Behavior Research Methods, Instruments, & Computers
1991, 23 (3), 415-429

The Observer: A software system for collection and analysis of observational data

L. P. J. J. NOLDUS
Noldus Information Technology b.v., Wageningen, The Netherlands

The Observer is a general-purpose software package for event recording and data analysis in behavioral research. It allows any IBM-type personal computer to serve as an event recorder. In addition, The Observer can generate dedicated event-recording programs for several types of non-IBM-compatible portable and hand-held computers and transfer files between the PC and such computers. The user specifies options through menus. The configuration can be either used directly for event recording on the PC or passed on to a program generator that creates a program to collect data on a hand-held computer. Observational data from either type of computer can be analyzed by the program. Event-recording configurations can be tailored to many different experimental designs. Keys can be designated as events, and modifiers can be used to indicate the limits of an event. The program allows grouping of events in classes and distinction between mutually exclusive versus nonexclusive events and duration events versus frequency events. Timing of events is accurate to 0.1 sec. An on-line electronic notepad permits notes to be made during an observation session. The program also includes on-line error correction. User comments as well as independent variables can be stored together with the observational data. During data analysis, the user can select the level of analysis and the type of output file. The Observer calculates frequency of occurrence and duration for classes of events, individual events, or combinations of events. For analysis of concurrence, one can select the number of nesting levels and the order of nesting. Output can be generated in the form of sorted event sequence files, text report files, and tabular ASCII files. The results can be exported to spreadsheet and statistical programs.

Hardware for Observational Studies

Many types of behavioral experiments involving direct observation of animals or humans require the recording and timing of sequences of events. While certain circumstances still require paper-and-pencil methods, computer event recorders are increasingly used (Horner & Storey, 1989; Lehner, 1979; Martin & Bateson, 1986). Most first-generation event recorders were dedicated devices with specialized keyboards and inaccessible firmware (Noldus, 1989; Noldus, van de Loo, & Timmers, 1989). This type of equipment is now rendered obsolete by the widespread availability of microcomputers, which are programmable and do not need custom-designed keyboards. The type of hardware selected depends on various aspects of the observational study. With regard to data collection, decisive factors are experimental design (sampling method, number of event categories, and number of events to be recorded per unit of time) and location of the observations. For example, complex continuous observations, with many different events occurring in rapid succession, require a fast computer, whereas a sampling method including irregular, unformatted notes requires the flexibility of a checksheet. When space is not a constraint, a desktop personal computer (PC) can be used; however, in many situations, one will need the compactness of a portable computer. Observations outdoors always require a portable device, such as a laptop computer for sedentary observations (Unwin & Martin, 1987), or a hand-held pocket computer (Whiten & Barton, 1988). The environmental settings in which the equipment is to be used limit the choice of hardware, with respect to robustness and resistance to temperature extremes, humidity, and dust. Experiments under low-light conditions require a computer with either a backlit or an externally illuminated display.

The development of The Observer was initiated at the Department of Entomology, Agricultural University, Wageningen, The Netherlands. Many people contributed to the development of the system. The author is indebted to colleagues and students at the department and to M. F. Wallis de Vries (Department of Nature Conservation, Agricultural University, Wageningen), E. L. H. M. van de Loo (Department of Clinical and Health Psychology, University of Leiden), R. M. Michiels van Kessenich (PC-Slide, Amsterdam), O. P. J. M. Minkenberg (Department of Entomology, University of California, Davis), and P. H. A. Timmers (Philips Data Systems, Apeldoorn). Special thanks to the critical beta testers M. L. Boccia (Department of Psychiatry, University of Colorado Health Sciences Center, Denver), W. A. van Giessen (Institute for Horticultural Plant Breeding, Wageningen), D. R. Papaj (Department of Ecology and Evolutionary Biology, University of Arizona, Tucson), and R. P. J. Potting (Noldus Information Technology). K. Storey (Allegheny-Singer Research Institute, Pittsburgh) kindly provided unpublished materials. The manuscript has further benefited from the comments of R. T. Cardé, M. Dicke, M. van Helden, J. C. van Lenteren, P. Roessingh, J. M. Schmidt, J. H. Visser, and three anonymous reviewers. Correspondence should be addressed to L. P. J. J. Noldus, Noldus Information Technology b.v., University Business & Technology Center, Vadaring 51, NL-6702 EA Wageningen, The Netherlands (E-mail: noldus@rcl.wau.nl).

Note: The author has a direct financial interest in some of the software or hardware described in this paper.—*Editor*

As far as permanent storage and analysis of data are concerned, currently available PCs are powerful enough (with respect to memory, disk capacity, and processing speed) to perform most of the required tasks. That means that the same type of device can be used for event recording and data analysis. However, in many circumstances, there will be no direct hardware compatibility between the apparatus on which data are collected and the one on which they are to be analyzed, and some kind of data transfer medium will have to be implemented. For example, many currently available IBM-compatible laptop computers are still rather expensive and unable to operate on batteries for more than a few hours, which makes them unsuitable for field work. Many are too heavy to be practical in situations where the observer has to be mobile.

Event-Recording Software

An event-recording program must be adapted to the machine on which it has to run, as well as to the experimental design for which it will be used. It should preferably do more than simply respond to a keypress with storing a code representing the key and the time at which it was pressed. The flexibility of microcomputers and high-level programming languages facilitates programs with error trapping and correction, distinction between mutually exclusive and nonexclusive events, storage of independent variables, and so on. However, as soon as such features are built into a program, one has to choose between two options: one can make them a fixed part of the program, making the program dedicated for a particular research application, or one can include a facility that allows the user to configure the program according to his/her personal desires. The former approach implies that any change in experimental design will require a change in the software, which means that the tedious and error-prone process of rewriting and debugging the program must be repeated. On the other hand, programs of more general nature may easily become unsuitable to be run on compact hand-held computers with limited memory and processing speed. The larger the amount of program code, inherent to the program's flexibility, the less memory space is left for data storage during operation. Also, the more evaluation (by the program) needed to process each individual keypress, the slower the program will become. Finally, a change from one type of computer to another (e.g., if one wants to change from a lab to a field setting, or if the model one has chosen ceases to be produced or supported) demands a new event-recording program. This means either writing a new program or adapting a program written by someone else. In the latter case, there is a fair chance that data files will be in a different format, which makes them incompatible with data files produced so far.

The problems outlined above were the impetus for the development of a general-purpose, integrated software package for event recording and data analysis. The development of this package, named *The Observer*, started at Wageningen Agricultural University in 1986. Since November 1989, development has been continued by Noldus Information Technology b.v. in Wageningen. This paper describes version 2.0 of The Observer and represents the first comprehensive publication of the system. A schematic representation of the system is given in Figure 1. The components of the system are described in more detail below. The use of the various features is illustrated with a hypothetical case study of the social behavior of monkeys (Appendix A).

DESCRIPTION OF THE SYSTEM

Hardware Requirements

The Observer uses any IBM-type PC and compatible clones (hereafter collectively referred to as IBM PC) as host computer and as event recorder. The program runs under DOS version 2.0 or higher and requires a minimum of 384K free RAM. Besides the IBM PC, various small portable computers are supported as event recorders. The list currently includes the TRS-80 Model 100, Tandy 102, Olivetti M10, Epson PX-4, Epson PX-8, Psion Organiser, and Husky Hunter. The first five models are typi-

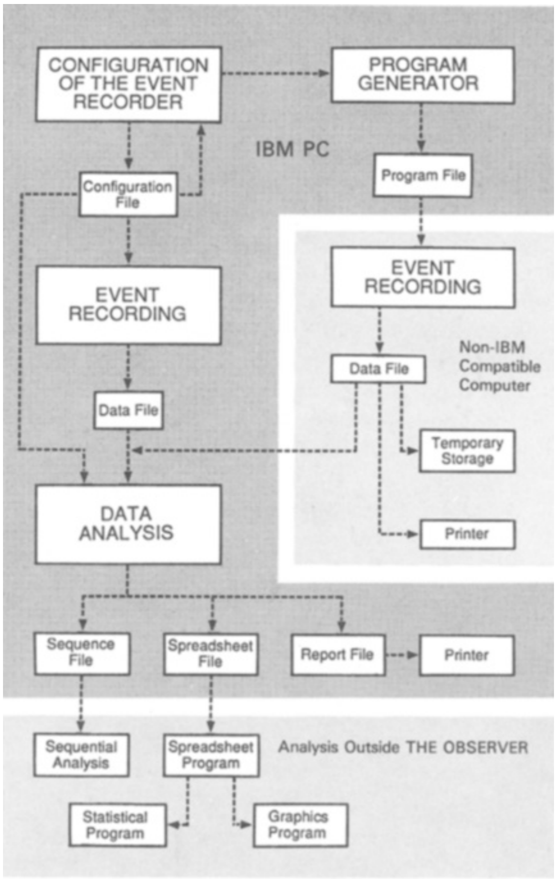


Figure 1. Schematic representation of the main components of The Observer. The user specifies options in settings screens in the configuration module. The configuration can be either used directly for event recording on the PC or passed on to a program generator that creates a program to collect data on a non-IBM-compatible portable computer. Observational data files from either type of computer are processed in the analysis module. The results can be exported to spreadsheet and statistical programs.

cal "notebook" computers with either an Intel 80C85 or a Zilog Z80 CPU. For details about the features of the TRS-80 Model 100, see Berman (1983), Deni, Szijarto, Eisler, and Fantauzzo (1983), and Gilreath (1985). For the merits of the Epson PX-8 as an event-recording device, the reader is referred to Martin and Bateson (1986) and Unwin and Martin (1987). Each of these computers is inexpensive and offers a compact combination of a programmable computer, a solid-state RAM that keeps data stored even when the machine is turned off, an RS-232 port for data transfer, and an eight-line display that allows menu-driven operations. The Psion Organiser is truly a handheld computer, with dimensions of only 8×14×3 cm and a weight of only 250 g (CPU: Hitachi 6303XF). The Husky Hunter is a very robust waterproof, dustproof, and shockproof A5-size computer in a metal housing (CPU: NSC 800). All models use batteries as well as AC voltage as power sources. Because of these features, they can be used in settings where most PCs cannot be used (e.g., in the field; see Berry, Holtzer, & Norman, 1987). The computers come with a BASIC interpreter (except the Psion Organiser, which uses OPL, a Pascal-like compiler), as well as other software in ROM.

Language

The Observer 2.0 has been developed with the Microsoft QuickBASIC Compiler, version 4.5, and the Microsoft Overlay Linker, version 3.69. It includes many assembly-language routines from the QuickPak Professional library, version 1.25 (Crescent Software, Stamford, CT). The system is currently being ported to MS-BASIC PDS 7.0.

User Interface

The Observer is fully menu-driven and does not require any programming by the user. It is designed around a main menu, from which the various procedures can be accessed. The program has been designed to be easy to use and intuitively understandable, even for people with little experience with computers or observational research. In menus, scrolling bars with explanatory text in a "quick reference" window help the novice user, whereas shortcut keys (Figure 2A) can be used by advanced users to select menu entries. In input forms, one can select options either from a pop-up window or by pressing the first letter of the desired option (Figures 2C and 2D). Screen colors and auditory signals are used in a standardized manner.

The user can attach descriptive comments to any file that is created with the program. Since DOS limits file names to eight characters, these comments facilitate the identification of files (Figure 2B).

The program contains an on-line manual through which the user can browse, as well as context-sensitive pop-up help screens. Furthermore, the program is resistant to most user-caused errors, which are trapped and return a relevant error message. Any entry or decision made by the user can be canceled by pressing the ESC key.

Configuration of the Event Recorder

In this part of the program, the user enters information to configure the event recorder. This allows him/her to tailor the event recorder to the particular experimental design. All information is saved in a *configuration file*, which is used at the start of a data-collection session on the PC and during data analysis (Appendix A). The configuration file can also be retrieved for further editing. A complete event-recording configuration has the following elements:

Computer used as event recorder. The user can select between IBM PC or compatible and one of the supported non-IBM-compatible portables (Figure 2C).

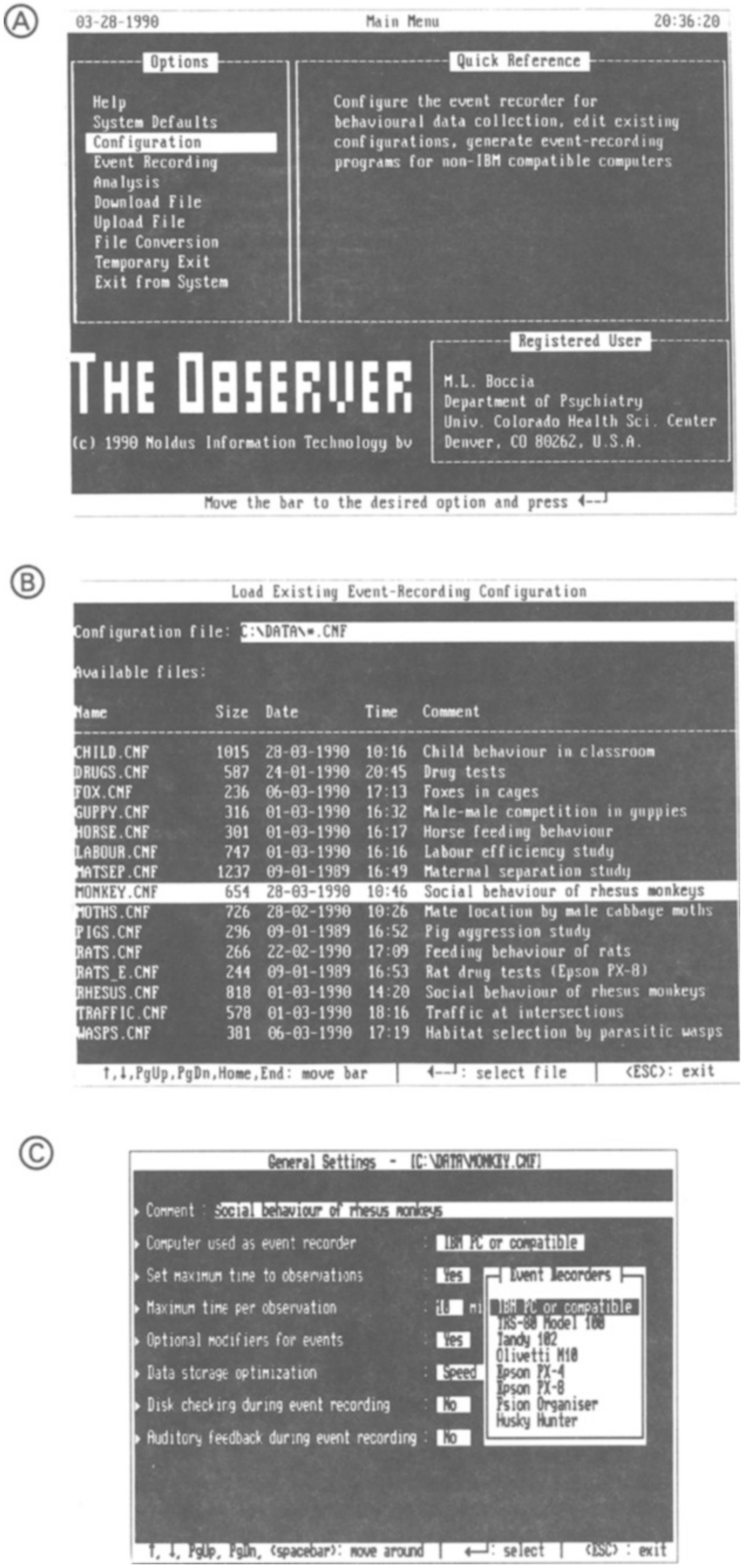
Maximum time for observations. One can set a maximum time (in minutes) after which an observation will be terminated automatically. Apart from this, observations can always be terminated by the user.

Data-storage method. One can let The Observer store data by optimizing speed (i.e., as fast as possible) or by optimizing space (i.e., as compact as possible). The former implies a little less efficient memory or disk usage, whereas the latter takes a little more time. This only affects performance when one of the non-IBM-compatible computers is used as an event recorder.

Disk checking during event recording. During event recording, data are continuously written to disk, which minimizes the loss of data in case of a power failure. However, data can also be lost if a write attempt fails due to a disk error. To prevent this, one can let The Observer check the status of the data disk prior to each write attempt. Since it may slow down the program on some computers, the use of this feature is optional.

Auditory feedback during event recording. One can let The Observer produce a short tone with a unique pitch at the depression of each event key during observations. This option can be useful if one has to operate the event recorder "blindly."

Classes of events. Events can be grouped in up to eight *classes*. This has two possible applications. (1) During data analysis, statistics can be calculated per class of events. In the present example, the program could calculate the time spent on social behavior (class with four behaviors) versus other behaviors (two behaviors). (2) Events within a class can be made *mutually exclusive*. This is discussed below. For each class (Figure 2D), one has to specify: (1) The *name* of the class. Examples: location, locomotory behavior, and social behavior. (2) The *number of events* in the class. This can be any number, as long as the total number of events does not exceed 92. In the example, the class *location* includes three events. (3) Whether or not the events in the class are *mutually exclusive*. In a class with mutually exclusive events, only one event can be active at a time. Consequently, an event in such a class, when turned on, automatically turns off the previous event in the same class. This feature can be used to create simple on/off toggles (as present on most dedicated event recorders), as well as



A

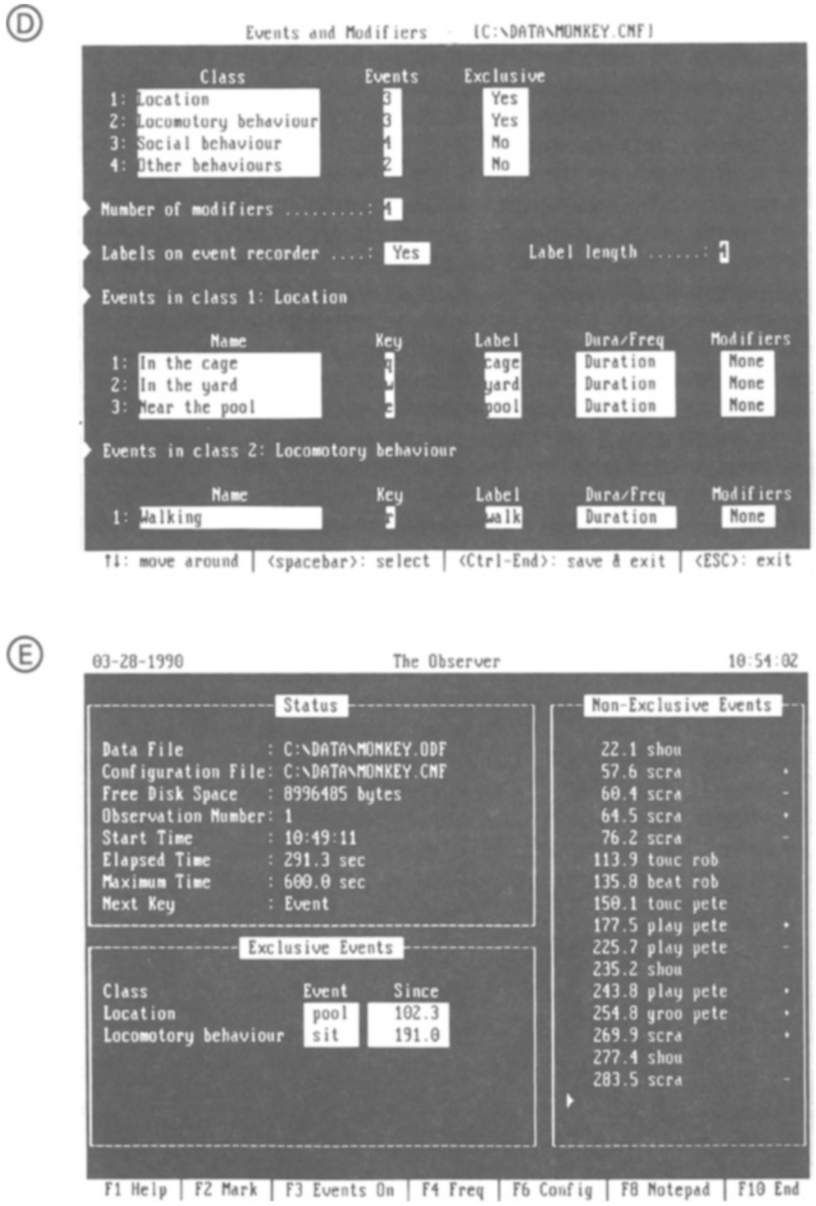
```
03-28-1990                     Main Menu                          20:36:20

  Options                        Quick Reference

   Help                           Configure the event recorder for
   System Defaults                behavioural data collection, edit existing
   Configuration                  configurations, generate event-recording
   Event Recording                programs for non-IBM compatible computers
   Analysis
   Download File
   Upload File
   File Conversion
   Temporary Exit
   Exit from System

                                        Registered User
THE OBSERVER
                                       M.L. Boccia
                                       Department of Psychiatry
                                       Univ. Colorado Health Sci. Center
(c) 1990 Noldus Information Technology bv   Denver, CO 80262, U.S.A.

           Move the bar to the desired option and press ←┘
```

B

```
              Load Existing Event-Recording Configuration

Configuration file: C:\DATA\*.CNF

Available files:

Name               Size  Date        Time   Comment
-----------------------------------------------------------------------------
CHILD.CNF          1015  28-03-1990  10:16  Child behaviour in classroom
DRUGS.CNF           587  24-01-1990  20:45  Drug tests
FOX.CNF             236  06-03-1990  17:13  Foxes in cages
GUPPY.CNF           316  01-03-1990  16:32  Male-male competition in guppies
HORSE.CNF           301  01-03-1990  16:17  Horse feeding behaviour
LABOUR.CNF          747  01-03-1990  16:16  Labour efficiency study
MATSEP.CNF         1237  09-01-1989  16:49  Maternal separation study
MONKEY.CNF          654  28-03-1990  10:46  Social behaviour of rhesus monkeys
MOTHS.CNF           726  28-02-1990  10:26  Mate location by male cabbage moths
PIGS.CNF            296  09-01-1989  16:52  Pig aggression study
RATS.CNF            266  22-02-1990  17:09  Feeding behaviour of rats
RATS_E.CNF          244  09-01-1989  16:53  Rat drug tests (Epson PX-8)
RHESUS.CNF          818  01-03-1990  14:20  Social behaviour of rhesus monkeys
TRAFFIC.CNF         578  01-03-1990  18:16  Traffic at intersections
WASPS.CNF           381  06-03-1990  17:19  Habitat selection by parasitic wasps

  ↑,↓,PgUp,PgDn,Home,End: move bar  |  ←┘: select file  |  <ESC>: exit
```

C

```
              General Settings  -  [C:\DATA\MONKEY.CNF]

▸ Comment : Social behaviour of rhesus monkeys

▸ Computer used as event recorder          : IBM PC or compatible

▸ Set maximum time to observations         : Yes   ┤ Event Recorders ├

▸ Maximum time per observation             : 10  mi IBM PC or compatible
                                                    TRS-80 Model 100
▸ Optional modifiers for events            : Yes    Tandy 102
                                                    Olivetti M10
▸ Data storage optimization                : Speed  Epson PX-4
                                                    Epson PX-8
▸ Disk checking during event recording     : No     Psion Organiser
                                                    Husky Hunter
▸ Auditory feedback during event recording : No

  ↑, ↓, PgUp, PgDn, <spacebar>: move around  |  ←┘: select  |  <ESC> : exit
```



```
Events and Modifiers   [C:\DATA\MONKEY.CNF]

        Class                Events    Exclusive
  1: Location                  3         Yes
  2: Locomotory behaviour      3         Yes
  3: Social behaviour          4         No
  4: Other behaviours          2         No

Number of modifiers .........: 4

Labels on event recorder ....: Yes        Label length ......: 4

Events in class 1: Location

        Name              Key     Label    Dura/Freq    Modifiers
  1: In the cage           q      cage     Duration     None
  2: In the yard           w      yard     Duration     None
  3: Near the pool         e      pool     Duration     None

Events in class 2: Locomotory behaviour

        Name              Key     Label    Dura/Freq    Modifiers
  1: Walking               r      walk     Duration     None

↑↓: move around | <spacebar>: select | <Ctrl-End>: save & exit | <ESC>: exit
```

```
03-28-1990                  The Observer                     10:54:02

-------------------- Status --------------------   --- Non-Exclusive Events ---
 Data File         : C:\DATA\MONKEY.ODF                  22.1 shou
 Configuration File: C:\DATA\MONKEY.CNF                  57.6 scra       +
 Free Disk Space   : 8996485 bytes                       60.4 scra       -
 Observation Number: 1                                   64.5 scra       +
 Start Time        : 10:49:11                            76.2 scra       -
 Elapsed Time      : 291.3 sec                          113.9 touc rob
 Maximum Time      : 600.0 sec                          135.8 beat rob
 Next Key          : Event                              150.1 touc pete
                                                        177.5 play pete  +
--------------- Exclusive Events ---------------        225.7 play pete  -
                                                        235.2 shou
 Class                   Event     Since                243.8 play pete  +
 Location                pool      102.3                254.8 groo pete  +
 Locomotory behaviour    sit       191.0                269.9 scra       +
                                                        277.4 shou
                                                        283.5 scra       -
                                                      ▶

F1 Help | F2 Mark | F3 Events On | F4 Freq | F6 Config | F8 Notepad | F10 End
```

Figure 2. (A) Main Menu of The Observer; (B) example of a file selection screen; (C) example of a settings screen; (D) settings screen with definitions of events and modifiers; (E) screen of a PC during a data collection session.

to implement multilevel switches. For example, the class *location* could contain the options "inside" versus "outside," or—as in the example—"in the cage," "in the yard," and "near the pool." Mutually exclusive event keys are operated by only one keypress, in contrast to non-exclusive duration events, which have to be turned on and off by subsequent keypresses. Events in different classes always operate completely independent from each other.

Events. These represent the actual data to be recorded. For each event (Figure 2D), one has to specify:

1. A *name*. Examples: "in the cage," "walking," and "grooming."

2. A *key* (i.e., the key that will be used to signal the event). One can choose from 92 keys (i.e., the total number of alphanumeric and symbol keys on the keyboard minus a few that have a reserved function in the program). In the example, the *g* is used to signal "shout," and the *h* is used for "scratching."

3. A *label*. The user can define descriptive labels for display on the event recorder's screen. If such labels are used, the label corresponding with a key, rather than the key, is displayed on screen and written to the data file. In the example, the *e* key is defined as "near pool," and "pool" is defined as a label for that key.

4. If the class to which the event belongs is nonexclusive (see above), one can label it as a *duration event* or a *frequency event*. The former type refers to events with a meaningful duration; for the latter, only frequency and rate matter. This distinction is similar to what has been referred to as "states" and "events" (Altmann, 1974) or "duration meaningful" versus "momentary" behaviors (Sackett, 1978). Duration events have to be turned on and off during operation of the event recorder; for frequency events, each keypress signals an occurrence. Mutually exclusive events, by definition, always have a duration. In the example, "beat," "touch," and "shout" are frequency events, while all others are duration events.

5. The *number of modifiers* connected to the event. In a configuration with modifiers (see definition below), any event in a nonexclusive event class can have zero, one, or two modifiers. If a key representing an event with modifier(s) is pressed, the event recorder will prompt for the modifier(s) before accepting the next event key. In a group observation, certain events could have two modifiers, the first one indicating the initiator and the second one the recipient of the behavior. In the example, all social behaviors have one modifier indicating the recipient of the interaction.

Modifiers. These are keys that are used to indicate the limits of an event. For instance, in observations of social interaction, one may want to record not only the behavior of a focal subject, but also the recipient of certain behaviors. Modifiers can also be used to record attributes of behavior, such as intensity or direction. Any number of keys can be defined as modifiers, as long as the total number of events and modifiers does not exceed 92. In the example, "beat," "touch," "playing with," and "grooming" are events; "John," "Rob," "Pete," and "Charlie" are modifiers that are used to indicate who is beaten, touched, played with, or groomed. For each modifier, one has to specify (1) name (e.g., the name of a subject in a group observation; in the example, "John," "Rob," etc.), (2) key, and (3) label.

Independent variables. One can store the values of up to 40 independent variables together with the observational record. The variables are entered in the form of questions to be asked by the event recorder during operation, either before or after observations. The answers are then stored in the data file (examples: name, sex, age, stimulus, treatment, observer, and environmental conditions).

Because of the flexibility and the practically unlimited number of combinations of classes, events, modifiers, and independent variables, the user can tailor the event-recording configuration precisely to experimental designs, ranging from studies with few behavioral categories to complicated group observations. This approach also makes optimal use of the selected hardware because the user can choose desirable optimizations (e.g., with respect to the way data are stored). This is of particular relevance if a non-IBM-compatible portable computer is used as the event recorder. For example, the use of descriptive labels for keys makes event recording easier and helps to prevent errors, but it also slows down program execution on those computers, which may be unacceptable in experiments with events occurring in rapid succession. Execution speed is also affected by the number of keys and classes defined. Furthermore, the use of many independent variables allows the user to store a lot of additional information, but each extra question takes up memory, which may become a limiting factor. However, since all configuration options are facultative, the user can decide where to compromise and where not to.

The Observer as a Program Generator

The configuration file contains all the information entered by the user, as described in the previous section, in a coded form (Appendix A). The contents of this file are sufficient to let the PC itself operate as an event recorder. However, non-IBM-compatible portable computers need separate event-recording programs. The Observer can generate such programs on the basis of an event-recording configuration (Figure 1). Programs written by The Observer are specifically adapted to the machine and experimental design chosen by the user. They contain only essential program code and occupy a minimum amount of memory in the portable computer. The sizes of these generated programs vary between 4K and 10K. As a result, complex behavioral observations can be recorded on small portable computers with limited memory and processing speed. For instance, The Observer is used for focal sampling of social interactions in a group of about 20 rhesus monkeys, where a continuous record is made of the occurrence of about 40 different events (varying in status as duration or frequency event, exclusive or nonexclusive event, with or without modifiers), with a Tandy 102 used as event recorder. The program generated by The Observer for this study occupies 7K (M. L. Boccia, personal communication, June 1990). Event-recording programs written by The Observer are in BASIC or OPL. However, the BASIC dialects used by the various computers are quite different, which makes the choice of event recorder crucial: an event-recording program written for the Tandy 102 will not run on an Epson PX-4, and vice versa. The Observer transfers generated event-recording programs from the PC to the portable computer via the RS-232 interface. The user is guided through this process in a step-by-step manner.

Operation of the Event Recorder

Timing of events. Computer timing of events has received considerable attention in research fields where accurate timing is an absolute necessity, such as operant conditioning or stimulus-response studies. Several software timers with millisecond resolution have been designed for different types of computers (e.g., Coyne, 1987; Graves & Bradley, 1988; Perone, 1985). However, The Observer has primarily been developed for direct-observation studies in which there is no need for sub-0.1-sec timing resolution, because of the reaction time of the user. Therefore, times are rounded to 0.1 sec. On the PC, this is easily accomplished with QuickBASIC's TIMER function, which returns real time accurate to ⅟₁₈ sec. The BASIC interpreters on the non-IBM-compatible portable computers, however, only have a TIME$ function that returns time in whole seconds, while events may have to be recorded in more rapid succession. Therefore, programs generated by The Observer for non-IBM-compatible computers contain a loop that counts the number of fractions of a second elapsed between subsequent keystrokes. These loops have been calibrated for the various models. With the processing time for each keystroke taken into account, an accuracy of 0.1 sec can be reached.

The current design of the software does not rule out all possible factors that may affect timing accuracy, such as differences in timing between individual keys on the keyboard, uncontrolled string space reorganization or disk access during operation, and the presence of memory-resident (TSR) programs. However, as far as the author is aware, none of these appears to affect the performance of the program at the 0.1-sec accuracy level.

Event recording. On the PC, an event-recording session takes place within the main program environment and is started by loading a configuration file. On the non-IBM-compatibles, event-recording programs are dedicated (i.e., generated for a particular experimental design). In spite of this structural difference, operation of the event recorder—from the user's standpoint—is practically independent of the type of hardware used.

Prior to the first observation, the user can enter comments that are stored together with the data. Initial values of independent variables can be set. Next, the program will prompt the user to start the observation by pressing the first event key. At each valid keypress, the program calculates the time elapsed since the start of the observation and stores this and the key label. On the PC, data are written continuously to disk.

If the user presses a key representing an event to which one or more modifiers are attached, the program prompts for the appropriate key(s). If a nonexclusive *duration* event key is pressed for the first time, the event is turned on and is stored together with a plus sign (+). At the next depression of the same key, it is stored with a minus sign (−) and thus turned off (Appendix B).

After the observation has been ended by the user (or by the program), the program displays a table with additional independent variables, if such were defined. Subsequently, one can proceed with the next observation or leave the program. In the first case, any of the independent variables can be edited if necessary, after which the next observation follows. After terminating the event-recording session, the user has access to the data file produced by the program. Since there is no on-line data reduction, the complete event record is available for subsequent analysis (Appendix B).

If a non-IBM-compatible portable computer is used, data files can be stored temporarily on disk or tape. For permanent storage and analysis, The Observer uploads files to the PC via the RS-232 interface. During file transfer, the user can simultaneously obtain a printout of the file.

Screen display and interrupt functions. An effort has been made to provide the user with appropriate visual feedback during data collection. During an observation, the computer's screen displays the following items (Figure 2E):

- Current date and real time.
- Number of the current observation.
- Time at which the current observation started.
- Time elapsed since the start of the observation.
- Amount of free disk space. This is updated after each keystroke. Data are not compressed; the amount of space (which varies between 7 and 14 bytes) needed to store all the information associated with a keystroke depends on the data-storage method selected by the user (speed or space), the way the key has been defined (exclusive/nonexclusive event, duration/frequency), and whether or not labels are used (and the label length).
- Names of the exclusive-event classes, with, for each class, the active event and the time at which it started. By definition, there is always one active event for each exclusive-event class.
- Nonexclusive events (labels or keys). These scroll in a window on the screen. They are preceded by the time of occurrence and can be followed by one or more modifiers and/or an on/off code (+ or −).

The bottom line of the screen displays a number of labels indicating the following function-key interrupts (depending on hardware and event-recording configuration):

- *On-line help*. This function displays an explanatory text about the use of the various keys during event recording.
- *Error marker*. This function is explained below.
- *Status of duration events*. In contrast with mutually exclusive events, of which only one can be active per class, nonexclusive events can occur at any time. Those events scroll off the screen, so that after a while, the user may no longer know whether a certain event is on or off. For that purpose, a function key can be used to display a list of all duration events (in combination with modifiers) that are on.
- *Event frequencies*. This function displays a list of events (in combination with modifiers) that have occurred and their frequency of occurrence, up to date to the last keypress.
- *Configuration*. This function triggers a window with the definitions of keys as events or modifiers.
- *Notepad*. With this function, the user has access to an electronic notepad that can be called at any time. As soon as the notepad key is pressed, a memory-resident editor (part of the program) is invoked and text can be entered, which is stamped with the (elapsed) time at which the notepad was invoked. This feature can be used for irregular notes during observations. It can also be used as a checksheet for any form of time sampling without interrupting the continuous real-time record.
- *End of observation*. With this function key, the user can terminate the observation.

On the PC, the information is displayed in pop-up windows. Information functions should be invoked only at times during a session that the user can direct his/her attention away from the subject under observation.

Error handling. Two types of errors can occur during observations. The user may press a key that has not been defined as an event (or modifier, if the program is waiting for one). This is detected by the program, which displays an error message and prompts the user to press a valid key. As soon as a valid key has been pressed, the *initial* time (upon which the invalid key was pressed) together with the *valid* event are stored. This type of error can be prevented by using a (e.g., cardboard) keyboard overlay that restricts user access to the defined keys.

One may also erroneously press a valid key while wanting to press another one. This error cannot be trapped by the program. As soon as the user realizes the mistake, the error-marker function key should be pressed, upon which a marker is written to the data file. The correct key can then be pressed and the observation can continue normally. The error can later be retrieved and corrected.

Data Analysis

Many high-quality general-purpose software packages for statistical analysis on personal computers are available. Therefore, The Observer 2.0 has limited analysis capability (which may be sufficient for many users), but will output files in a format compatible with more advanced analysis packages.

Observational data files (Appendix B) are plain ASCII files with a structure entirely independent of the type of computer used as event recorder (PC or non-IBM-compatible). They contain the name of the configuration file that describes the event-recording configuration with which the data were collected. This file is linked to the data file during analysis. In this way, The Observer can analyze any kind of data file, given that it has access to the corresponding configuration file. After a data file has been loaded, the user can select the level of analysis and the type of output files. Statistics calculated by The Observer include frequency of occurrence and duration (total, mean, and standard deviation of the mean) (Appendix D). These can be calculated for:

1. *Classes of events*. Example: "total time spent on *social behaviors* versus *aggressive behaviors*."

2. *Events* (in combination with modifiers, if appropriate). Examples: "total time spent *grooming*," "average duration of a *walking* bout," "frequency of *shouting*," and "total time spent *grooming Pete*."

3. *Combinations of events*. This refers to a nested analysis, applicable to configurations with one or more exclusive-event classes. The program then calculates concurrences, with up to seven levels of nesting: four exclusive-event classes, non-nested events, and two levels of modifiers can be included in the analysis. Examples: "frequency of *walking* while *in the cage*," "frequency of *grooming Charlie* (event + modifier) while *sitting* (exclusive event) *in the yard* (exclusive event)."

The user can further select:

- The *number of nesting levels*. For example, if the configuration includes three classes of exclusive events, and a number of nonexclusive events, not all exclusive-event classes have to be included in a nested analysis. If one selects one class, the events of the other classes are treated together with the nonexclusive events as non-nested events.
- The *order of nesting*. In the example with classes *location* and *locomotory behavior*, one can obtain an analysis of all events "by location * locomotory behavior" or "by locomotory behavior * location." Appendix D shows the effect of the order of nesting on the arrangement of statistics.

The user can choose any (or all) of the following output files:

1. *Sequence file*. This is a file containing the complete event protocol listing, for each observation, with exclusive events (if those were defined) sorted per class (Appendix C). This format is suitable for analysis of sequences of events. The sequence file can be transferred to other programs for further analysis.

2. *Report file*. This is a text file (Appendix D) containing the configuration of the event recorder, initial user comments, a report for each observation in the data file (with independent variables), notes made during the observation (with the time at which they were made), and calculated statistics—according to the type(s) of analysis selected by the user.

3. *Spreadsheet-import file*. This file contains independent variables and calculated frequencies and durations in a compact tabular format suitable for direct import into a spreadsheet program (e.g., Lotus 1-2-3) or a statistics package (SAS, SPSS, etc.). The format is plain ASCII with text labels between double quotes and either commas or tabs (user-definable) as delimiters between cell entries. A spreadsheet format facilitates the calculation of summary statistics pooled across observations, contingency analysis, and so on. The spreadsheet can also serve as an interface with more advanced statistical or graphics software.

During initial analysis, the data file is checked for possible errors (error markers or editing errors). If errors are detected, a relevant error message is displayed, analysis is abandoned, and the user has to edit the data file before it can be analyzed successfully. Any editor or word processor can be called from inside The Observer, as long as the computer has sufficient memory.

COMPARISON WITH OTHER SOFTWARE

Another published software package for recording of real-time events on the IBM PC is EventLog by R. W. Henderson (available through Conduit, University of Iowa). There are a few technical differences between The Observer and EventLog. Unlike The Observer, EventLog does not support non-IBM-compatible portable computers, nor does it include any data analysis. Both produce data files suitable for export to statistical packages.

The most obvious difference between EventLog and The Observer lies in the way events are entered by the user and timed by the program. With EventLog, the user signals the start of an event by pressing the key representing that event and keeping it pressed down for the full duration of the event, which is then calculated by the program. This has several implications. The number of events that can occur simultaneously is limited by the number of keys that can be pressed down by the user (e.g., 10 when using 10 fingers), and the method appears quite inconvenient if events last several minutes or longer. Furthermore, all events are automatically nonexclusive duration events (as defined above). Finally, program functions apart from data entry (such as entry of notes) temporarily halt event recording and timing. With The Observer, the way an event is entered depends on how it has been defined by the user. As described above, events can be categorized as exclusive or nonexclusive events, frequency or duration events, and modifiers can be attached to events. Since events are started with a single keystroke, the number of events that can occur simultaneously is only limited by the number of duration events that have been defined (multiplied with the number of modifiers, if such were defined). All extra program functions (e.g, the notepad, on-line statistics, etc.) can be accessed without effect on event recording.

EventLog uses scan codes to convey information from the keyboard to the program and bypasses the system clock to time events. By doing so, it can achieve millisecond accuracy. The Observer, in contrast, uses standard DOS routines to scan the keyboard and to retrieve the system time. This limits The Observer's timing accuracy to about 0.1 sec (see above). In contrast to EventLog, The Observer's performance—as far as the program's 0.1-sec timing accuracy is concerned—is independent of the type of PC used (different PCs produce scan codes in different ways) and remains unaffected by memory-resident (TSR) software.

EventLog stores data in memory, which has the advantage that timing of rapidly occuring events is not affected by disk access. However, in the event of a power failure, all data are lost. In contrast, The Observer writes data continuously to a file on disk. With most modern disk drives, this does not affect timing accuracy at the 0.1-sec level. However, if desired, data can be stored on a virtual disk in RAM.

SUMMARY

The Observer has been used for more than 3 years and has been adopted as a useful tool for research and educational purposes by many research groups. Due to the flexibility of the system, most experimental designs can be handled. Also, the user can simply alter the experimental design or sampling method or change to other hardware, while observational data remain compatible. The Observer has been used for studies of the behavior of adult and infant humans and for lab or field studies on a wide variety of animal species (primates, carnivores, ungulates, rodents, other mammals, song birds, birds of prey, penguins, various species of fish, mollusks, mites, spiders, and many species of insects).

The system is in continuous development. The IBM PC has been chosen as host computer; however, it may be desirable to port The Observer to the Apple Macintosh or to UNIX-based systems in the near future or to develop program generators for other types of hand-held computers. In a future version, existing functions will be improved (e.g., optimization of timing routines for sub-0.1-sec timing accuracy) and new features will be added (e.g., alternative data-entry methods, interobserver agreement analysis, sequence analysis).

The Observer is available in the form of a Base Package, which includes all the software necessary for data collection and analysis on the PC, and Support Packages for non-IBM-compatible portable computers, each of which contains a program generator, file upload and download programs, and sample applications for one or more related models. The software is not copy-protected. It is distributed on the basis of single-user or site licenses. Each package includes the software on 3.5- or 5.25-in. disks, a printed user's manual, access to upgrades, and unlimited free technical support (via mail, phone, fax, or e-mail). The manual includes a theoretical introduction, a comprehensive reference section covering all the system functions, instructions for advanced data analysis with a number of popular spreadsheet programs, a glossary of technical terms, technical information, and a bibliography. In addition, it contains a series of sample applications in which coding schemes are developed and data are collected and analyzed in a step-by-step manner, with increasing complexity as the reader goes along. This section has been written for the observation novice and can be used as a tutorial on the use of the system. The current price of a single-user license is $1,950 (commercial organizations) or $895 (nonprofit/educational institutions). Prices are subject to change without notice. For more information, readers should contact the author.

REFERENCES

Altmann, J. (1974). Observational study of behavior: Sampling methods. *Behaviour*, **49**, 227-267.

Berman, J. (1983, July). Tandy's towering totable. *80 Micro*, pp. 158-168.

Berry, J. S., Holtzer, T. O., & Norman, J. M. (1987). Computers in the field. *Bulletin of the Entomological Society of America*, **33**, 209.

Coyne, A. C. (1987). A software-based millisecond timer for the CP/M operating system. *Behavior Research Methods, Instruments, & Computers*, **19**, 47-48.

Deni, R., Szuarto, K., Eisler, A., & Fantauzzo, C. (1983). BASIC programs for observational research using the TRS-80 Model 100 portable and Model 4 computers. *Behavior Research Methods, Instruments, & Computers*, **15**, 616.

Gilreath, J. P. (1985). Description of a BASIC program for data collection using a portable microcomputer. *HortScience*, **20**, 301.

Graves, R., & Bradley, R. (1988). More on millisecond timing and tachistoscope applications for the IBM PC. *Behavior Research Methods, Instruments, & Computers*, **20**, 408-412.

Horner, R. H., & Storey, K. (1989). Putting behavioral units back into the stream of behavior: A consumer report. *Behavior Therapist*, **12**, 249-251.

Lehner, P. N. (1979). *Handbook of ethological methods*. New York: Garland STPM Press.

Martin, P., & Bateson, P. (1986). *Measuring behaviour: An introductory guide*. Cambridge: Cambridge University Press.

Noldus, L. P. J. J. (1989). *Chemical espionage by parasitic wasps.* Ph.D. Thesis, Wageningen Agricultural University, Wageningen, The Netherlands.

Noldus, L. P. J. J., van de Loo, E. L. H. M., & Timmers, P. H. A. (1989). Computers in behavioural research. *Nature*, **341**, 767-768.

Perone, M. (1985). A software system for real-time use of TRS-80 microcomputers. *Behavior Research Methods, Instruments, & Computers*, **17**, 119-121.

Sackett, G. P. (1978). Measurement in observational research. In G. P. Sackett (Ed.), *Observing behaviour. Vol. 2: Data collection and analysis methods* (pp. 25-43). Baltimore, MD: University Park Press.

Unwin, D. M., & Martin, P. (1987). Recording behaviour using a portable microcomputer. *Behaviour*, **101**, 87-100.

Whiten, A., & Barton, R. A. (1988). Demise of the checksheet: Using off-the-shelf miniature hand-held computers for remote fieldwork applications. *Trends in Ecology & Evolution*, **3**, 146-148.

APPENDIX A

Social behavior of monkeys: A hypothetical case study illustrating the use of The Observer.

The study group consists of 4 monkeys (John, Rob, Pete, and Charlie) in an enclosed area, consisting of a cage, a yard, and an outdoor pool. Observations are 10-min focal samples of 1 monkey. Prior to an observation, a record is made of name, sex, age, treatment code, and group number. During the 10-min period, location and behavior of the focal subject are recorded according to the classification below.

The configuration specified by the user is saved in a file on disk (MONKEY.CNF). In a real configuration file, the information is stored in a more condensed, coded form.

General settings

Computer used as event recorder:	IBM PC or compatible
Maximum time per observation:	10 minutes
Data storage optimization:	Speed
Auditory feedback during event recording:	No
Disk checking during event recording:	No
Modifiers:	Yes
Labels for events:	Yes
Label length:	4

Class	**Exclusive**[1]	**Event**	**Key**	**Label**	**Dur/Freq**[2]	**Modifiers**
Location	Yes	In the cage	q	cage	Dur	0
		In the yard	w	yard	Dur	0
		Near the pool	e	pool	Dur	0
Locomotory behavior	Yes	Walking	r	walk	Dur	0
		Standing still	t	stan	Dur	0
		Sitting	y	sit	Dur	0
Social behavior	No	Beat	a	beat	Freq	1
		Touch	s	touc	Freq	1
		Playing with	d	play	Dur	1
		Grooming	f	groo	Dur	1
Other behaviors	No	Shout	g	shou	Freq	0
		Scratching	h	scra	Dur	0

Modifier	**Key**	**Label**
John	1	john
Rob	2	rob
Pete	3	pete
Charlie	4	char

Independent variables
Name
Sex
Age
Treatment code
Group
Comments

[1] Yes = mutually exclusive events; No = non-exclusive events

[2] Freq = frequency (point) event; Dur = duration event

APPENDIX B

Monkey.ODF: Example of an *observational data file* produced with The Observer. The file is the result of a 10-min focal sample. The comments in the lines (after the quote) are not included in real data files. For meaning of codes, see Appendix A.

```
@CF,MONKEY.CNF                           'name of configuration file
@CO,Social behavior of rhesus monkeys    'initial comment
  0.0 yard                               'start observation, in yard
  0.0 walk                               'starts walking
 22.1 shou                               'shouts
 36.0 cage                               'in cage
 49.6 stan                               'stands still
 57.6 scra,+                             'starts scratching
 60.4 scra,-                             'stops scratching
 64.5 scra,+
 76.2 scra,-
 85.5 walk                               'starts walking
 95.6 yard                               'enters yard
102.3 pool                               'near pool
113.9 touc,rob                           'touches Rob
128.7 stan
135.8 beat,rob                           'beats Rob
145.7 walk
150.1 touc,pete
159.1 stan
177.5 play,pete,+                        'starts playing with Pete
191.0 sit                                'sits
225.7 play,pete,-                        'stops playing with Pete
135.2 shou
243.8 play,pete,+
254.8 groo,pete,+                        'starts grooming Pete
269.9 scra,+
277.4 shou
283.5 scra,-
360.3 groo,pete,-
398.1 play,pete,-
452.7 shou
565.4 walk
600.0 {eo}                               'end of observation
@DT,03-28-1990,10:49:11                  'date and time
@QB,1,John,male,3,trl,2                  'independent variables: obs.no.=1, name=John, etc.
@QA,no                                   'indep. var. (question after obs.): comments=no
@NO,291.3
@NO,falls                                'note made 291.3 sec after start
@NO,420.1
@NO,picks up nut                         'note made at t=420.1
@NO,{en}                                 'end of notes
```

APPENDIX C

MONKEY.SEQ: Example of a *sequence file* produced by The Observer. This file contains two columns for the exclusive-event classes *location* and *locomotory behavior*, followed by the other (nonexclusive) behaviors. The latter are followed by a modifier and/or an on/off-code (+/−) if appropriate. For meaning of codes, see Appendix A.

```
@CF,MONKEY.CNF
@CO,Social behavior of rhesus monkeys
     0.0 yard, walk
    22.1 yard, walk, shou
    36.0 cage, walk
    49.6 cage, stan
    57.6 cage, stan, scra, +
    60.4 cage, stan, scra, -
    64.5 cage, stan, scra, +
    76.2 cage, stan, scra, -
    85.5 cage, walk
    95.6 yard, walk
   102.3 pool, walk
   113.9 pool, walk, touc, rob
   128.7 pool, stan
   135.8 pool, stan, beat, rob
   145.7 pool, walk
   150.1 pool, walk, touc, pete
   159.1 pool, stan
   177.5 pool, stan, play, pete, +
   191.0 pool, sit
   225.7 pool, sit, play, pete, -
   235.2 pool, sit, shou
   243.8 pool, sit, play, pete, +
   254.8 pool, sit, groo, pete, +
   269.9 pool, sit, scra, +
   277.4 pool, sit, shou
   283.5 pool, sit, scra, -
   360.3 pool, sit, groo, pete, -
   398.1 pool, sit, play, pete, -
   452.7 pool, sit, shou
   565.4 pool, walk
   600.0 {eo}
@DT,03-28-1990,10:49:11
@QB,1,John,male,3,trl,2
@QA,no
@NO,291.3
@NO,falls
@NO,420.1
@NO,picks up nut
@NO,{en}
```

APPENDIX D

MONKEY.REP: Example of a *report file* produced by The Observer (part). The file is the result of analyzing the data file MONKEY.ODF (Appendix B). The configuration is represented in Appendix A.

```
********************************************************************************

Data file analyzed          : MONKEY.ODF
Linked configuration file : MONKEY.CNF
Data file comment           : Social behavior of rhesus monkeys
Date of analysis            : 03-29-1990
Analysis comment            : Nested analysis

********************************************************************************

Date                            : 03-28-1990
Start of observation            : 10:49:11 h
Total duration of observation : 600 sec
Observation no.                 : 1
Name                            : John
Sex                             : male
Age                             : 3
Treatment code                  : trl
Group                           : 2
Comments                        : no

--------------------------------------------------------------------------------

NOTES MADE DURING OBSERVATION

TIME     NOTE
291.3    falls
420.1    picks up nut

--------------------------------------------------------------------------------

FREQUENCIES AND DURATIONS, LUMPED PER EVENT CLASS
(for classes with frequency > 0)

DURATION EVENTS

CLASS                             N       DURATION     DURATION    DURATION
                                           (TOTAL)       (MEAN)        (SD)

Location                          1          600.0       600.00        0.00
Locomotory behavior               1          600.0       600.00        0.00
Social behavior                   2          202.5       101.25       75.02
Other behaviors                   3           28.1         9.37        5.77


FREQUENCY EVENTS

CLASS                             N       DURATION     DURATION    DURATION
                                           (TOTAL)       (MEAN)        (SD)

Social behavior                   3              -            -           -
Other behaviors                   4              -            -           -

--------------------------------------------------------------------------------
```

APPENDIX D (Continued)

```
FREQUENCIES AND DURATIONS, LUMPED PER EVENT
(for events with frequency > 0)

EVENT                  MODIFIER              N  DURATION  DURATION  DURATION
                                                 (TOTAL)    (MEAN)      (SD)
(Location)
In the cage            -                     1      59.6     59.60      0.00
In the yard            -                     2      42.7     21.35     20.72
Near the pool          -                     1     497.7    497.70      0.00

(Locomotory behavior)
Walking                -                     4     140.8     35.20     15.78
Standing still         -                     3      84.8     28.27      9.96
Sitting                -                     1     374.4    374.40      0.00

(Social behavior)
Beat                   Rob                   1         -         -         -
Touch                  Rob                   1         -         -         -
                       Pete                  1         -         -         -
Playing                Pete                  2     202.5    101.25     75.02
Grooming               Pete                  1     105.5    105.50      0.00

(Other behaviors)
Shout                  -                     4         -         -         -
Scratching             -                     3      28.1      9.37      5.77

--------------------------------------------------------------------------------

FREQUENCIES AND DURATIONS, FOR COMBINATIONS OF EVENTS
(for combinations with frequency > 0)

ORDER OF NESTING: Location * Locomotory behavior

LEVEL 1    LEVEL 2     NON-NESTED  MODIFIER     N  DURATION  DURATION  DURATION
LOCATION   LOCOMOTORY  EVENT                        (TOTAL)    (MEAN)      (SD)

In cage    Walking     -           -            2      23.7     11.85      2.47
           Standing    -           -            1      35.9     35.90      0.00
                       Scratching  -            2      14.5      7.25      6.29
In yard    Walking     -           -            2      42.7     21.35     20.72
                       Shout       -            1         -         -         -
Near pool  Walking     -           -            3      74.4     24.80     10.69
                       Touch       Rob          1         -         -         -
                                   Pete         1         -         -         -
           Standing    -           -            2      48.9     24.45     10.54
                       Beat        Rob          1         -         -         -
                       Playing     Pete         1      13.5     13.50      0.00
           Sitting     -           -            1     374.4    374.40      0.00
                       Playing     Pete         2     189.0     94.50     84.57
                       Grooming    Pete         1     105.5    105.50      0.00
                       Shout       -            3         -         -         -
                       Scratching  -            1      13.6     13.60      0.00

--------------------------------------------------------------------------------
```

APPENDIX D (Continued)

```
ORDER OF NESTING: Locomotory behavior * Location

LEVEL 1    LEVEL 2     NON-NESTED  MODIFIER     N  DURATION  DURATION  DURATION
LOCOMOTORY LOCATION    EVENT                       (TOTAL)    (MEAN)      (SD)

Walking    In cage     -           -            2      23.7     11.85      2.47
           In yard     -           -            2      42.7     21.35     20.72
                       Shout       -            1        -         -         -
           Near pool   -           -            3      74.4     24.80     10.69
                       Touch       Rob          1        -         -         -
                                   Pete         1        -         -         -
Standing   In cage     -           -            1      35.9     35.90      0.00
                       Scratching  -            2      14.5      7.25      6.29
           Near pool   -           -            2      48.9     24.45     10.54
                       Beat        Rob          1        -         -         -
                       Playing     Pete         1      13.5     13.50      0.00
Sitting    Near pool   -           -            1     374.4    374.40      0.00
                       Playing     Pete         2     189.0     94.50     84.57
                       Grooming    Pete         1     105.5    105.50      0.00
                       Shout       -            3        -         -         -
                       Scratching  -            1      13.6     13.60      0.00

********************************************************************************
```

(Manuscript received September 18, 1989;
revision accepted for publication February 28, 1991.)