure would be executed only if the specified condition was <u>not</u> met. In a few instances a clumsier sequence of GOTO statements and LABELS had to be used.

(iii) The present version of Prime PASCAL does not include an OTHERWISE option in its CASE statements, and yet does not permit unlisted case values to merely "fall through". Explicit instructions must be provided for every case value which may occur, or a STACK OVERFLOW error may be generated at execution time. All CASE statements in the n.o.w. program had to be enclosed in protective coding to ensure that such an error would not occur.

Objective (2) - "To transfer the converted program to a Research account on the IPM network and to create the files of sample data which will be necessary for its operation."

Transfer of the n.o.w. program to run successfully on the IPM network also entailed changes to make it compatible with the hardware/software environment of the prime 250. These changes were of two types:

(i) Adaption of the interactive I/O and file handling facilities of the n.o.w. program.

Some of the interactive features of the n.o.w. program depend upon a READ command to retrieve single-character user-responses from the keyboard. In order that they function properly on the Prime 250, extra care had to be taken to ensure that the input buffer was initially empty. Often a dummy READ statement had to be inserted to first clear the buffer (after a READLN command, for example, which does not pick up its own "carriage return"). Interactive screen-output routines in the original n.o.w. program took advantage of a UCSD PASCAL intrinsic function, GOTOXY, to control the position of the cursor. Prime PASCAL does not have this feature, so a substitute had to be created with the use of an "escape sequence". The effect produced by the control characters in the sequence varies from one terminal type to another, however since all terminals on the IPM network are of the same type (ADDS REGENT 60) there should be no problem.

String-handling routines, required for the manipulation of 'string' variables in the interactive I/O features of the n.o.w. program, and intrinsic to UCSD PASCAL,had to be created for the IPM version of the program. They include the

Procedures:

READSTR - reads any 'string' (up to 80 characters) from a text file;

WRITESTR - writes any 'string' (up to 80 characters) to a text file and leaves the file pointer at the end;

WRTSTRLN - same as WRITESTR, but moves the file pointer to the beginning of the next line;

WRITEFORM - right justifies a 'string' on an output line, leaving the file pointer at the end of the line;

CONCAT - joins two 'strings' to produce a new combined 'string';

PRINTEVENT - converts an enumerated data type, EVENTKIND, to a printable, 'string' form,

and the

Functions:

LEN - return the length of a 'string' variable;

POS - returns the position of the first occurence of a 'string' within another 'string';

EQUAL - tests whether two 'strings' are identical, in all of their characters;

LESSTHAN, GREATERTHAN - tests the ASCII character ordering of any two 'strings'.

In the interactive n.o.w. program various text and data files--assistance, parameters, temperature values, population profiles--are meant to be accessed by the user for display, modification, or loading into the program. If the specified file does not exist or otherwise cannot be opened through user oversight or an input error on its name, the command cannot be executed and an error flag would be passed to the operating system, which would then abort the run. To prevent this from happening, the original n.o.w. program took advantage of another UCSD PASCAL intrinsic, IORESULT, to test the status of a file before access was attempted. Prime PASCAL has no such facility so one had to be created. A FORTRAN 77 function, EXIST, was written to test the status of a file, using a basic FORTRAN intrinsic called INQUIRE. This routine was linked into the n.o.w. PASCAL program as an external function and invoked whenever needed by another PASCAL procedure, IORSLT, created for this purpose.

(ii) Restructuring of the program for independent module compilation. It became clear in the early stages of this project that compilation of the n.o.w. PASCAL program on the Prime 250 would have its problems. Full program compilation took upwards of an hour! Complaints were received concerning system access for other users of the Prime 250--complaints related to the enormous amount of CPU time that was being used for PASCAL compilations. This is clearly a Prime compiler/operating system problem; apparently the current version must use repeated memory swapping. Comparison should be made with the TERAK emulator on the DEC VAX 11/750 where full compilation takes from 2 to 5 minutes, depending upon system load, and with the TERAKs themselves, where full compilation (of an earlier version) was completed in less than 8 minutes.

A compromise solution was found, by taking advantage of the "separate module compilation" feature of Prime PASCAL. Subsections of the program could be modified and compiled on their own, provided appropriate global variable definitions were available. However, implementation of this solution required considerable restructuring of the n.o.w. program. All previous "include files" and "forward referenced procedures/functions" had to be discarded and a new structure of "definition" and "include" files created. The end result undoubtedly justified the effort, even apart from resolving the compilation-time problem, since the new modular form of the program will be more convenient for making future changes. Nonetheless it was a time-consuming task, and it did create an unexpected delay in the project, of 2-3 weeks.

The program must also be supplied with sets of data on orangeworm and orchard parameters, an initial population profile, and daily max-min orchard temperatures, before a simulation can be run. The parameter and population files generally are not a problem since they are designed to be created by the user, interactively from within the program. Several were generated to test the operation of the system. (See the appended sample output for parameter and population file examples.) Temperature files are more extensive and are handled in a different manner. There is a separate "weather utility program", written in UCSD PASCAL, which permits a user to create, modify, display (to the screen), or print (hardcopy) temperature data files which would be accessed by the n.o.w. program during a simulation run. Some existing temperature files which had been created earlier on the TERAKs and the VAX 11/750 with this utility routine, using data from thermograph records from specific orchards for 1979-82, have been transferred to the IPM account by simply taping copies. Of course it would be more convenient to have the "weather utility" routine itself on the IPM account. A Prime PASCAL conversion/revision of this program is in progress. Hopefully it can be expanded to include an additional user option - available only on the IPM network - of interfacing with the on-line statewide weather data base in order to retrieve existing temperature files from the data base, and to prepare them for use by the navel orangeworm program.

Objective (3) - "To conduct trial simulation runs on the IPM network in order to check the operation of the program and its interactive I/O features."

Although extensive modifications have been made to the original n.o.w. simulation program in order to convert it to a form which will run on the IPM network, both programs are based upon the same underlying model and use the same processing algorithms. They should produce the same results. Differences can only occur through incorrect adaption of the I/O commands or programming errors which may have been made during the conversion.

Considerable care was taken to ensure that the interactive I/O features were identical for both versions, so that the program would appear to be the same to a user on either the VAX 11/750 or the IPM network. In fact suggestions made for changes in some of the screen formatting on the IPM terminals were also implemented later in the original n.o.w. program.

Coding errors which may have crept in during the conversion process and which have not been caught by a careful proofreading of the program can often be exposed by comparing "benchmark" simulation runs made on both systems. Project delays caused by the need to restructure the program for separate module compilation did not leave much time for this testing, however a number of trials were conducted. Numerical results from the two versions of the program were virtually identical (see Appendices B and C), with discrepancies apparently caused by the manner in which Prime represents and displays decimal real numbers. (Prime truncates them on output, instead of rounding, to the specified number of decimal places.) Additional comparison trials should be run to pinpoint exactly the source of these differences, but for all practical purposes it appears that an accurate program conversion has been achieved.

Linkage With Other Projects: Implementation

As mentioned above, the possibility of linking the n.o.w. program through a "weather utility" routine to the statewide weather data base is being explored.

Work is continuing on the original n.o.w. simulation project, under Almond Board grants, to test and refine the model and to evaluate its use as a tool for management decisions on sanitation and the timing of spraying and harvest. Any changes that may be made will be incorporated into the IPM version of the program at the same time. Detailed documentation of the coding is nearing completion. A description of the underlying conceptual model and a users' manual for operation of the program on the VAX 11/750 are to be prepared. Only minor modifications will be required to explain operation of the program on the IPM network. Hopefully, preliminary versions of these manuals can be ready during summer, 1983. At that point the program will be presented to the IPM Almond Commodity Group for evaluation and pilot testing by Area IPM Specialists and other qualified individuals.