CMSR—A personnel information system

J. W. Bridle and R. J. Gregersen

Statistics Division, Civil Service Department

This paper describes a personnel information retrieval system which became operational in the Civil Service Department in June 1970. The system is known as the Central Management Staff Record (CMSR). Information about senior civil servants in all Departments of the Civil Service is stored on a computer. The information is used by central management to assist in vacancy filling, manpower planning and for statistical purposes. The system operates from a remote terminal and uses an information retrieval language in a conversational mode, has the ability to store, retrieve and amend questions 'on line', and is able to edit terminal output.

(Received June 1971)

One recommendation of the enquiry, headed by Lord Fulton, into the Civil Service was that selection of people for senior posts should be made from across the Civil Service, irrespective of individuals' origins or disciplines. The recommendation was accepted, creating a need for a central bank of records. Data existed already in the Civil Service Department on some of the senior people concerned but it was decided to create a new system to be known as the Central Management Staff Record (CMSR) based on a computer. Costs of a computer system were reckoned to be comparable with setting up manual systems capable of providing a similar service.

The task

The objective set in the autumn of 1968 was to create, in as short a time as possible, a file of about 6,500 senior civil servants together with an information retrieval system which would be easy to use and could provide information quickly to personnel managers in Whitehall to assist them in vacancy filling, manpower planning and statistical work. Whatever the system chosen, whether manual or computer, the method of producing results would have been similar, namely the matching of characteristics of posts to be filled against characteristics and job experience of the people on file. By clerical methods, matching can be laborious and error prone; by computer, using a retrieval program, the process of selection becomes more reliable. In this particular project, time and resources were in short supply and it was clear that an existing retrieval program would have to be found. After some investigation, SPECOL (Smith, 1968 and 1970a and b) was selected as being, for this application, the best and most advanced available software. Use of an existing program influenced to some extent the approach to the scheme and solutions to some of the problems encountered, for example, the type of record structure. Choice of SPECOL meant also that either an IBM 360 or ICL System 4 computer would have to be chosen and it also soon became apparent that the need for quick response meant remote access and in this case use of a teletype. It was decided that results from the teletype would have to be presented in such a way that they could be immediately understood by people who were not familiar with computers. This meant that all output would have to be de-coded and explanatory text words provided. In addition, because the main purpose was to provide information about people to assist in vacancy filling, ways of codifying job experience and qualifications would have to be found. Registering experience in turn meant that the people themselves would have to be involved in data capture, as only the individual himself can adequately describe his job experience.

System requirements

It was eventually decided that the principal requirements of the system should be:

1. Updating must be simple and error correction straightforward.
2. It must be capable of providing information quickly.
3. The retrieval language must be capable of use by non-programmers.
4. The method of communication with the computer during retrieval of data must be direct and uncomplicated.

These main requirements were met, first by adopting a record amendment system in which the layout of each amendment form corresponds with the equivalent record entry and the whole or part of an entry may be amended as required; secondly, by a teletype to link the Civil Service Department in Whitehall with the computer; thirdly, the availability of SPECOL and, finally, by expending not a little effort to devise a simple, conversational remote access method of communicating with SPECOL which would produce output in a comprehensible form.

Data capture

To set up the main file a questionnaire booklet was completed for each person. Information of a general nature was entered by personnel officers but details of jobs and qualifications were filled in by the individuals concerned. The information in the booklets was punched into cards, checked and transferred to the computer. After processing, check prints were sent both to each individual and his personnel officer, giving them the opportunity to correct or amend the new record. Whenever a record is amended a fresh check print is sent to the personnel officer and annually a complete print is sent to each of the civil servants on the file. These prints are a vital feature of CMSR. They help to keep the file accurate because the individual has the opportunity to point to errors or omissions and, at the same time, he is able to see the extent of the data recorded about him. Also, errors may be put right by entering the correct data on amendment forms together with the reference number, shown on the latest print, of each item to be corrected.

Record content

Each record on CMSR contains personal information of a kind found in similar systems within and outside Government. For example, name, date of birth, salary and qualifications are among the 35 or so items recorded. In addition, a system of job classification was devised based on five elements:

1. Occupation code (or field of work), e.g. financial accounting.
2. Function type (or role in the organisation), e.g. project manager.
3. Organisation type, e.g. nationalised industry } for jobs outside Government.
4. Organisation activity, e.g. transport } for jobs outside Government.
5. Salary level.

Jobs which are, or have been, held by senior civil servants both within and outside Government are coded using the first four of the above elements. There is also a limited area for entering a plain language description of the post. The fifth element—Salary level—defines each individual according to salary band. People, irrespective of grade or job, within a specified salary range can be identified using the appropriate salary level code.

Record structure

Records were structured with SPECOL in mind. Each record consists of a variable number of entries. Each entry, up to 150 bytes in length, contains a number of related items; for example, an entry may consist of pieces of information about a specific job or position held. A record may contain from 3 to 43 entries depending on the length of service of the person concerned. Every record starts with a G entry which contains the record key. All data are held in character form.

Layout of a CMSR record:

G entry	General information—surname, date of birth etc.
H entry	
J entry 01	Current job details
J entry 02	Previous job details (up to 20 previous jobs)
Q entry 01	Qualifications data
Q entry 02	

In SPECOL terms the G and H entries are 'headers' and can be considered as the fixed length part of the record; the J and Q entries are 'trailers'. There may also be S and T trailers in a record. References to look-up tables appear on each record next to the data.

System outline

CMSR has time on a System 4/70 computer at the Road Research Laboratory (Department of the Environment) used mainly for scientific work. It has 256,000 bytes of core, six magnetic tape units, six disc drives, two printers, two card readers and one communication control unit. The remote terminal is linked to the computer by public telephone line using the Post Office's Datel 200 data transmission facilities working in half duplex mode. CMSR uses about $1\frac{1}{4}$ hours per week updating and about 5 hours per week interrogating the file.

The organisation of programs falls naturally into two parts; a main file suite and an interrogation suite. In addition to these, and equally important to both, is a tables suite. The main file suite vets incoming data, creates and updates records, and produces check prints. The interrogation suite controls the tele-processing logic and arranges for the SPECOL compiler to be called when a question is assembled ready to run. The tables suite creates and amends look-up tables which are used both to de-code output and also for vetting new data. The latter use enables extensions of permissible code ranges without program changes.

The relationships between the three suites is represented in simplified form at **Fig. 1.**

With the exception of the SPECOL compiler, which existed already as a package, all programs were written by the Applied Programming Department of International Computers Ltd (ICL) under a contract from the Civil Service Department.

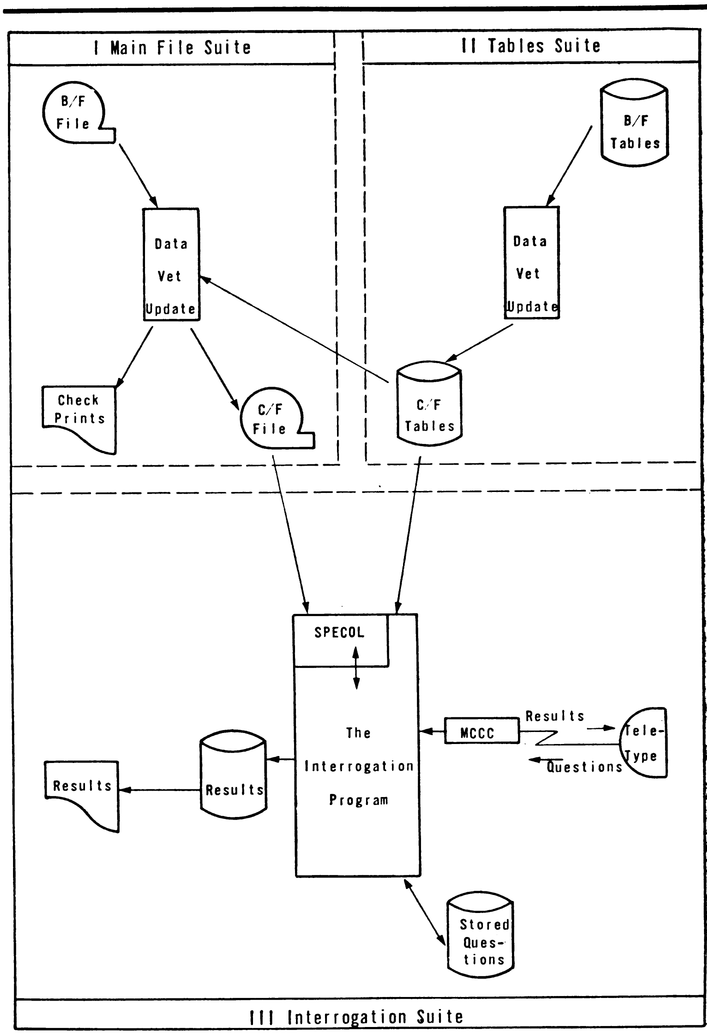

Fig. 1

'In-house' programmers were not therefore needed during the early days but they were required later to receive the completed programs and to maintain them. Programs were written mainly in COBOL but assembly language was used in some areas, in particular the interrogation program.

The interrogation program

This program is divided into six segments; one of these, the interface segment, calls in the SPECOL compiler as required. **Fig. 2** is a simplified diagram of the interrogation program showing its relationship with SPECOL and the nature of the part played by each of the six segments.

A more detailed description of the main functions of each of the segments is as follows:

1. Controlling teleprinter conversational logic.
   Accepting SPECOL questions from the teletype.
   Retrieving SPECOL questions from store.
   Accepting amendments to SPECOL questions from the teletype.
   Reporting error conditions to the teletype.
   Controlling entries and exits to other routines.
2. Converting the variable format lines into 80 byte card images suitable for SPECOL.
   Removing and processing special characters from the SPECOL question.
   Vetting input for errors.
   Obtaining a specific line from a question for amendment.
   Processing amendments.
   Vetting and preparing a question for running.
   Reporting errors.

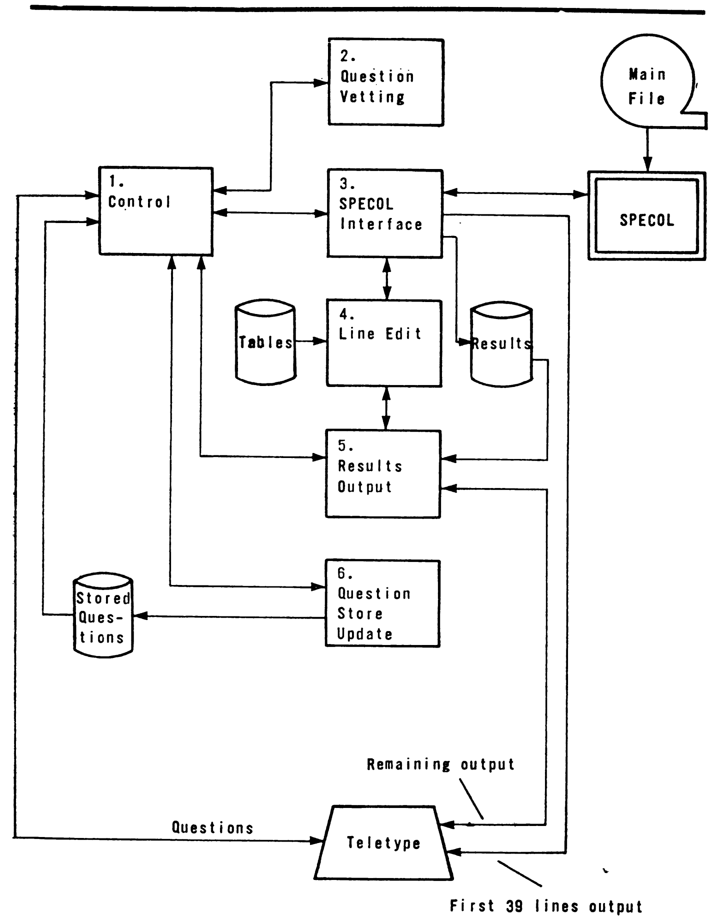

Fig. 2. The interrogation program and SPECOL

3. Providing an interface between the SPECOL compiler and the card image input from the terminal.
   Providing an interface between the compiled SPECOL program and the line image output.
   Arranging output of the first 39 lines of results.
   Keeping counts to enable further teletype output time to be calculated with reasonable accuracy.
4. Searching output lines for coded items.
   Decoding of coded items by referencing look-up tables.
   Informing the control of any line overflow.
5. Passing decoded lines of output to the teletype.
   Arranging the various options regarding repeating lines of output.
   Determining when the end of the results file has been reached.
6. Arranging the storage of questions on disc if specified by the teletype.
   Amending index of stored questions.

Terminal operations

From the terminal operator's view-point an important feature of the interrogation program is that at each stage in the dialogue he is presented with the possible commands open to him. For example, at the beginning of a session, after the unique identifier has been typed in, the program responds:

SEND MODE GET OR FINISH

The operator is obliged to select one of the three commands MODE, GET or FINISH in order to continue. This disciplined dialogue has two advantages. It has simplified the logic of the program which is therefore less likely to contain errors and also the operator's response time is short because his choice of commands is limited to those appropriate to each phase in the dialogue. The following are examples of output messages and their responses:

Program message	*Permissible response commands*
SEND IDENTITY	Requests the operator to send his personal identity number as a security check.
SEND MODE GET OR FINISH	MODE means that a SPECOL question (which must start with the word 'MODE'), is to be read directly from the teleprinter either by hand typing or using the automatic send facility. In most cases the question is prepunched on paper tape and then fed through the teleprinter. GET nn means that SPECOL question nn is to be retrieved from a permanent question file. FINISH means that the session is to end.
SEND GIVE GO AMEND OR ABANDON	GIVE requests the program to provide a specified line of the retrieved SPECOL question. GO informs the program that the question is ready to be compiled and run. AMEND is followed by an amendment to one or more lines of the question. ABANDON, this question is no longer required for running.
SEND REPEAT TYPE PRINT OR CANCEL	REPEAT causes a repeat of the results just output. TYPE continue with the results on teleprinter. Options are given every 39 lines to continue or halt output of results. Results are automatically held on disc for off-line printing if the TYPE option is taken. PRINT stops production of any further results on teleprinter but retains the results on disc for off-line printing. CANCEL deletes the results from disc and continues with session. This would happen if the complete results had been received on the teletype and no line printer output was required.
***	This message appears after 39 lines have been output in answer to the REPEAT or TYPE options if they have been given in reply to the previous message. Hieroglyphics are used as a reply to this message, so as not to interfere with the typed results.))) cease producing results (((continue with the next 39 lines &&& continue to the end of the results—no further options /// repeat the last 39 lines
SEND SAVE OR LOSE QUESTION	SAVE, the question is to be held in the specified slot on the permanent question file. LOSE, do not save question.

```
TIME 09:58:34  DATE 28:01:71
SEND IDENTITY
[1]
SEND MODE GET OR FINISH
[MODE 3  ACCOUNTANCY QUALS  <----------------------- Question title
TYPA GN17 (1010 TO 1013) AND GN10 ( > 18)    }          } Search parameters
TYPB QN02 (2) AND QA02 ($P430)               }
PNTA GA04 GA05 GX07 GN07 // GX01 GA01        } SPECOL   }
PNTB 0/ JX14 JN14 // JX17 JN17 // JX02 JA02  } question } Print parameters
AND QA03 // QX07 QN07 // QX11 QA11           }
END]
SEND GIVE GO AMEND OR ABANDON  <--------------------
[GO]  <---------------------------------------------
A.N.  OTHER               BORN:  010121  DEPTMT: AGRIC FISH AND FOOD }
J    JOB STRT:1268  JOB END:        GRADE: PRINCIPAL EXEC OFFICER    } Result
J    JOB STRT:0959  JOB END:1268    GRADE: SEN CHIEF EXEC OFFICER    }   of
J    JOB STRT:0855  JOB END:0959    GRADE: CHIEF EXECUTIVE OFFICER   } search
Q    MEMBER PROF INST        QUAL YR: 1938 PROF.QUAL:ACWA            }

      NO OF LINES SEARCHED:     44462

      OUTPUT TOTALS

            1 RECORDS
            5 DATA LINES
            6 PRINT LINES

      ***END OF RUN***
CMSR RUN ACCOUNTANCY QUALS
001  MINS TYPE TIME
SEND REPEAT TYPE PRINT OR CANCEL
[CANCEL]
SEND SAVE OR LOSE QUESTION
[SAVE 32]  <--------------------------------- The above question is stored on
                                              disc with serial number 32
SEND MODE GET OR FINISH
[GET 32]  <---------------------------------- Retrieve question 32
32  ACCOUNTANCY QUALS
SEND GIVE GO AMEND OR ABANDON
[AMEND                  }
02. TYPA GN17 (0113)]   } <------------------ Amend line 2 of question 32
SEND GIVE GO AMEND OR ABANDON
[GO]

      NO OF LINES SEARCHED:     44462 }  Result of second search. No record
                                      }  meets changed criteria.
      OUTPUT TOTALS

            RECORDS       }
            DATA LINES    }  Therefore no output totals
            PRINT LINES   }

      ***END OF RUN***
CMSR RUN ACCOUNTANCY QUALS
001  MINS TYPE TIME
SEND REPEAT TYPE PRINT OR CANCEL
[CANCEL]
SEND SAVE OR LOSE QUESTION
[LOSE]  <------------------------------------ Amended version of 32 not
                                              retained on disc
SEND MODE GET OR FINISH
[MODE 3  QUALIFICATION TYPE                   Next question
TYPB QN02 (1 OR 2)                         }  Question – to count numbers of
OVCNT QA03 ($N01) ($N02) ($N10) ($N20)     }  certain types of qualification.
END]
SEND GIVE GO AMEND OR ABANDON
[GO]

      NO OF LINES SEARCHED:     44462

      COUNT TOTALS

          5460    TYPA LINES
          9946    TYPB LINES


            559  QA03      DOCTORATE                }
            560   "        POST GRADUATE DEGREE     }  Results of count.
            393   "        OXCAM MA - 1ST           }
           1159   "        FIRST DEGREE - 1ST CLASS }

      ***END OF RUN***
CMSR QUALIFICATION TYPE
001  MINS TYPE TIME
SEND REPEAT TYPE PRINT OR CANCEL
[CANCEL]
SEND SAVE OR LOSE QUESTION
[LOSE]
SEND MODE GET OR FINISH
[FINISH]  <---------------------------------- End of session
FINISH TIME 10:19:26
```

Fig. 3. Example of teletype output

Fig. 3 is an example of teletype output illustrating some of the messages and resulting responses. Operator commands (responses) are enclosed in square brackets [ ].

Output

Printing from the teletype is relatively slow at 10 characters per second so that allowance has been made within the interrogation program for different amounts of output. The first 39 lines of results are invariably output on the teletype. During this time (39 lines takes about 4 minutes) the file continues to be read and results output to disc. The program keeps a count of the number of lines output and presents the terminal operator with two options. If total output will be less than 450 lines a message indicates how long it will take to print (the first 39 lines have already been printed) and gives the choice of continuing to output on the teletype ('TYPE') or on the line printer ('PRINT'). Alternatively if total output will be greater than 450 but less than 10,000 lines the program gives the operator the choice of outputting to line printer, or of cancelling the question ('CANCEL'). If the calculated total output will exceed 10,000 lines the program automatically abandons the question on the basis that it has not been formed correctly. Experience may show that the limits of 450 and 10,000 lines are either too low or too high and they may have to be adjusted.

Question storage and retrieval

One way of making efficient use of a terminal for information retrieval is by ensuring that questions are not only phrased correctly but also that the logic and search parameters of the question are correct. Lack of care in preparing questions wastes computer time. It is equally important that teletype output which is to be sent direct to users is presented in the best possible way. The CMSR system goes some way in solving these problems in that up to 70 questions which have previously been run may be stored and retrieved by the command GET nn, where nn is the serial number of the required question. The question may then be run immediately with the command GO. Alternatively, and more commonly, a stored question may be amended by inserting, deleting or amending lines of the original question. The question, as amended, may then be run leaving the original untouched in store for future retrieval. If desired the amended version may be stored by the command SAVE after it has been run, either in place of or in addition to the original question.

SPECOL

As mentioned earlier it was decided to build the information retrieval system around an existing, proved, enquiry program and we were fortunate that SPECOL had been operating successfully for some time on IBM 360 and could be adapted to System 4.

In SPECOL a series of comparisons are made between values specified in the search parameters of a question, and the values of data on records being interrogated. If a comparison is successful it means that the record satisfies the conditions laid down in the question. Even the most complex conditions can be expressed in terms of the three principal operators used in logic—AND, OR and NOT.

In questions, these operators are used to link the 'names' which are allocated to the fields of a record. These 'names' are listed in the SPECOL program with their locations in terms of byte positions.

Each SPECOL question is written in two parts; the first contains search parameters to identify records which satisfy the characteristics specified in the question and the second part stipulates what is to be printed out from each record. Information can be output in any form or position required. The SPECOL compiler is called in by the interrogation program. The question is compiled into the SPECOL program and, if valid, control is handed to this program which proceeds to search the file.

In the CMSR version of SPECOL each coded field can be referred to by using up to three 'names'. For example GN01, GA01 and GX01 each refers, for a different purpose, to the field 'Department'. G*N*01 (123) could appear as a search parameter of a SPECOL question and in effect means 'look for any record containing a Department with value 123'. G*A*01, on the other hand, would be used in the print part of a question to output, in plain English, the Department contained in a 'hit' record. And finally, if G*X*01 were used in the print part of a question it would cause the actual word 'DEPARTMENT': to appear as an explanatory text.

All coded fields on the CMSR are treated in this way. The terminal operator decides whether or not he wishes to use all

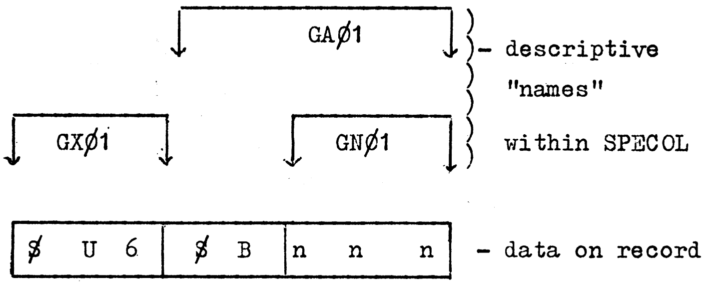

Fig. 4

three facilities in every question.

Fig. 4 shows how the item 'Department', a three digit code in character form, is held on a CMSR record. Following Fig. 4 is a simple question to illustrate the use of the three names.

Simple question

Identify all people in either Fisheries Department or Factories Department (codes 123, 456) and print surname, date of birth and Department.

Question written in SPECOL

MODE 1
TYPA GNØ1 (123 OR 456) } search parameters
PNTA GAØ5 GNØ7
AND GXØ1 GAØ1 } data to be output from 'hit' records
END

(GAØ5 is surname, GNØ7 is date of birth)

Specimen result

JACKSON	12 02 20
DEPARTMENT:	FISHERIES
SMITH	19 12 25
DEPARTMENT:	FACTORIES

In these results the surname and date of birth are data as held on the record. 'DEPARTMENT': is a text word (from GX01); 'FISHERIES' and 'FACTORIES' are decoded versions of codes 123 and 456 (GAØ1)

In the above example the text 'DEPARTMENT:' is probably unnecessary as it is clear that 'FISHERIES' and 'FACTORIES' are Department names. In the case of other items, for example dates, text words are extremely useful. The other words in the question, namely 'MODE 1', 'TYPA', 'PNTA' are SPECOL commands:

'MODE 1' indicates to the SPECOL compiler that output is required from 'header' fields only. Modes 2, 3 and 4 signify that output is required from combinations of trailer and header fields.

'TYPA' indicates to SPECOL compiler that the data on which search is to be made are in header part of record. (TYPB used for trailer fields.)

'PNTA AND' specifies that fields to be printed out from hit records are in header part of record.

Batch SPECOL

While the ability to interrogate file and obtain immediate answers over a terminal link is the main feature of the CMSR system there is, in addition, a standard batch method of interrogation again using SPECOL. This method is used in three situations; first, when the output is expected to be large and the teletype would clearly be too slow, secondly when the information is not required immediately, and thirdly, when results must be sorted.

Operations

When the system became operational in June 1970 the operating system would not permit simultaneous working of the Communications Control Program with that part of the operating system concerned with storing output on disc for subsequent printing. In practice this restriction has meant that CMSR can run only under an older version of the operating system and therefore only a limited number of other jobs can run simultaneously. For this reason CMSR terminal sessions take place at a fixed time daily to minimise the disruption caused by switching operating systems. This problem will be overcome with the next version of the operating system and the pattern may well be two or more shorter sessions during the day, as questions arise.

The file occupies about half a tape and interrogation takes place within tape passing speed. At the start of a session the current main file is mounted and control is passed to the teletype operator. Each session is usually about 1 hour in length during which time the file is read and re-wound several times. Rewind time cannot at present be used for submitting the next question and consequently each question takes, on average, 5 minutes to answer. Actual times depend on whether all or part of the file was searched.

Experience of CMSR

The system has been operational for about 8 months. As expected, the updating, being dependent on people remembering to notify the centre when changes occur, is the least reliable part of the system, justifying the extensive use of check prints to encourage individuals and personnel officers to keep the records up to date. Nevertheless a reasonable standard of accuracy was attained when the file was created and we expect this to be maintained and improved upon.

Practical experience of the information retrieval side of the system is encouraging. Potential users are being educated on ways in which they can use CMSR and people are beginning to turn to it for information previously obtained from diverse and less accessible sources. During the first 6 months a considerable number of requests have been met for information and prints of individual records. These requests have come from personnel managers with a wide range of responsibilities—pay, training, statistics, manpower planning, etc.—in the Civil Service Department indicating that already the need for information held on CMSR is broadly based. It is worth stressing that although the computer is employed to produce lists of people who possess the required characteristics for vacancies, these computer selections are not final. This is so for two reasons; firstly because performance and ability are not recorded on the file and secondly because a system of this kind must permit human intervention before final judgements are made.

Programs, hardware and telecommunication links have worked well although the short time the latter are used is probably not a fair test. Operator expertise both at computer and terminal have reached an acceptable level.

Future plans

CMSR became operational in June 1970 and is working successfully; in the immediate future the file will be extended to include a further 2,000 people. Consideration is being given to adapting the multi-question version of SPECOL to the system. In the longer-term the main file may be extended beyond 8,500 records.

In parallel with CMSR, work is going ahead on a much larger system called PRISM (Personnel Record Information System for Management) which, when its first phase is completed in 1974-5, will provide information about the whole of the non-industrial Civil Service. Both pay and personal data will be sent to PRISM from computers in Departments throughout the United Kingdom, initially by transfer of magnetic tapes but, it is hoped, eventually by data transmission. By linking PRISM with Departmental computer pay record systems it is expected to maintain a higher level of accuracy than can be achieved by other methods. CMSR will be merged with PRISM: the combination of the two records will, first, provide personnel managers with detailed information about senior people and, secondly, provide information for man-power planning in respect of the Civil Service as a whole.

References

SMITH, B. T. (1968). SPECOL—A computer enquiry language for the non-programmer, *The Computer Journal*, Vol. 11, p. 121.

SMITH, B. T. (1970a). Developments in SPECOL, *The Computer Journal*, Vol. 13, p. 10.

SMITH, B. T. (1970b). SPECOL—A Computer Enquiry Language for the General User. Civil Service Department. Crown Copyright 35p net.

THE CIVIL SERVICE. Report of the Committee (Cmnd. 3638), Vol. 1, Chap. 6.

Book reviews

Integer and Nonlinear Programming, by J. Abadie (editor), 1970; 544 pages. (*North Holland Publishing Co.*, £10·50 or $25.20)

Most workers in the field of mathematical programming will already be familiar with the first volume 'Nonlinear Programming' edited by Abadie in 1967. Now in 1970 comes his second volume; twice the size and unfortunately at twice the price, but just as indispensable as the first. The 26 papers and three appendices cover nearly every aspect of modern mathematical programming, from the latest and most general theoretical results to the most practical methods and applications. Roughly speaking the papers fall into five groups—general nonlinear programming theory and algorithms, quadratic programming and least squares, stochastic programming, integer programming and graph-theoretic results.

The first three groups of papers reflect the considerable progress which has been made in the last few years in nonlinear optimisation, both constrained and unconstrained. There is a welcome highlighting of practical algorithms and computational results although these are still extremely thin in comparison to linear programming. It is invidious to single out individual contributions but particular attention should be paid to Wolfe's paper on 'Convergence Theory in Non-Linear Programming', an important topic which has only recently come into prominence in comparison with the long emphasis on existence and duality theorems.

Another too long neglected topic in mathematical programming has been the numerical stability of algorithms. This state of affairs is now beginning to be redressed, and Golub and Saunders's contribution to this volume analyses a number of methods for least squares and quadratic programming.

Of the eight papers on integer programming three are concerned with the recent theoretical work of Gomory on the asymptotic integer problem and the faces of integer polyhedra (including a survey by Gomory himself). Of more immediate practical importance there are three papers on the implementation of branch and bound methods in production codes with very encouraging computational results. Another welcome contribution is Balas's survey of his results on mixed integer duality theory and their practical application to mixed integer algorithms (these ideas have also been implemented commercially).

The final section is perhaps highlighted by Dantzig's diverting paper on complementary spanning trees—an interesting example of Lemke's principle. The three appendices give valuable background material for a number of the papers in this collection.

In summary this is a book with something for everyone and should be on the shelves of everybody seriously interested in mathematical programming.

J. A. TOMLIN (London)

Algorithms and Recursive Functions, by A. I. Mal'cev, 1970; 372 pages. (*Wolters-Noordhoff Pub.* $15.50)

This is a very readable translation of a standard Russian text published in 1965. The first chapter defines some fundamental concepts such as alphabets, words, functions and the basic operations of composition, substitution, primitive recursion and minimalization. The next three chapters cover the basic ideas and theorems concerning primitive recursive functions, partial recursive functions, recursively enumerable sets, universal functions and various enumerations of functions and sets up to the level of defining productive and creative sets. Further chapters cover Turing machines and their relation to recursive function theory and variants such as normal algorithms, operator algorithms, multitape machines and tag systems. Applications of the theory are made to the word problem for semigroups, the decision problems for first order logic and for arithmetic, the non recursive-enumerability of the theorems of second order logic and Diophantine equations. In this last application Hilbert's tenth problem—still an open question when the book was written—has now been proved to be unsolvable, the final link in the argument being provided by the Russian, J. B. Matijasevic.

The book therefore treats very standard material such as would be covered in a first course on the subject and could well serve as a reference book for workers in the theoretical aspects of computation or as a text book for a course. Finite state machines are not discussed, nor of course are the latest ideas in the mathematical theory of computation. This reviewer liked the style of the book, the theorems are clearly stated and the proofs rigorous without being over mathematical. At the end of all chapters are examples and exercises which both illustrate the topics treated and also suggest recent research lines.

These complimentary remarks do not apply to the first chapter which attempts to define basic concepts such as words and functions. Several of these attempts seem rather unsatisfactory, for example the definition of a word as a sequence of cells occupied by letters, the definition of a class of algebras as an arbitrary system of algebras of the same type (what is a system?) and the difference (if any is intended) between a function and an operation is unclear.

The book should not be judged by its first chapter. Scattered through the book are a few misprints which affect the sense (e.g. on pages 31, 129, 235, 288, and 305) but these should easily be discovered. The book will be recommended reading on a course to be given by the reviewer.

D. C. COOPER (Swansea)

Information Theory for Systems Engineers (Econometrics and Operations Research XVII), by L. P. Hyvärinen, 1970; 197 pages. (*Berlin: Springer-Verlag*, $12.10)

The first (paper back) version of this book, which is based on lectures given at the IBM European Systems Research Institute, was reviewed in the *Computer Journal* in May 1969 (vol. 12 p. 182). This new version has been improved by the addition of specific illustrative examples, e.g. of the semantic difficulties arising in the translation of natural languages. The change from 205 pages of quarto in double-spaced pica typewriting to 197 pages of 23 cm × 15 cm in letterpress allows for some *increase* in content, and in particular there is an additional chapter covering the applications of noiseless coding to data compression, information retrieval, taxonomy and cryptography. The section on error-correcting codes has been extended a little, but this is a large subject to try to include in a book on information theory.

The book has a list of 56 useful references and 30 problems with detailed solutions. It is a pity that it contains occasional blemishes in English spelling.

D. A. BELL (Hull)