NORSAR ROYAL NORWEGIAN COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

Scientific Report No. 2-84/85

SEMIANNUAL TECHNICAL SUMMARY

1 October 1984 - 31 March 1985

L. B. Loughran (Ed.)

Kjeller, July 1985

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

VII.9 The RST/RSTE research project - current status

The RST/RSTE (Remote Seismic Terminal Enhanced) project is aimed at developing an intelligent seismic field terminal with capabilities of detecting and analyzing events in an automated mode. The operational capabilities would be roughly equivalent to those of the dedicated data center for a large-aperture array, although the number of channels to be handled would be far less. This entails a multitude of tasks like real-time event processing, off-line processing of detected events, and external transfer to an RST of raw and processed signal data. The viability and realizability of the RST/RSTE concept was demonstrated in Geneva in 1982 (e.g., Husebye and Thoresen, 1984); for a schematic view, see Fig. VII.9.1.

The mentioned operational task, say for 8 short period seismometers, can be "accomodated" with a hardware framework of mini -or personal computers like the IBM PC AT. The corresponding software implementation requires a highly efficient multitasking operating system, as well as a high level programming language. Our choice was polyFORTH II, which is an integrated operating system/high level language (FORTH). The detection process is a voting one tied to STA/LTA thresholding on individual channels - a Walsh detector replacement is under consideration. In the event processor detected signals would be subjected to f-k analysis, and a suitable phase picker for signal onset times. Important, in view of the very recent 3-comp. seismogram analysis results reported in Section VII.7, a viable alternative here would be to replace the mini-array with 3-comp. high-quality instrumentation. It should be added that recent progress in fault mechanism studies entail that reasonable moment tesnor solutions are feasible in many cases, even when single 3-comp. station records are used (prof. A. Dziewonski, personal communication).

Seismological motivations for RST/RSTE

Theoretical advances in seismology cannot be sustained unless researchers have easy access to an abundance of high-quality seismic data in digital form. The NORSAR array and the SRO and ASRO seismic observatories are typical examples of realizations of these data collection goals. On the other hand, most seismometers in operation are tied to analog recording systems (photographic paper to be changed daily). In other cases the seismometer outputs are transmitted continuously say via dedicated telephone lines to a data center, although only a few per cent of the data flow contains significant event information. Also, the use of small arrays/3-comp. stations will very much facilitate signal analysis, that is, Level 1 data extraction, both qualitatively and quantitatively. The obvious thing to do here is naturally to decide at the seismometer site which part of the recordings are useful, and in this way greatly reduce excessive transmission costs. Likewise, extraction of Level 1 data in the field would permit much faster and more advanced event analysis with a minimum of analyst interaction and delays. It should be noted that if mass data storage in the field is found desirable by any user, this can be achieved by installing minitape recorders. For example, makes currently available can store about 70 Mbytes of data (cost ~ $3000), while most recent models can store ca 130 Mbytes (at the same price) which is equivalent to the standard 9T 6250 BPI computer tape.

We feel confident that our hardware/software designs of the RST and RSTE to be detailed in the next chapters will accomplish these goals. Finally, let us add that in the case of broad-band seismograph stations the data recording is primarily of interest for research and in such cases the RST/RSTE would serve as a data logger system.

RSTE and RST software developments

Basically, our intention is to make the RSTE a smart peripheral device interfaceable to a variety of host computers or RSTs. Its main task is time series analysis in the field and data transfer on the basis of intelligent handling of outputs from a variety of geophysical instrumentation, like small arrays of short period seismometers, hydrophone arrays and 3-component broadband seismometers. Naturally, we do not intend to incorporate all possible functions and tasks in one RSTE, instead aim at designing a flexible software framework from which special purpose RSTEs can be realized relatively easily. For example, seismic data recording and subsequent analysis are real-time multitasking processes which motivated the choice of polyFORTH (Whitney and Conrad, 1983) as programming language. This is an integrated operating system/high level language with minimal overhead for multitasking. Likewise, the choice of the KERMIT protocol for two-way RST-RSTE communication was motivated by its extensive use in universities and research institutions (e.g., see DaCruz and Catchings, 1984a,b).

The RSTE functional structure as realized in the software is shown in Fig. VII.9.2. All boxes in the figure represent independent and concurrent processes, each having a well-defined function and interface. These concurrent processes are executed and synchronized by the Round Robin multitasker of polyFORTH on a single processor. The arrows visualize the communication between processes, with the vertical line in the middle representing a "software bus". All processes connected to this software bus may "talk" to one another. The actual implementation of inter-process communication is done using "mailboxes". Each process has an associated mailbox where other processes may put messages. Details on the RSTE software processes shown in Fig. VII.9.3, most of them now completed, are given below.

KERMIT

This process implements the communications channel to the outside world, using a modified/extended version of the KERMIT protocol. The extensions include having KERMIT running permanently in server-mode, login procedure, long packets (up to 255 bytes) and transfer of 8-bit data without quoting. File transfers are handled in cooperation with the File System. Commands received through KERMIT are handed over to the "Command Interpreter" for parsing. The KERMIT protocol is implemented as a "finite state machine", and an example of state description of the protocol is illustrated in Fig. VII.9.4. It is a pleasure to note that a seismic data logger under development at Harvard University will feature a KERMIT protocol very similar to ours (J. Steim, personal communication).

MODEM

This is the interface between KERMIT and the hardware used for communications. MODEM accepts commands (DIAL, DISCONNECT, INITIALIZE, ...) and data-packets from KERMIT, while sending messages (incoming call, carrier lost, etc.) and data-packets back to KERMIT.

KA

KA is short for "KERMIT Allocator", and is needed because the communications channel cannot be shared. Every process wishing to use KERMIT must first allocate it for exclusive use, and this is done through requests to KA.

COMMAND INTERPRETER

This process is responsible for parsing and execution of commands received through KERMIT.

FILE SYSTEM

The function of this process is obvious: it offers all other processes necessary file-handling functions. Multiple open files are allowed. The file system can be implemented using one or both of RAM (memory) and disk storage. The actual implementation is transparent to the rest of the system. Standard DOS naming conventions are used with up to 8 characters name and 3 characters extension.

CONFIG

Activated by a command from the COMMAND INTERPRETER, this process enables remote configuration of the entire RSTE-system. This includes setting filter parameters, sampling frequency, detection sensitivity, type of post-processing to be used and so on. The required configuration is specified in a file that has been prepared transparently and transferred in advance by the controlling RST.

OPSYS

OPSYS is nothing more than a collection of utilities and system functions needed by other processes.

EVENT FILE DISPATCHER

This process handles dispatching (and possible queuing) of event files that have been declared "interesting" by the EVENT post-processor. The EVENT FILE DISPATCHER has available a phone directory, which is used to dial an RST and transfer the file.

WATCHDOG

The WATCHDOG is responsible for monitoring the state of the system, reporting and logging minor errors. If this process after crash does not restart and external timer, that timer will automatically restart the system.

EVENT POSTPROC

The EVENT post-processor, one of the major RSTE modules, consists of various analysis programs that are run in background on data stored by the DATAFLOW PROCESS. These are data blocks which are part of the output from the real-time event detection operation. The principal task here is to provide a listing of detected events in terms of extracted Level 1 parameters and rough epicenter locations.

TIME

TIME is just an interfae to the available real-time hardware, offering precise absolute and relative time to all processes.

TEST & CALIBRATION

This process, together with some external hardware (oscillator), provides the ability to calibrate seismometers.

DATAFLOW PROCESS

This is the critical module in the RSTE handling system. The seismometer data stream is subdivided into many small concurrent processes, each implementing a part of the data-flow like the multiplexed digitizer, data windowing and filtering, etc.

Comments. The RSTE software system is now nearing completion with the EVENT POSTPROC being a major exception here. The reason for this is that we are currently somewhat ambivalent about whether the "best" solution would be a miniarray or a high-quality 3-comp. installation. The latter alternative is rather attractive not only as regards decomposition of complex recordings but also for simpler field logistics (avoiding stretching of cables, etc.).

RST functional structure

The RST functional structure, shown in Fig. 9, is grossly similar to that of the RSTE. Of course, the RST has no DATAFLOW process and no EVENTFILE DISPATCHER, while the KERMIT/MODEM/KA and FILE SYSTEM perform the same functions as in the RSTE.

There is little concurrency in the RST. Only two tasks are sharing the processor: KERMIT/FILE SYSTEM and the COMMAND INTERPRETER. Boxes to the right of the "software bus" denote subprograms (modules), to be outlined below, that are activated one at a time under control of the COMMAND INTERPRETER, which is responsible for the user interface.

Signal analysis and displays

This module enables the user to do different kinds of analysis on data received from RSTEs. This includes:

- Display of traces with zoom/pan on both color graphics screen and printer/plotter. Use of joy stick/mouse would enable an analyst to re-estimate onset time, signal amplitude and period, etc.

- Spectral, f-k plots and/or 3-comp. wave train decomposition

- Joint display of traces from more than one RSTE.

RSTE configuration utilities

This module is not required to receive/analyze event files, but instead is aimed at easy preparation of configuration files, for use in the remote RSTE.

Device-drivers

Modules named "PRINTER", "PEN PLOTTER" and "GRAPHICS DISPLAY" are simply device-drivers used by the various display functions of ANALYSIS.

Concluding remarks

Our RST/RSTE concepts have evolved from what we may term Personal Seismometry Now, as demonstrated in Geneva in 1982 (Husebye and Thoresen, 1984) to "Seismic Arrays for Everyone" as sketched by Husebye et al (1985). The motivation for undertaking this project is simply the obvious need for the seismological community at large to have easy access to high-quality short period seismic data and without literally being ruined in the process. Similar concerns are shared as regards broadband seismometry which has motivated the establishment of regional and global broadband station networks.

The RST/RSTE system is modular and flexible, and this can be modified/extended to meet special user modifications. An important step in this direction is the planned replacement of the present RSTE microcomputer OMNIBYTE with an IBM PC AT. In the extreme this would entail RST/RSTE to serve either as a data logger or at the other end as a seismological analyst work station.

E.S. Husebye
E. Thoresen, Consultant
J.A. Mathiesen
I.C. Myrvold

References

Allen, R. (1982): Automatic phase pickers: their present use and future prospects, BSSA, 71, 225-242.

Anderson, K.R. (1978): Automatic processing of local earthquake data; Ph.D. thesis, MIT, Cambridge, Mass 02139, USA.

DaCruz, F. and B. Catchings (1984a): KERMIT: A file transfer protocol for universities. Part 1: Considerations and specifications, Byte, June 84.

DaCruz, F. and B. Catchings (1984b): KERMIT: A file transfer protocol for universities. Part 2: States and transitions, heuristic rules, and examples, Byte, July 84, 143.

Goforth, T. and E. Herrin (1981): An automatic seismic signal detection algorithm based on the Walsh transform, BSSA, 71, 1351-1360.

Husebye, E.S. and E. Thoresen (1984): Personal seismometry now!, EOS, 65, 441-442.

Husebye, E.S. and S.F. Ingate (1985): Seismic arrays - a new renaissance, submitted for publication.

Husebye, E.S., E. Thoresen and S.F. Ingate (1985): Seismic arrays for everyone, Terra Cognita, in press.

Ingate, S.F., E.S. Husebye and A. Christoffersson (1985): Regional arrays and optimum data processing schemes, BSSA, in press.

Murdock, J.N. and C.R. Hutt (1983): An algorithm that detects and analyzes earthquake signals - a fast signal detector analyzer, Albuquerque Seismological Lab., U.S. Geol. Survey technical note.

Romanowizc, B. (1984): Project GEOSCOPE, Technical Rep., Inst. de Phys. du Globe, Paris, France.

Smart, E. and E. Flinn (1971): Fast frequency-wavenumber analysis and Fisher signal detection in real-time infrasonic array data processing, Geophy. J. R. Astron. Soc., 26, 279-284.

Whitney, A. and M.C. Conrad (1983): Call FORTH for realtime control programming, Computer Design, April 1983, 81-84.

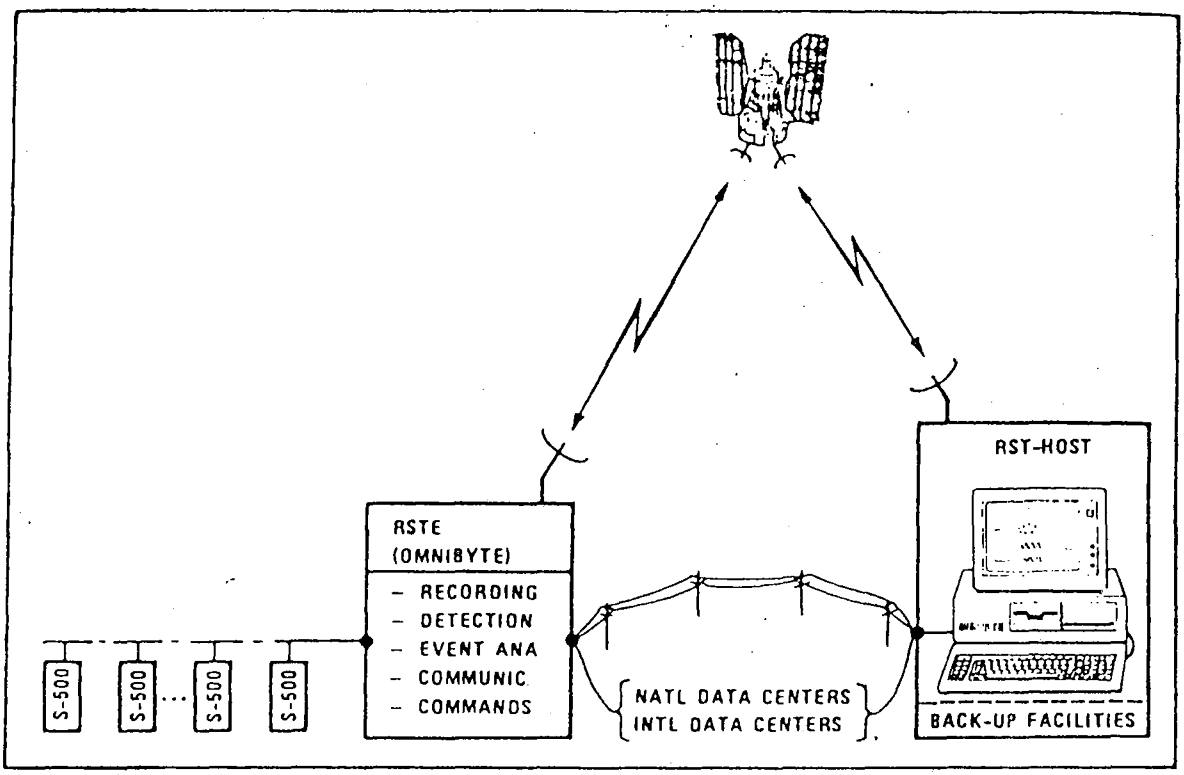

Fig. VII.9.1 Schematic view of the RST/RSTE concept.

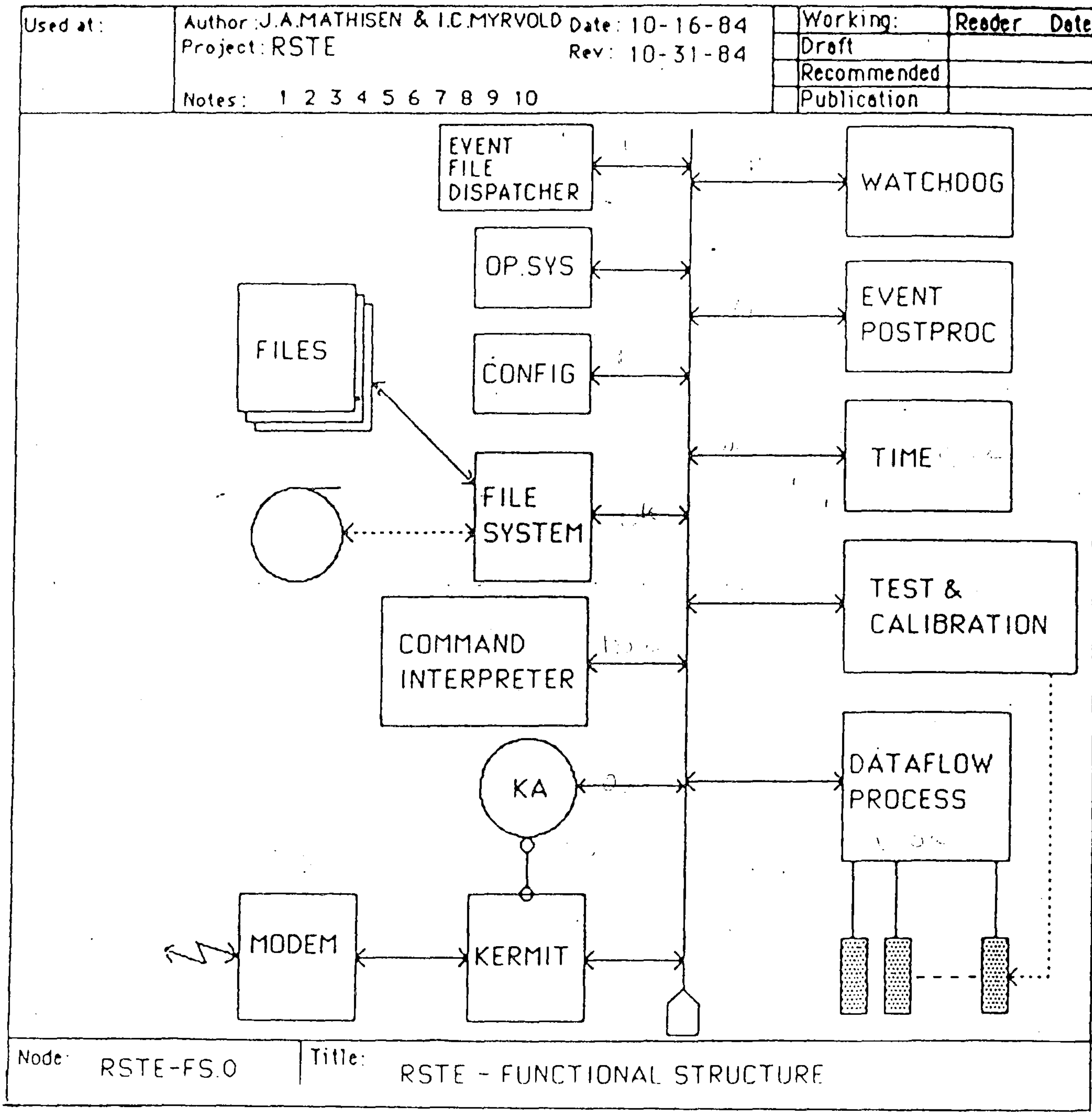

Fig. VII.9.2 The RSTE functional structure. Details on functional tasks are given in the text.

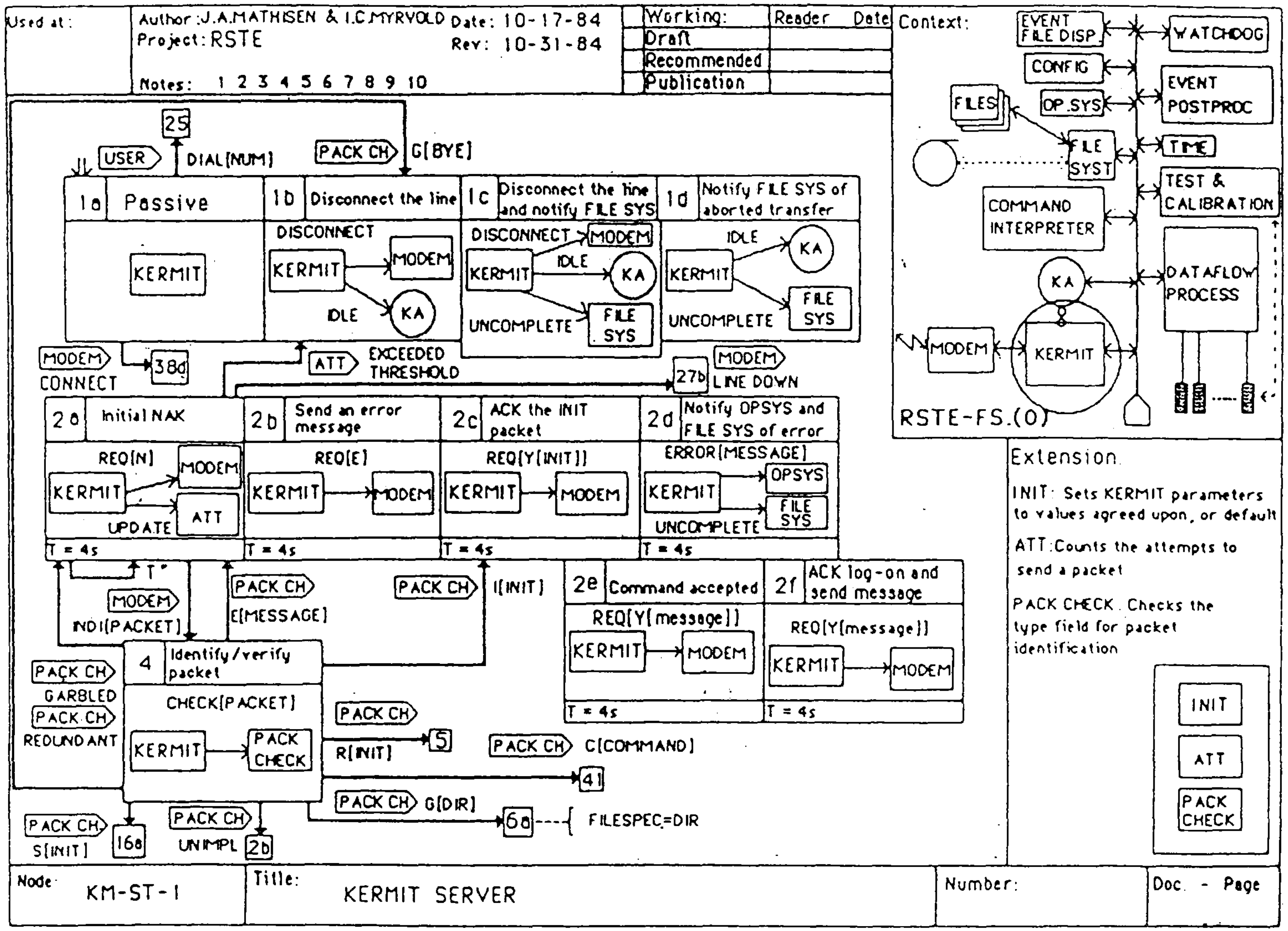

Fig. VII.9.3 One out of ten KERMIT communication protocol state transmission diagrams illustrating its usage in the RSTE.

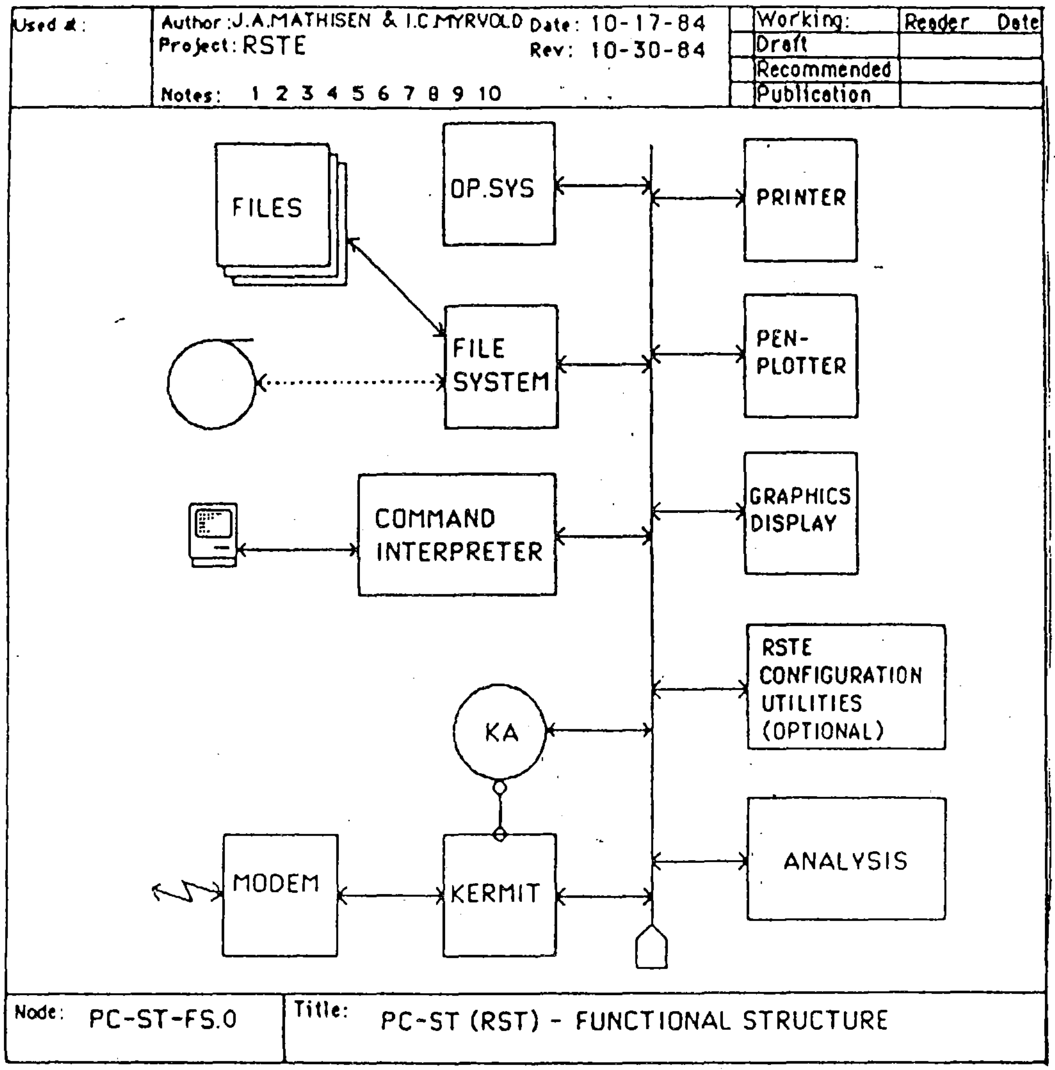

Fig. VII.9.4 The RST functional structure. Details on tasks are given in the text.