Strategies for the online dissemination of large geographically disaggregated time-series

Peter Mark Henderson

Strategies for the Online Dissemination of Large Geographically Disaggregated Time-series

by

Peter Mark Henderson

A thesis presented for the degree of Master of Science

University of Durham

Department of Geography

2000

DECLARATION

The work contained in this study has not been submitted elsewhere for any other degree of qualification and that unless otherwise referenced is the author's own work.

P M Henderson

STATEMENT OF COPYRIGHT

Acknowledgements

I would like to thank my supervisor, Prof. M.J.Blakemore, not only for his support throughout the preparation of this thesis, but also for giving me the opportunity of working at Nomis as part of the development team responsible for the redesign of the Nomis system.

I am also grateful to my second supervisor, Dr D.N.M.Donoghue, for his support, encouragement and comments which were important especially in the later stages of this thesis.

I would like to thank the Nomis team for their support – especially Dr Mark Ireland, for his guidance regarding some of the technical aspects of the Nomis system.

Special thanks to my wife Maureen and my son James Peter who have both given me the encouragement and enthusiasm to complete this work.

Strategies for the Online Dissemination of Large Geographically Disaggregated Time-series

P.M.Henderson

Abstract

This thesis investigates the various strategies used to produce a large on-line database which stores its data in the terms of dates and areas. The database studied was Nomis which was released in 1982 and provides remote access to official labour market data. The Nomis system underwent a complete redesign in 1997. A detailed account of the 'Old Nomis' system is provided in order to set the context for a study of its limitations. This study discusses the limitations due to the data design of the original model and code organisation as well as the limitations of the command interface and geographical management.

The next part of the study was to investigate different methods of implementing a redesign of the original system. This focused on the choice of technology available both in terms of software and hardware, ways of improving the user interface, designing a new data model and the development of a new geographical management system.

Once the new system was in place a menu-style interface was developed to provide a more user-friendly way to access the Nomis data. Various technologies were considered and the decision was taken to use the basic Web technology of HTML and forms due mainly to its accessibility to the majority of Nomis users and supported by all WWW browsers. Future Web technologies are also discussed.

The success of the redesign was evaluated in terms of examining the 'New Nomis' interface to see if limitations had been addressed. Also the 'Beta Testing' process was discussed with various users feedback indicating possible strengths and weaknesses of the 'New Nomis' system. Usage level performance graphs were also considered which compare usage levels of the 'Old Nomis' system against the 'New Nomis' system.

Table of Contents

List of Figures

Trademarks

Windows™ is the registered trademark of Microsoft Corporation

Nomis® is the registered trademark of the Office for National Statistics

r•cade® is the registered trademark of the University of Durham

Preface

Nomis is an on-line database developed at the University of Durham and since 1982[1] has been run under contract to Manpower Services Commission (MSC), then the Employment Department Group and more recently for the Office for National Statistics (ONS). The system provides remote access to a comprehensive range of geographically disaggregated official labour market related statistics for the UK on themes of:

- Demography
- Employment
- Unemployment
- Vacancies
- Earnings

This information is stored in a number of data series (or data sets). In late 2000 over 90 data sets were available. Each data set contains information on a discrete subject, built from a specific geographical base. The data from the base geography are aggregated to form larger areas allowing users to extract data at a wide range of geographic scales. The database resides on a dedicated host computer called Lmis (Labour Market Information System) which is located in the University of Durham Information Technology Service.

[1] 1982 was the year when Nomis became an 'operational' system for the Manpower Services Commission. Between 1979 and 1982 the system had developed out of a research project to analyse employment data. The MSC recognised that this was a more efficient way for them to access their data.

Aims and objectives

The aim of this study is to investigate the various strategies used to produce a large on-line database storing its data in terms of time and area. This aim will be achieved through the following objectives:

1. Provide a detailed account of the background to the development of the initial Nomis system ('Old Nomis').
2. Identify the strengths and weaknesses of the 'Old Nomis' system as it developed over 15 years.
3. Investigate different methods of implementing a redesign of the system.
4. Develop strategies to improve the overall geographical management of Nomis.
5. Expand the interfaces to the database, taking advantage of technologies such as the World Wide Web.
6. Evaluate the success of the redesign, as it was beta tested during 1998, and made operational in January 1999.

Structure

This study contains seven chapters. Chapters 1 and 2 detail the Nomis legacy and evaluate the 'Old Nomis' system. Chapters 3 to 5 describe the redesign of the Nomis system covering technologies, geographical management and interfaces. Chapter 6 evaluates the redesign. Chapter 7 summarises development and indicates possible future development.

Format

This thesis may differ in approach to some more academic studies. This is due to the nature of the author's work as part of the development team at Nomis, where a lot of the technical development work involved areas of research that are appropriate to this thesis. There are also some references to other on-line databases. However, the majority of current on-line databases for official statistics are of recent origins and have been specifically designed for the Web; for example ONS StatBase, Australian Bureau of Statistics, UNESCO, IMF, World Bank. Also, databases in existence in the

UK, which are of the maturity of Nomis, tend not to be comparable; for example the databases of datastream targeted for the financial sector, or SPSS-MR for the Labour Force Survey, which is specifically a one-data set processor.

Chapter 1 The Nomis Legacy

1.1. Introduction

This chapter covers the Nomis legacy from the initial release in 1982 to mid 1999. Issues studied are software, hardware and communications. The discussions will show the development of Nomis from a simple system used by some 20 civil service users, to an established database with a user base at over eight hundred sites. It will also give a technical description of the Nomis data structure, geographical structure and command interface.

The main purpose of introducing the Nomis system is to set the context for further study into the limitations of the system which lead to the reasons that a complete redesign was necessary.

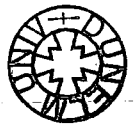
1.2. The development of Nomis

While it is common to remark on the increasing speed of computer developments in the 1990's, the 1980's also witnessed significant changes in hardware and software platforms. For example mainframes to minicomputers to microcomputers and local/wide area networks and these are reflected in the major changes made to the Nomis system since its first release. These are shown below in figure 1.1.

1.2.1. Hardware legacy

Date	System	Storage	Operating System
-1982	IBM 370/168 mainframe	3.6GB	MTS
1982-1986	IBM 4341 mainframe	1.2GB	MTS
1986-1988	Amdahl 470/V8	8.4GB	MTS
1988-1992	Amdahl 5860	14.6GB	MTS
1992-1995	SUN Sparcstation 2	9GB	UNIX
1995-1999	SUN Sparcstation 20	16GB	UNIX
1999-	SUN Ultra 2	30GB	UNIX

Figure 1.1. The Nomis hardware legacy



The Nomis system was originally housed on an IBM 370/168 mainframe computer located in Newcastle University running the Michigan Terminal System (MTS) operating system. This was due to the fact that Durham University initially had very limited computing resources so those available at Newcastle were shared under the Northumbria Universities Multiple Access Computer (NUMAC) agreement. Due to an increasing workload on the 370/168, Durham bought its first mainframe computer an IBM 4341 running MTS in the autumn of 1982. This machine had 4MB of memory and 1.2 GB of disk space. By 1986 the demand within the University for a more powerful machine resulted in the Computer Centre purchasing an Amdahl 470/V8 mainframe (an IBM clone 6 times faster than the 4341). Over the next five years the new peripherals were added to the system increasing the disk capacity and the processing power available.

In 1992 the University moved away from a large central server structure to an infrastructure supporting a far more distributed computing environment based on a range of new machines running the UNIX operating system. Bourne (1982) describes UNIX as a family of computer operating systems originally developed at Bell Laboratories. The operating system manages the resources of the computing environment by providing a hierarchical file system, process management and other housekeeping functions. A powerful command interpreter is available which allows individual users or groups to tailor the environment to suite their own style.

Prior to these changes, the Nomis system shared the central Computer Centre mainframe used by the entire University. Subsequently, Nomis would run on its own separate Sun UNIX server, the benefit being that Nomis users would not be effected by system loads imposed by other University users. However, the change to UNIX introduced new challenges to the Nomis system, and these are discussed in the next section.

1.2.2. Software legacy

Nomis was originally written mostly using FORTRAN 77, a small amount of assembler code and some advanced features specific to the MTS (Michigan Terminal

Service) operating system. The MTS facilities included the use of indexed variable length records, which were essential in the enabling of large data sets to be stored. While conventional direct access files allow indexing they did have two serious limitations. First, there were maximum lengths possible for any one record (32767 words). Second, all records had to be of equal size. Given the sparsity (here meaning the number of cells with zero, not with missing values) of many data series equal size records would result in much wasted file space.

Consequently, when the University of Durham IT Service moved to a UNIX system in 1992 the migration of the Nomis database to this new operating system was not simply a matter of moving the code and recompiling it. The entire data structure needed to be redesigned. Instead of the system automatically indexing the data files required for data retrieval each data file would now require a separate file to index the data. This is discussed in more detail later in this chapter.

1.2.3. Communications

Reliable communication between the user and the database has always been a key feature in the success of the Nomis system. Initially two means of access to Nomis were available. The first method early in the 1980's was through a bank of four 300 bpi dailup modems. Secondly for academic users, access was available through the Joint Academic NETwork (JANET). Both forms of access have improved due to more efficient technology and by mid-1998 the two means of access to the system were used:

1. British Telecom Dialplus service (formerly called GNS Dialplus) - This is designed to allow computers at remote sites to communicate with each other. Direct links to the network can be very expensive and Dialplus offers a cheap alternative by allowing the user of a telephone line to connect a local PC to the network via a local telephone exchange. Dialplus was used to provide non-academic users (i.e. outside JANET) with low cost reliable dial-up. The cost of Dialplus (excluding a local telephone call) was embedded in the Nomis charges. A single quarterly invoice for the one corporate Nomis account is then paid to British Telecom.

2. The Internet - Communication is available to the system using Telnet (or other communications software) on a PC to access the Nomis system directly through the Internet.

1.3. Data structure

When Nomis was originally designed a data structure was implemented to meet the intense demand for data once a month when the labour market data series are updated. This release at a set time (currently 0930 on a Wednesday) is called 'Press Release'. Characteristics of the structure include :

- Rapid access to data records using indexing - each record has a unique key determined by the geography code. Quick access was important to handle the high simultaneous demand for the new data released each month at 'Press Release'.
- Storage of large data series within limited storage using data compaction methodologies.

1.3.1. Nomis data files

The storage structure for Nomis data sets is :

- The data for each data set are stored in one or several subdirectories.
- Within these subdirectories pairs of files (one index file, one data file) are used to store each particular time period.
- There are several disk chains, each of which has a particular data domain associated with it (shown in figure 1.2. as D1 to D6). Each data domain is split into its own thematic group so that the data is stored in a logical structure.

The following diagram shows how the data files are structured for the data set USW91 which relates to the monthly 'Claimant Count' unemployment series.

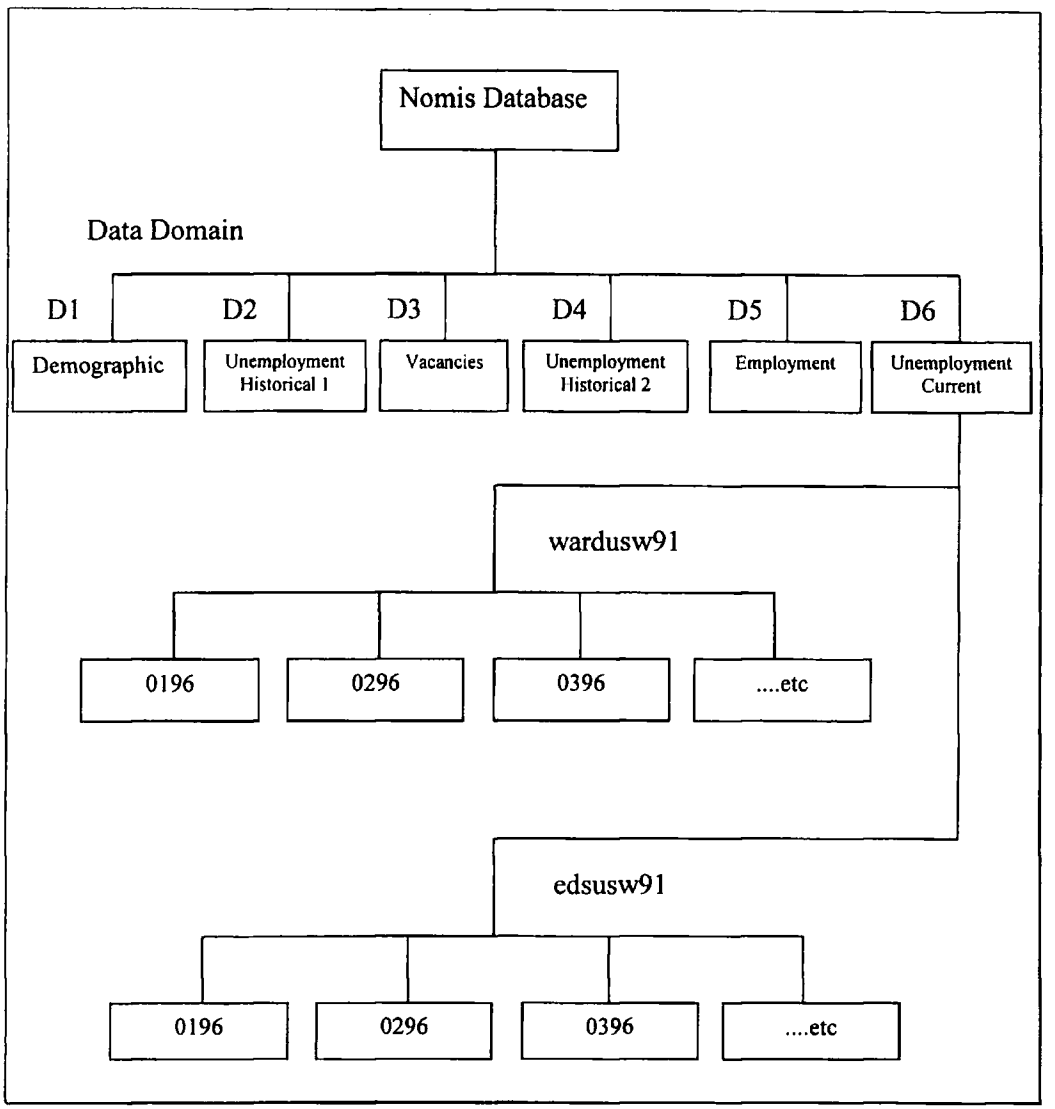


Figure 1.2. 'Old Nomis' data file structure

The data set USW91 is part of the domain "Unemployment Current". It is split into two subdirectories wardusw91 and edsusw91, which hold 1991 ward data and higher aggregate areas[1] respectively. These subdirectories hold files for each time period (0196, 0296, etc). Due to the geographical nature of the data the preferred method of storage was originally a single line of data per geography (unique area) using direct access. This allowed individual geographies to be equated to line addresses, so allowing fast access to data records.

[1] 'higher aggregate areas' are the geographical structures which are built up from the smallest building blocks. For example counties and regions are built from electoral wards.

1.3.2. Using indexed variable length records

Whilst Nomis ran under the operating system MTS, the system accessed data using indexed variable length records - a built-in utility of the operating system. However, the University of Durham IT Service's decision to move to UNIX meant that this facility was no longer available. No automatic solution was available, the only option was a new approach to the storage structure. This involved using pairs of files, one as an index and the other to access the actual data. These pairs of files are called .dir and .pag files.

- The .dir file is a file of line addresses and byte offsets. The offset is the physical location where data for this area starts
- The .pag file contains variable length compacted data records.

The two files interact in the following way:

The .dir file has two numbers per line (i.e. per area)

line number **offset**

The line number refers to the index number for a particular geography, and the offset value indicates the physical location in the .pag file where the data for this area begins.

The .pag file has the structure

line number **length** **data**

Where the line number matches the value held in the .dir file (used as a cross-check mechanism) and the length is the length of the data element in a compacted form (described later). The first byte of the data indicates the compaction code. In effect the .pag file is best visualised as a very long 'sausage' which is chopped up into slices representing geographies.

Part of the file '0198.dir' (directory file for January 1998 data) for the data set usw91 (unemployment stocks based on 1991 ward geographies) is shown in figure 1.3.

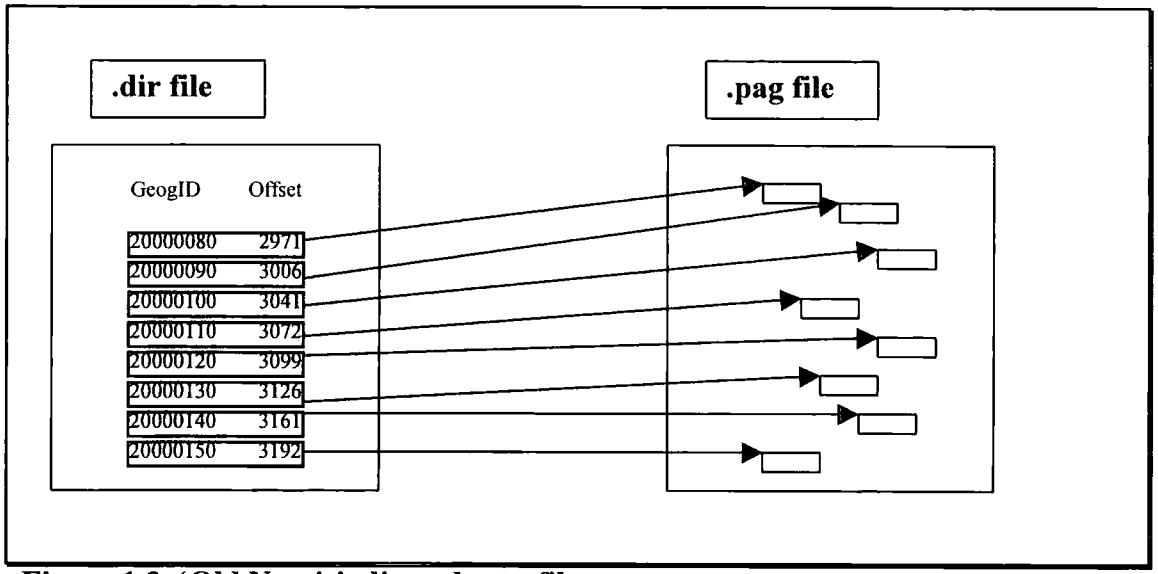


Figure 1.3. 'Old Nomis' .dir and .pag files

Using the above example the first line of data refers to area 20000080 and the data start at 2971 offset bytes into the file. The first value read in is the cross-check line number followed by the length of the data string. For this example it would be:

20000080 27

The 27 indicates that there are 27 bytes of which the first byte is always the compaction code.

4 bytes	4 bytes	27 bytes
20000080	27	1 byte compaction code, 26 bytes of data

1.3.3. Data compaction

Nomis compresses data records using a two-stage process.

First, there is run length encoding. It is assumed that all data values are non-negative integers. In the first twelve years of activity this was not a problem while there were no negative values - negative unemployment is highly unlikely. It is also assumed that

the number of zero data values will increase as both geographical and thematic resolution increase; this is also the case with a lot of the Nomis data sets. So, for the given line below

454 1365 1876 0 0 0 324 0 0 0 432 243 0 0 0

The compression uses a negative number to represent the number of zeros in any one sequence. For the above example the fifteen values are replaced with the following

454 1365 1876 –3 324 –3 432 243 –3

This reduces the number of storage locations required to nine giving a compression of 60%.

Secondly, to take the data compression a stage further, each line is scanned to determine the magnitude of the data values. Using the above example the maximum value is 1876. Left unchanged, this value and all of the other values would be stored in 32-bit variable elements, most of which would be wasted space. Nomis checks the maximum value of the line against a look up table which attempts to compress the 32-bit values into 12, 16, 24 or remain in 32-bit format[1]. The following relationships exist

Compaction Type	**Value**	**Minimum Storage Required**
1	0 – 2047	12-bit
2	2048 – 32,767	16-bit
3	32,768 – 8,388,607	24-bit
4	8,388,608 – 2,147,483,647	32-bit

All of the values in the earlier example lines could fit into 12-bit sequences. The overall effect on the storage of the example record is as follows:

[1] The maximum value of 2,147,483,647 well exceeds the population of the UK (circa 60,000,000) or the number in the workforce (circa 25,000,000). This was a robust assumption up to the stage where fractional numbers of employees were provided in the Census of Employment in the late 1980s

Description	Total	% Saving
Uncompacted	15x32 bits = 60 bytes	
After stage 1	9x32 bits = 36 bytes	40%
After stage 2	9x12 bits = 15 bytes*	75%

* One extra byte is added to the beginning of the record to identify the compaction type. If the overall length of the record is not divisible by 8 then extra zeros are added to fill up the remaining byte.

The impact of this compaction regime was dramatic. If run-length encoding had only been used the database would be compacted to about 85% of original size. The full regime resulted in compaction to 12% of original size.

1.4. Command structure

'Old Nomis' is a command-driven system. To extract data commands are typed in rather than selected from a menu. Generally six command components are required :

1. **What** data set is required?
2. **When** in time?
3. **Who** (disaggregation by sex, industrial classification etc) is required?
4. **Where** geographically is required?
5. **Which** categories are required?
6. **How** is the output to be produced?

Once the system has established the required commands the output can be delivered.

To extract the August 1996 unemployment data for the 12 UK regions the following commands are required

data=usw91	(what?)
date=896	(when?)
sex=all	(who?)
region91=1-12	(where?)
item=1	(which?)
print	(how?)

This command sequence produces the following output shown as figure 1.4.

```
USW91 Stocks Ward91 Base Aug 1996 Nomis (Crown Copyright) Oct 7,1998

Title : Unemployed claimants

Warning: Claimant count data are subject to discontinuities over
time, see Labour Market Trends, November 1995 for details.

        Persons       REGION91 Name
   1    292,291       South East
   2     60,907       East Anglia
   3    368,854       London
   4    147,798       South West
   5    194,729       West Midlands
   6    135,685       East Midlands
   7    195,732       Yorkshire and Humberside
   8    241,219       North West
   9    135,023       Northern
  10    105,282       Wales
  11    206,375       Scotland
  12     92,554       Northern Ireland
     2,176,449       Column Totals
```

Figure 1.4. Query output

Nomis makes extensive use of code numbers in several of its commands including geographies, items of information as well as occupational and industrial breakdowns (not used in this example). The large range of codes used in Nomis are retrievable via user manuals or on-line help.

If the next requirement from the system was to output the same information for the 66 counties in Great Britain, the user need only type the following extra lines :

county91=1-66
print

The Nomis system stores previously input commands. When further output is required the whole command sequence does not have to be re-input. Only the options that need to be changed must be typed followed by the output command (in this case changing to another set of geographies and ***print***).

The ***print*** command is the simplest output command and is available for all data sets. Other output commands include ***change*** and ***average*** which will output the change or average figures for a particular date range. Using the above example typing :

date=196-197

change

will display the change in unemployment between January 1996 and January 1997. Replacing ***change*** with ***average*** would produce the average figure for the same date range.

This command approach was initially very elegant. In FORTRAN 77 commands switch on/off 'flags' in a series of arrays. Limitations started to emerge as new data facilities were added over the years and the arrays were extended. Eventually unforeseen interactions would result, requiring occasional software patches. Furthermore, this placed increasing reliance on retaining programmers with a long developmental 'memory'.

1.4.1. Command syntax

The Nomis system has several general rules that can apply to most commands.

- To select a single instance of a command, simply specify the command followed by an "equals" followed by the command value that is required.

  e.g. ***date=198*** (select the single date January 1998)

- To select several sequential instances of a command use the "minus" character to specify that a range is required.

  e.g. ***item=1-4*** (select the items 1 to 4 inclusive)

- To select a list of command values which are non-sequential use the "comma" character to separate the command values.

  e.g. ***region91=1,3,9,11*** (South East, London, Northern and Scotland)

- Ranges and lists can be combined in one command sequence.

e.g. ***date=197-497,697,897*** (selects the dates January – April 1997 and June 1997 and August 1997)

1.4.2. How a Nomis query breaks down

Several indexed files are used in any typical Nomis query for example :

```
1. data=usw91
2. county91=44
3. date=0198
4. sex=male
5. item=1
6. print
```

Figure 1.5. A typical nomis query

- Using the above example the first command (data=usw91) selects the unemployment claimant count data set built up from 1991 ward geographical areas. The system determines two things from this selection

  1. The geography series – 'Old Nomis' used two geographic series "a" for 1981 ward based geographies and series "b" for 1991 ward based geographies.

  2. The directory in which the data is stored – in this case edsusw91.

- The second command (county91=44) selects the geography type and area - in this case County Durham (based on 1991 Census of Population wards). The geography selected determines the offset into the data file.

- The third command (date=0198) determines which file in the data directory to select – in this case "0198.dir". The number obtained by the geography selection will return the line address in the "0198.pag" file which contains the actual data record.

- The fourth and fifth commands (sex=male & item=1) determine which part of the data record the user is interested in – in this case the male wholly unemployed claimants.

- Print is one of the data output commands. It is the default setting for the data to be displayed to the screen.

1.5. Geographical bases

All data stored on Nomis are accessed using geographical areas, and each data set has associated with it one or more base geographic areas. A geographical base is the smallest area for which information is available. These areas are combined by the system to create aggregate areas but can also be combined by the user to create custom-defined areas. There are three types of base geographic area available:

1. **Wards** – these form the building blocks for most administrative areas. Wards are combined to form local government areas such as unitary authority areas or local authority districts. Local authority districts are combined to form counties. Other administrative areas formed from wards include parliamentary constituency areas and Training and Enterprise Council areas.

   To enable comparison over time a consistent geographical base must be used. These are known as 'frozen wards' and their boundaries are fixed at the time of a decennial Census of Population (currently 1981, 1991), to form a stable geographic area.

2. **Postcode sectors (ps)** – There are approximately 9200 postcode sectors in Great Britain. Postcode sectors consist of one or two letters (indicating the postcode area) followed by a one or two digit number (indicating the district) and lastly a single digit which identifies the sector (ie the full postcode minus the last two letters. For example

SR – Sunderland postcode area name

SR1 to SR9 – subdivided into nine postcode districts (psd)

SR1 1, SR1 2, SR1 3 – district SR1 is subdivided into three sectors

Postcode sectors are liable to frequent change. Nomis code numbers are changed whenever the Central Postcode Directory (CPD) is updated , usually every six months.

3. **Jobcentre areas** – These are available for use with the vacancy data sets. The Employment Service Jobcentre Network contains all offices that report job placings. This includes sub-offices which advertise and fill vacancies but to which vacancies are not notified directly.

1.5.1. The frozen ward hierarchy

As shown above, wards form the building blocks for most administrative areas. The following diagram shows the concept of wards forming aggregate areas. Wards build up to create districts, which build into counties, which build into regions, which finally build into countries (not shown).

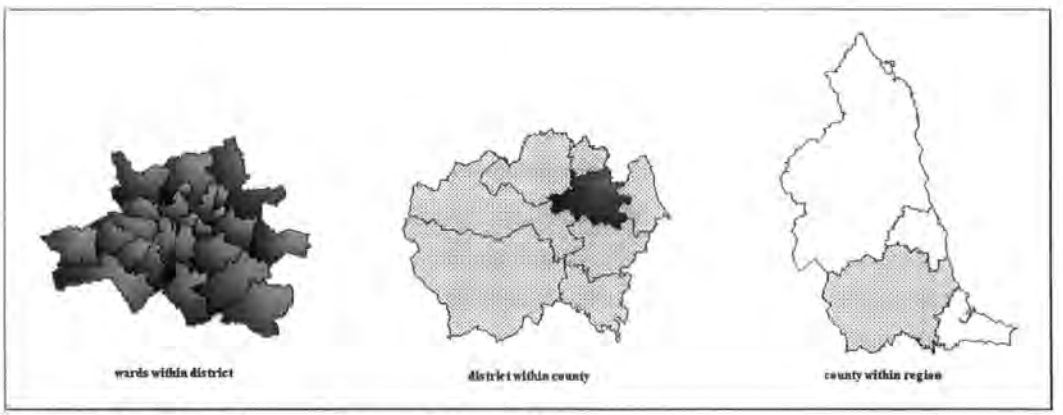


Figure 1.6. The "frozen" ward hierarchy

1.6. Conclusion

This chapter has introduced the 'Nomis' Legacy and showed the development and expansion of the system from the initial release. From 1982 until 1992 the 'Old Nomis' system changed and developed quite frequently. The expansion was due partly to the hardware changes at Durham and also the significant change in operating systems in 1992. The system was becoming more limited in what it could achieve in terms of the complexity of data sets it could store, the difficulties of handling frequently changing geographical areas and the inflexibility of the command interface. As the system developed, it was apparent that it was becoming more "out-of-date", particularly in the user interface, and a radical view of how the system would cope in the future needed to be undertaken. The limitations of the 'Old Nomis' system are the subject of chapter 2.

Chapter 2 Limitations of the 'Old Nomis' System

2.1. Introduction

Nomis was initially designed in the late 1970's as a local research database allowing on-line access to a single data set (the Annual Census of Employment 1971-1978). The technology initially available provided very limited system resources in terms of memory and disk storage. The primary objective in 1978 was to provide internal access to researchers in the University of Durham and Newcastle. The data 'owners', the Manpower Services Commission (MSC), were subsequently keen also to have access to their efficiently structured data, and this took place via standard telecommunication with acoustic modems. As they used the system they encouraged Durham to add other data series such as unemployment data. New data sets and more geographical building blocks were added to the original design. This expansion occurred at such a rate that by the time the system was well established, the limitations inherited from the original model were embedded deeply into the structure of the system. This chapter will discuss these limitations and identify the mechanisms developed to overcome them.

2.2. Data design

The data and file structures introduced in chapter one overcame the problems of storage and access, but at the same time introduced several new problems:

- Inefficient updating

If any new values needed to be added to an existing data set (for example a new or a revised geography), then the values are appended to the data files. These files are then sorted and part of the sorting process is the removal of the old data records. This is a result of the way that the 'Old Nomis' system structures the data.

- Platform dependency

The original system made use of "indexed variable length records" which was a system-specific feature of the Michigan Terminal Service (MTS) operating system

available at the time of design. All of the Nomis data storage was organised using this facility and when the University of Durham moved to a UNIX system in 1992 this facility was lost. Major software changes were made to emulate the indexing facility which were very efficient for reading data from the database but these were not nearly as efficient when writing data into the database.

- No provision for missing data

At present if data is requested for a data set which contains certain items that are missing the system returns a zero value and reports the requested information as missing data. A more sophisticated method of handling and reporting missing data values would allow us to discriminate between truly missing data and data which have not yet been loaded.

- Time series processing is slow

Nomis was optimised to allow fast access to many different geographic areas for a single time period. The system achieves this by storing the data for each time period in separate files. This organisation results in time-series queries being computationally inefficient, so making time-series queries comparatively expensive for on-line customers.

- Non-integer and negative data values

The original system assumed that all labour market data would be non-negative. It is not conceivable to have negative employees or negative unemployment. Additionally, the numerical values were integers (whole numbers) since fractional employees were not present. This assumption allowed the storage and compaction regimes to be based on run-length encoding and integers. During the late 1980's developments in the Census of Employment caused problems. There was a move to sampling, and to provide fractional employee numbers with grossing-up factors, which results in floating point values. The 'Old Nomis' system was modified and used implied decimal points to store floating point numbers. For example a figure of 1.17 is stored as 117 and when retrieved is divided by 100. If a large amount of these values are required for a query then each value once retrieved must be calculated which becomes increasingly inefficient. Numbers greater than MAXINT (2147,483,648) are

impossible, which caused an overflow when the Great Britain figure total was accumulated.

- The maximum record size is limited by a fixed buffer size

The maximum record length must be less than 32,767 bytes. This did not present a problem until the late 1990's. Even with the 1991 Census of Population all of the variables, some 20,000 of them, could easily be stored within the record limit. The addition of the New Earnings Survey in 1997/8 meant that the limit would be considerably exceeded. A worked example is provided in section 2.3.1.

2.3. Code organisation

FORTRAN 77 programming, with the original use of restricted memory mainframes, forced the extensive use of shared memory and common blocks. The common blocks in particular meant that unrelated procedures could end up being linked via shared memory.

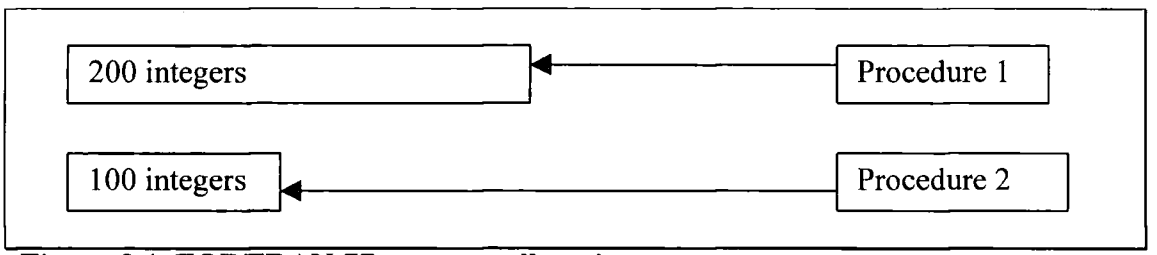


Figure 2.1. FORTRAN 77 memory allocation

In the above diagram two procedures use unrelated arrays of different sizes. Due to Nomis using shared memory, these two procedures can potentially interact unpredictably because the arrays may be stored in the same area of memory.

Furthermore, FORTRAN 77 has no dynamic memory allocation. This led to limits being set on the volume of data that could be extracted in a single query. The limit was forced by system design and not by user needs. For example, it was not possible to extract data for all 12,000 wards in a single query, and there were arbitrary limits set on the number of user-specified combinations of ages and durations or of standard industry or occupation codes.

Adding new data sets was requiring constant non-standard code modifications. Nomis FORTRAN 77 subroutines had specific functions that were often slightly different between data sets. As more data sets were added the routines contained an increasingly large set of switch statements of exceptions

i.e. if data set=1 do .....
else if data set=2 do .....

In spite of the fact that Nomis was originally set up as a demonstration system in 1978 with one data set, it was still in operation some 20 years later, which is an indication of the robustness of the original concept. The hardware restrictions available when Nomis was first developed meant that efficient memory usage was an essential part of the design. To achieve this with the software available at the time meant that the code was not very modular. Adding new data sets required changes to certain parts of the code, which could effect other parts of the system. A very good understanding of the system was needed to trace and eliminate such knock-on effects.

2.3.1. Examples of the Limitations of 'Old Nomis'

In 1998 one of the new data sets required for availability online by the Office for National Statistics (ONS) was the New Earnings Survey (NES)[1]. This data set consisted of information such as average wages, hours of work, overtime, etc. The NES data, like many other Nomis series, does not come readily packaged in the geographical building blocks such as wards and pre-set thematic groupings. The aggregate on-line series are built-up from anonymised microdata.

The size of the 'line' of data on Nomis is the result of the number of vertices in a data matrix. For example, standard unemployment claimant count monthly series would be gender (3) by items of detail (7) giving 21 elements. By age and duration this is gender (3) by age (14 bands) and duration of unemployment (17 bands) giving 714 elements. The NES has 3 vertices, which result in a potential data line exceeding the

[1] The fact that the release to users did not occur until 2000 was the result of protracted ONS managerial decisions, not for any technical reason.

32,767 byte limit of the existing data model. This data set would not fit into the existing model due to the following:

- The NES data is not a specific count, it is an average figure. This makes the values non-additive. 'Old Nomis' assumed that you could get data for ***n*** areas and add them together.
- The values are floating point. In the old system, they would have to be stored as implied floating point, which makes retrieving them very inefficient.
- Many values stored would be larger than MAXINT. Storing such values is impossible under the 'Old Nomis' system.
- Under guidelines from ONS certain values must be suppressed from users. The data suppression rules are based on standard errors, which are non-additive and complex.
- Each data record would be far greater than the 32,767 byte limit. For a single area and time period a NES record could be anything up to 223,200 64-bit values which is far in excess of the old limit.

4 Sexes*
372 Occupation
150 Items

* 9 different sex breakdowns are available from the NES, which are combinations of 4 base sex breakdowns (male full-time, male part-time, female full-time and female part-time)

Another new data set added to the system in 1998 by ONS was Unemployment Stocks by Occupation, Age and Duration. Assuming that each record for a particular data set is located using a date and geographic area, each record would have 3 sex types (male, female, total) by 2 items by 13 age breakdowns by 16 duration breakdowns by 112 occupations. This equates to 139,776 elements for each record. The 'Old Nomis' system could not implement such complex data without splitting it up into several sub-sets. That was neither practical for users, nor sensible technically.

2.4. Command interface

Four main problem areas were identified:

1. New commands have been added over time, so diluting the original consistency.
2. There was unnecessarily complicated syntax for some commands.
3. The command system was not context sensitive.
4. There was a need for a Graphical User Interface (GUI) given the advent of the World Wide Web and increasing user expectation of menu-driven graphical data selection.

2.4.1. Inconsistent commands

The additive development of Nomis during the 1980's meant that a large number of Nomis commands do not follow a logical structure. They can have similar but slightly different rules of syntax, which ends up being confusing particularly for new users.

For example in 1999 there are three demographic data sets available – migrations (MIGR), mid-year population estimates (PEST) and population projections, births and deaths (PPBD). All of these data sets have an age breakdown available but in each case a different command is required to select it.

Data set	**Command**
MIGR	migrantage
PEST	pestage
PPBD	progroup

The existence of three separate age selection commands is logical in the context of the statistics being processed, but is not logical for new users. Each data set has been introduced at different times, with different demands on selection being imposed by the statisticians responsible for the data. Nevertheless, the way in which new data sets are added to 'Old Nomis' also tends to force the adoption of new command mnemonics.

2.4.2. Unnecessarily complicated commands

Some of the commands on the Nomis system are unnecessarily complicated. When users are looking at data sets which have an "age and duration" breakdown if they require only a partial breakdown they use the subset command in the following way.

subset

age=1-5,dur=5-7,head="17-24 year old 8-26 weeks"

end

The system expects three separate commands in order to define one "age and duration" subset definition. The only information that the system should require is the middle line, i.e. the command that contains the actual user's requirements. Also, if a user defines a subset but forgets to specify the ***end*** command the system still expects ***age/duration*** definitions to follow and will not recognise standard commands until the ***end*** command has been read. This particular problem has been one of the single biggest causes for help-desk calls on 'Old Nomis'.

2.4.3. Command system insensitive to context

The Nomis command system has no idea of what position a user is in within a query. When a user constructs a query, the system is unable to inform the user of any information regarding the commands they have input or need to input to complete the query. The only method of obtaining any information as to whether a query is correct is first by typing an output command such as ***print*** in order to receive an error message, or second by using the recap command which lists the command options both selected and set by default. The latter is not really instructive to a novice user.

```
*data=usw91
*print
     ###  Error: 1058
     ###  Output command seen.  No Geographical areas defined!
Ready for input:
*region91=1
*print
     ###  Error: 1059
     ###  Output command seen. No Sex defined! SEX = required.
*sex=male
*print
     ###  Error: 1061
     ###  Output command seen. No Date defined! DATE= required.
Ready for input:
*date=598
*print

wait - processing your commands

USW91 Stocks Ward91 Base            May 1998  Nomis (Crown Copyright)
Jan 14, 1999

REGION91      South East

Warning: Claimant count data are subject to discontinuities over
time,
   see Labour Market Trends, November 1995 for details.

      Male     Female    Persons

   116,283     36,576    152,859  Unemployed claimants
         0          0          0  Students on vacation
        29          1         30  Temporarily Stopped
       759        614      1,373  Claimants under 18 years
         0          0          0  Unused
         0          0          0  Unused
         0          0          0  Non-Thursday T.S.
         0          0          0  Careers Office non-claimants
                8,836             Married Females
                24.16             Percentage of Female claimants
                          22,963  Under 4 weeks,under 60 years
                             372  Under 4 weeks,over 60 years
                         128,202  Over 4 weeks,under 60 years
                           1,318  Over 4 weeks,over 60 years
```

Figure 2.2. 'Old Nomis' error messages

Figure 2.2. illustrates how the 'Old Nomis' system responds each time a command is entered followed by the output command (print). The first command (data) is used to select the data set required (usw91 - Unemployment stocks 1991 ward base). When

the print command is entered the error message informs the user that they have failed to enter a suitable geographical area. The second command attempts to rectify this by selecting an appropriate area (region91=1 is the South East region) but this time after the print command the error message informs the user that they have not selected a sex. Once a suitable sex option has been selected (sex=male) the print command informs the user that a suitable date is required. Finally once a date is selected (date=598) the print command outputs the requested data.

The above problem suggests that a system sensitive to user needs should either prompt for each command as it is required, or display a table showing commands already entered and commands yet to be entered. The system should also allow rapid completion of a query using pre-set defaults.

2.4.4. Adding a simple Graphical User Interface (GUI)

A Graphical User Interface (GUI) is a "Point and Click" style interface (e.g. the menu bar on a modern word processor). Even in mid-1999, while the Nomis command-driven interface remains very popular with experienced users, it can be difficult to master for the inexperienced. In addition, the expansion of the user base to nearly 800 sites from all sectors, has meant the user base has moved well away from the historical core of labour market specialists who understood the detail and nuances of the data. Users now tend not to be data specialists so there is an increasing need to supply them proactively with metadata and guidance.

Adding a GUI to the system would be of great help to inexperienced users as all of the options would be available to them "on screen" and all that they would have to do is select their appropriate choices. In the past adding a GUI has not been possible due to the technology to the majority of our users. This is no longer the case, but it would still be difficult to implement a GUI on 'Old Nomis' due to the following:

- Nomis commands are idiosyncratic – this would mean having several similar menu options for idiosyncratic commands (such as the three different age commands in the example earlier). This would not simplify the interface for users.

- Nomis is not context sensitive – the Nomis system relies on the user typing in a command and then responding to that command (either successfully or reporting an error). It would be difficult to implement in the reverse order i.e. Nomis prompting the user with the available options (e.g. the dates available for a particular data set) and then users selecting one of these options.

2.5. Geographical limitations

Several geographical limitations exist within 'Old Nomis' these are discussed below.

2.5.1. Handling 'mobile' geographical definitions

During the early years of Nomis through to the advent of the Training and Enterprise Council geographical definitions were relatively stable. Wards built into local authorities, parliamentary constituencies, counties and regions etc. While parliamentary constituency definitions were revised occasionally, old ones were replaced by new ones, for example pca91 superseded pca81 (constituencies based upon 1991 and 1981 frozen wards respectively). Here the constituencies were 'frozen' so that time series analysis was possible on a consistent basis. The Nomis design used this feature to pre-build all higher-aggregation geographies. This was important for two reasons. First, it delivered data quickly to users without having to build the data at the time of retrieval. Second, it meant that higher aggregation data files were accessed at speed during the very high user load around the release time for unemployment series.

If an aggregation file could be 'read' into memory then many users would gain the impression that they simultaneously accessed the data at the release time. In the 1990's frequently changing definitions for Training and Enterprise Council areas, and for the unitary authority areas, have resulted in the need to rebuild data aggregation frequently. The major problem occurred with Training and Enterprise Council area boundaries which are 'owned' by the individual enterprise companies. They can decide to change them almost at will – for example one instance where two Chief Executives agreed to exchange some wards and their analysts then phoned the Nomis help-desk to see if we could reflect the changes immediately.

With the old data model every time an area changes all of the data for that area has to be reloaded into the database for all time periods which are available for that area. A solution was needed to build data for these areas at retrieval time. Then as long as the new geography definitions are updated then the data would not have to be reloaded. Faster and cheaper processors would also help to improve the speed of this type of data aggregation.

2.5.2. Difficulty in handling geographic discontinuities

Every ten years, when a Census of Population is carried out, the frozen ward boundaries are redefined. This presents a major discontinuity to all of the data sets that contain ward data. Nomis distinguishes between different frozen ward versions by adding a year suffix to the appropriate geography command - e.g. ward81 or ward91.

The 'Old Nomis' system puts this data into separate data series rather than using a single data set. The system should be able to handle such discontinuities by allowing consistent geographical types to be aliased (i.e. county81 and county91 would just be County) and displaying geographical discontinuity warnings.

It should be acknowledged that this proposal was a contentious one. Significant boundary changes can occur with wards that keep the same name, so data that are output may show changes that are the result not of the thematic issue to which they refer, but to the effect of boundary change. Better metadata (for warning messages) and supporting documentation would be needed to educate users about geographical complexities.

2.5.3. Multiple geographical bases for a data set

Several similar data sets have to be separate because of geography differences. Some data series have non-meshing geographical building blocks. For example, the claimant count unemployment was built for frozen wards, postcode sectors and unemployment

benefit office areas. The data were held in separate data domains. This separation was the result of the geographical building blocks being introduced at separate dates by the Department of Employment and the resulting fact that a new geographical base had to be regarded as the start of a new data series.

2.6. Increased output flexibility

Nomis was designed mainly to generate data output in a predefined table format. This was at a time which predated data transfer into other applications such as spreadsheets. Given the ancestry of the system in the late 1970s early 1980s this is not surprising. In the late 1970s and early 1980s using teletype terminals only had the option of a hard copy record of a session. As PCs with Windows™ became popular the production of comma-separated data files/output was provided. Files were saved locally by capturing to disk or via File Transfer Protocol (FTP). Now users expect data to be 'fired' directly into local applications packages such as Excel. While it would be technically feasible to program such facilities into 'Old Nomis' the result would have been a patchwork quilt of replicated code that would only add to the logistical burden of maintaining the system.

2.7. Conclusion

This chapter has established the limitations in both the data design and the code organisation. The logistics of adding new complicated data sets to the existing Nomis system increasingly became resource intensive. The other limitations that have been identified could be resolved within the existing system, but this would not detract from the fact that in order to integrate more sophisticated data sets a complete re-design of the major components of Nomis was required. A completely new file structure would be required to accommodate large complex data sets. The next chapter will discuss the challenges in a complete redesign of the Nomis system.

Chapter 3: The Nomis System Redesign

3.1. Introduction

Limitations with the existing Nomis system, discussed in the previous chapter, resulted in a decision during 1997 to undertake a complete redesign of the system. This chapter will describe the technology considered for the new system and will discuss the resulting enhancements.

3.2. Technologies

The system development tools available in the late 1990s were orders of magnitude more sophisticated compared with the late 1970s when Nomis was first devised.

The software strategies considered were as follows

1. A move to 'Open Systems' software such as a Relational Database Management System (RDBMS)
2. A "Home grown" system.

In 1994 there had been a review of the Nomis software strategy as part of the periodic re-tendering process. Civil Service software approaches mandated that we evaluate RDBMS systems against the in-house software solution. At that time the outcome was to remain with the in-house approach because RDBMS would only address some 10-15% of the software demand and would require a massive expansion of disk and processor resulting in a cost increase of 30%. We re-visited this in 1997.

3.2.1. Relational Database Management Systems

There are various Relational Database Management Systems (RDBMS) available at present, but two widely used in business and academia are Oracle and Ingres. These were the platforms reviewed in 1997-1998. The main advantages of using an RDBMS are :

- Open systems approach, so (arguably) leading to easy transfer of software to another contractor.
- Reducing the amount of development time required. RDBMS have been developed specifically to allow fast and effective system design.
- Ease of data insertion and modification using Structured Query Language (SQL). Date (1990) describes SQL as a language used to formulate operations that define and manipulate data in relational form. SQL is a recognised standard and is used as the basis for many large database systems.
- Tools are available to develop full-screen user interface. Simple menu-driven interface screens can be designed relatively quickly using in-built tools.

Despite these advantages there are also several drawbacks associated with using a RDBMS system. These are in the areas of table limits, Nomis-specific features and data storage.

The typical storage page size of a RDBMS is 2000 bytes. Individual records are unable to extend beyond page boundaries, so this limits the number of items in an individual record to between 200 and 300. Nomis data sets can have over 8000 data items for one record. Nomis data records are used to store the data items for a single geographical area for a single date. Of the 99 tables defined in the 1991 Census of Population Local Base Statistics, 25 have too many data items to fit into a single Ingres record and 31 have too many to fit into a single Oracle record.

One possible solution to this problem is to reorganise the Nomis data for a single area/date into several records. However, this would result in a 350% increase in storage required for any data set that could not fit its data into a single record.

Increased processing overheads would also be introduced, as several records would have to be accessed instead of a single record.

The 'Old Nomis' system recorded audit information regarding the number of items accessed by individual users. This information is used to provide management information and also to allow data costs to be charged and passed on to data suppliers (where appropriate). Indeed, it is the powerful and comprehensive audit capability which has been a major strength in negotiating with data owners to put their data on the system. They understand that the extensive feedback mechanism both informs them of data usage and encourages ethical data usage by the user community. All commands entered into Nomis are logged, which further benefits help-desk support and allows suitable system monitoring. This type of data logging is not readily available within standard RDBMS systems and would require extra coding if this type of system were chosen.

Many data sets display warning messages along with the data. Selecting and displaying data using SQL would not allow these warning messages to be displayed. A considerable amount of programming would be required to build this information into the system.

The 'Old Nomis' system currently held (mid 1999) approximately 6 physical Gb (gigabytes) of data in a highly compressed format. Uncompressed this becomes 125 Gb. Using a RDBMS the data would have to be stored in an uncompressed form as well as being stored in tables that use hash-indexing, to allow fast access to data records. The result of this would be to increase the disk storage required to approximately 125 Gb. Furthermore, one of the new data sets required on the new system is USOCAD (Unemployed Stocks by Occupation, Age and Duration). This is a very complicated quarterly data set, requiring an extra 5Gb of storage each quarter if we were to use a typical RDBMS. Although large disk storage is no longer a problem on cost grounds, there must be a limit to the size of a database that can be sensibly maintained and a limit to the cost benefits of using RDBMS technologies.

3.2.2. "Home Grown" system

The 'Old Nomis' system was a "home grown" system developed using FORTRAN 77, a language designed to solve scientific problems. We accepted that the user service could greatly benefit from a rewrite taking advantage of modern programming language technology. In particular a more modular design using object-oriented programming (OOP) techniques would greatly facilitate the addition of new data sets and data update procedures. The advantage of the "home grown" approach is that the system can be written geared towards exact requirements as opposed to a "best-fit" approach.

3.2.2.1. Object-Oriented Programming

The OOP approach allows us to represent objects that we find in the real world as objects in software (Parsons, 1994). By identifying and classifying real world entities as objects, as well as defining the operations that need to be performed on these objects, a solution that accurately addresses the original problem can be developed. Unlike traditional programming methods, the OOP approach unifies inside each object two things

1. Data – this is stored in the private section of a particular object. Data can only be manipulated through methods stored in the appropriately named public section.

2. Methods – these are written to perform specific operations on data stored in the private section of an object. This builds in data security because only the required methods are written for each data element in a particular object.

Avison and Fitzgerald (1995) state that there are many different approaches to the analysis and design of object oriented systems (for Example, Booch 1991, Coad and Yourdon 1991, Martin and Odell 1992). These approaches assume a large development team and a greater level of resource than available at Nomis. The 'New Nomis' system is based on a simple model so identifying the objects and methods that are required (which is a major part of the analysis and design of object oriented systems) is not a major problem.

Using this approach results in the following benefits

- Easily maintainable code. With a strict set of methods or interfaces to each data set defined, programmers with limited knowledge of the system could easily add new facilities.
- Protection. Error conditions are confined to the module in which they occur. An error occurring in one data set module will not have a knock-on effect in any other data set module. This reduces the risk of errors caused by adding new data sets.
- Extensibility. New components can be developed from existing ones without any effect to the original components.
- The system is broken down into manageable units.

It was decided that the new system should be developed using C++. Parsons (1994) describes C++ as an OOP language developed by Bjarne Stroustrup in 1983. C++ derives directly from C in terms of syntax, but the object-oriented aspects are derived from earlier programming languages. The main reason for this decision was that both programmers in the development team has experience of using C, so the learning curve would be relatively easy.

3.2.3. Hardware Options

Only one hardware platform was available to develop the new system on a UNIX Workstation. The 'Old Nomis' system, although developed on an IBM mainframe, was moved onto a UNIX system in 1992. The hardware has proved both reliable and efficient as a platform for Nomis. This hardware is readily available to us via the University of Durham IT Service and the extra storage cost of accommodating the re-design is minimal. On the whole, the academic community in the UK has adopted UNIX as the standard operating system to use.

3.3. User interface

From a user's point of view, the most important part of any on-line database system is the gateway between the end user and the system itself - the user interface. 'Old Nomis' achieved this reasonably successfully through a simple command interface.

However, it was decided that for the new command interface the following design criteria should be achieved

- Standardise commands to achieve consistency.
- Simplify the commands where possible.
- Maintain reasonable backward compatibility with the previous Nomis system. It was accepted that with a large 'legacy' user base the cost of re-training should be kept to a minimum.
- Provide a 'menu-style' interface.
- Provide increased functionality of commands.
- Provide a context sensitive help system which could assist users at any stage during a query.

Although one of the requirements of the new system was to deliver a 'menu-style' interface, there remains a need for a basic command interface for the following reasons :

- To ease the transition for users from the 'Old' to the new system.
- Many of our users are restricted in the equipment they have to access our system, so we were unable to insist that everyone use a Web interface.
- A fast and efficient command language provides the hooks for a graphical system. The graphical interface generates a list of commands in the background, which are then passed through to the system and parsed in the same way as a command-style query.

These are the factors defining the redesign of the command interface. Taking this into consideration, a series of enhancements have been possible.

3.3.1. Context sensitive help

One major criticism of 'Old Nomis' was the steep learning curve for new users. As time went on this became a double overhead. New and complex data, more geographies, more discontinuities and more diverse end-user applications in themselves placed a substantial training overhead quite aside from the technicalities

of data access and manipulation. Without reading through an information pack, some trial and error, and attending training courses, there was no simple way of getting to grips with the system. There was no on-line way of showing what command to type in next, or which commands have already been entered. The new system introduces an easy to use context sensitive help system. Typing ***?*** or ***check*** at any stage informs the user which commands the system is expecting and which commands have already been entered. Typing ***help*** at any time will give a general help message relative to the position within a query. Similarly typing ***help*** ***<command>*** will give help on a particular command relative to its use within that particular query.

3.3.2. Improved error messages

Error messages have been greatly improved. Figure 3.1 shows the resulting output from an invalid date being requested on 'Old Nomis' such as September 1990 for the data set ***USW91*** (Unemployed Claimant Stocks – 1991 Ward Base)

```
£££ Error: 1049
£££ Data=usw91 for 0990
£££ File does not exist or access not allowed at present,
£££ or you've requested the wrong time period!?
£££ Check on LATEST and MISSING in HELP INFORMATION
```

Figure 3.1. 'Old Nomis' date error message

This error message would be produced at the end of the query and not immediately after the date command. This is due to the way that the previous system parsed commands, where a lot of major checking is not done until the very end of the query. Figure 3.2. shows the resulting output from an invalid date being entered into the new system.

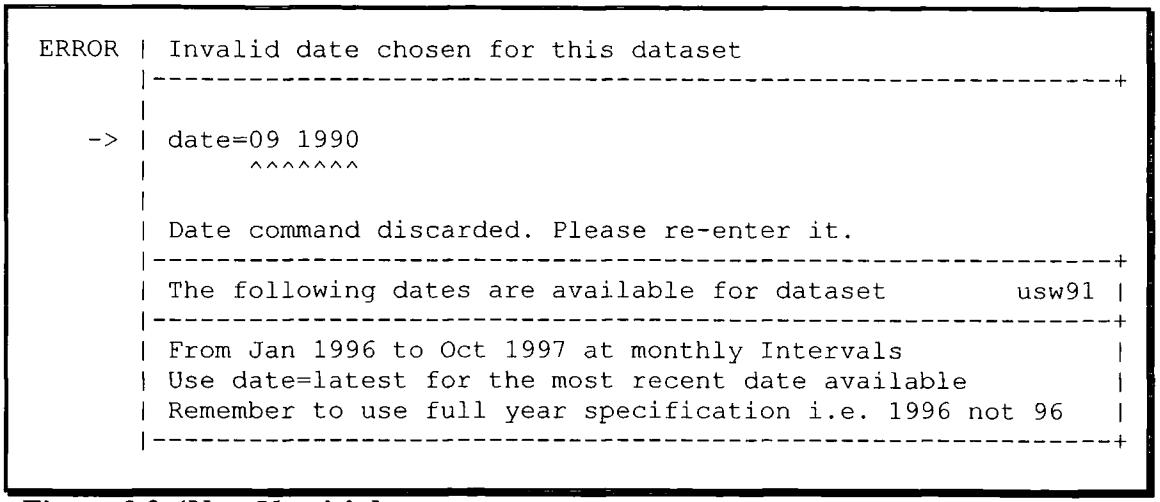


Figure 3.2. 'New Nomis' date error message

This error message occurs immediately after the ***date*** command. This is because the new system is structured in a way that allows each element to be checked as soon as it is parsed. The error message is clearly defined, as is the position within the command where the error occurs. A list of available dates is also shown and this demonstrates a use of context sensitive help within the new system.

3.3.3. A logical command structure

The previous Nomis system structured queries in such a way that the output command was the last command specified prior to receiving data. This restricted the amount of checking that could be done after each command. That is, if an average analysis was required, this command would be specified at the end of the query and only then could the system check if the user had requested more than one date.

The new system has been structured differently to improve the amount of checking that can be done. The first two commands entered are always the data set required, and the analysis to be performed (which is defaulted depending on the data set selected). The data set will always be the first piece of information requested by the new system as the remainder of the query hinges on which data set has been selected. The analysis is required because this, together with the data set, determines which other elements are required by the system.

A metadata file exists for each data set containing relevant data set information. This information in conjunction with the data set and analysis allows complete checking of each query component without having to access the data itself.

Figure 3.3. shows how an individual query is deconstructed

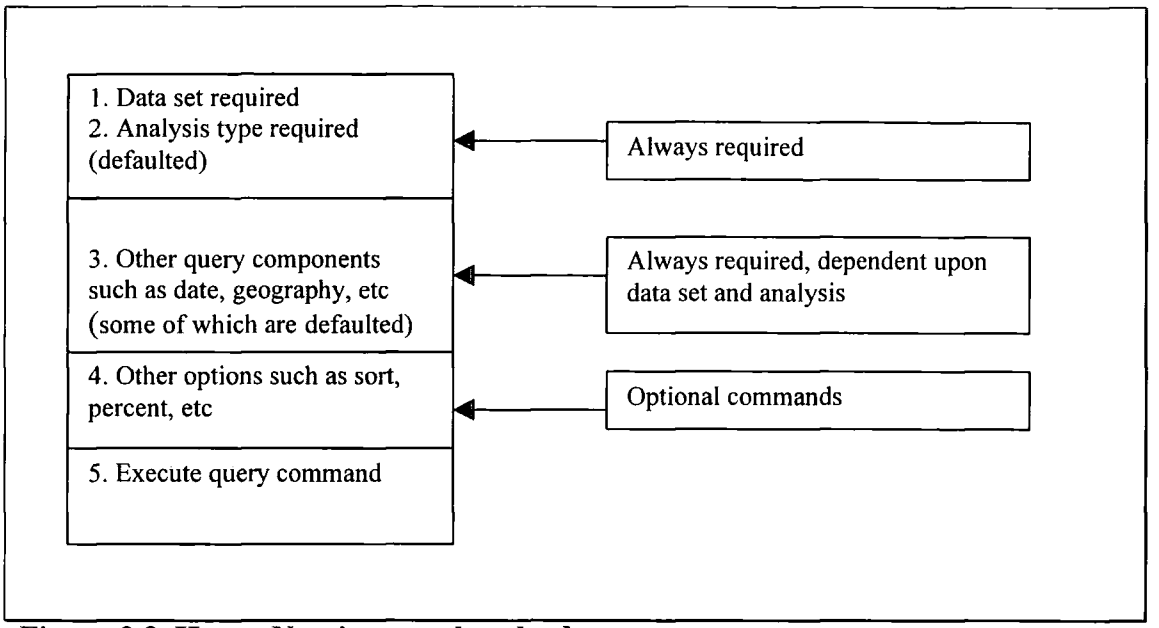


Figure 3.3. How a Nomis query breaks down

Users only need to know which data set they are interested in to initiate a query. The rest of the query components required to complete that query are determined by this initial selection. Therefore as soon as the data set is selected a user can type *?* or ***check*** at any time to see which query components are required to complete a particular query (see figure 3.4. below)

```
              * | data us
                | Initial setting : date latest (June 2000)
                | Initial setting : sex total
                | Initial setting : item 1
              * | ?
----------------+-------------------------------------------------------------+
command         | specified so far
----------------+-------------------------------------------------------------+
data            | data us
analysis        | analysis standard
date            | date latest
geography       |    --> Not specified yet
sex             | sex total
item            | item 1
layout          | layout automatic
----------------+-------------------------------------------------------------+
No Options Specified
----------------+-------------------------------------------------------------+
```

Figure 3.4. Displaying Nomis query components

3.3.4. Default selections

One advantage of the new command structure illustrated in figure 3.4. is that the system can make a 'calculated guess' at certain selections. With the data set **US** (Unemployed Claimant Stocks – 1991 Ward Base) the new system can assume that the user would require a total gender breakdown, so the command ***sex total*** (i.e. male + female) is selected by default. If the user does not require a total gender breakdown then they can type in the gender selection that they do require. The default selection will then be overridden with the new request. The new system also assumes that for data set ***US*** users are generally only interested in 'Wholly Unemployed Claimants', which is also defaulted as ***item 1*** in figure 3.4. Another assumption made with the above query is that data are required for the latest date. This is due to the fact that the data set ***US*** is updated each month on "Press Release" day. A lot of users access Nomis once a month just to retrieve the up to date figures, so the command ***date latest*** is also assumed for this data set. Generic dates are also allowed so that pre-set command files can be stored without any need to edit date fields. For example; ***date latest*** will always access the latest date; ***date previous to latest*** will take the two most recent dates available; ***date previousyear to latest*** allows data extraction over the most recent yearly period.

3.3.5. Selecting all values within a query component

Query components are commands such as date, geography, sex and item. These are used to make up part of a Nomis query. It is now possible to select all of the values available within a query component.

Typing ***date=all*** selects all dates available for a particular data set. With the previous system, if the following dates where available, January 1996 to November 1997, these dates would have to be looked up and then entered as ***date=196-1197***.

This facility is available for other breakdowns such as geography types, which have a far more complicated breakdown than for dates. Typing ***lad91=all*** selects all local authority districts built up from 1991 frozen wards. In order to select these areas with the previous Nomis system a user would need to know that there are 489 local

authority districts before they could select them. One software imposed limitation of the previous system was only a maximum of 1499 areas could be selected within a singe query. It is now possible to select all 11330 wards if required.

Implementing this facility for selecting all elements also brought up a major change to sex selection. Typing ***sex=all*** in 'Old Nomis' selects a single element which equates to 'all persons', and to select all individual gender breakdowns the command was ***sex=male,female,all.*** A decision was made that for the new system to remain consistent, typing ***sex=all*** would be used to select all individual gender breakdowns and typing ***sex=total*** would be used to select 'all persons'. This is one example of where the old and new systems clashed. There was no simple way of avoiding this because the new command overcame previous logical inconsistencies. Clear user documentation was prepared and circulated well in advance to start the re-education process.

3.3.6. Improved date handling

It is now possible to enter date information in a variety of formats. It was also decided that the full year would be required. That is '1997' as opposed to '97' to avoid any "Year 2000" compliance problems.

Users can now request all available dates in a single command, but more importantly, without having to use the help facility to lookup which dates are available. This is also the case when asking for the latest available date, a facility which will assist any of our users who use command files to pull off the most recent figures on a regular basis. Acceptable date formats are listed in Figure 3.5.

Action	'Old Nomis'	'New Nomis'
Select the date January 1997	date=0197	Date=01 1997 or Date=jan 1997 or Date=January 1997
Select the year 1997	year=1997	Date=1997
Select all dates available	N/A*	Date=all
Select the latest date available	N/A*	Date=latest
Select all dates available between June 1996 and February 1997	date=696-297	Date=06 1996 to 02 1997 or Date=jun 1996 to Feb 1997
Select all dates between Jan 1996 and the latest available date	N/A*	Date=01 1996 to latest or Date=jan 1996 to latest
Select all dates between January 1996 and December 1996 at 3 month intervals	date=196-1296 &by=3	Date=01 1996 to 12 1996 by 3

Figure 3.5. Acceptable date formats

* Only possible by users knowing what dates were available

3.3.7. Command review and re-entry

The ***review*** command lists all of the commands entered in a particular session. This is useful if a user wishes to recall a command typed in earlier to be used again in a new query. Each command listed by ***review*** has a unique command number associated with it. If a user wishes to re-enter a particular command they simply use the '*!*' character followed by the number they require. Only successful commands are stored for the review facility so that only valid commands may be re-entered. Figure 3.6. illustrates an example of the review command.

```
              * | review
----------------+-------------------------------------------------------------+
                1| data ucad
                2| make "ages 17-19yrs unemployed 1-8 weeks" age 2-3 plus dur 1-5
                3| region all
                4| go
                5| age total
                6| go
                7| data udad
                8| go
                9| review
----------------+-------------------------------------------------------------+
              * | !2
   re-issuing   | make "ages 17-19yrs unemployed 1-8 weeks" age 2-3 plus dur 1-5
```

Figure 3.6. An example of the review command

The above example illustrates a typical user query. A particularly complex 'age and duration' has been selected shown above as command number '2' (ages 17-19 years unemployed for 1-8 weeks). Having selected further commands and data sets, if the same 'age and duration' command is required again, by typing in ***review*** and then ***!2*** the complex 'age and duration' command is automatically re-issued.

3.3.8. The ability to save and re-use popular queries

By using the ***save*** command a user can save their current query components into a save file of their choice. Then all that is required is to type ***read <filename>*** and the new system would process that query. A good use of this facility would be in conjunction with the latest date function. The same query file could be loaded each month just after "Press Release" to pull off the latest unemployment figures. Figure 3.7. illustrates the saving of a typical Nomis query.

```
                *  | check
                   |
-----------------+-------------------------------------------------------------+
command            | specified so far
-----------------+-------------------------------------------------------------+
data               | data us
analysis           | analysis change
date               | date previous to latest
geography          | region all
sex                | sex total
item               | item 1
layout             | layout automatic
-----------------+-------------------------------------------------------------+
No Options Specified
-----------------+-------------------------------------------------------------+
                *  | go
-----------------+-------------------------------------------------------------+

 Title      : Unemployment Counts and Rates
 Source     : ONS Crown Copyright Reserved [from Nomis on April 17 2000]

 Item Name  : Wholly unemployed claimants
 Sex        : persons

  Jan 2000    Feb 2000      Change    %Cha     region

    134,949     133,222      -1,727    -1.3    South East
     35,062      34,859        -203    -0.6    East Anglia
    191,824     190,689      -1,135    -0.6    London
     75,158      74,327        -831    -1.1    South West
    117,277     116,089      -1,188    -1.0    West Midlands
     78,436      78,128        -308    -0.4    East Midlands
    123,022     122,306        -716    -0.6    Yorkshire and Humberside
    146,231     145,609        -622    -0.4    North West
     91,625      90,784        -841    -0.9    Northern
     64,435      63,455        -980    -1.5    Wales
    133,963     133,516        -447    -0.3    Scotland
     44,400      44,018        -382    -0.9    Northern Ireland
  1,236,382   1,227,002      -9,380    -0.8    Total
       Filename? | unempl
    Description? | latest regional unemployment change
              -> | Saved file unempl
```

Figure 3.7. A 'saved' Nomis query

The query was saved on April 17th 2000 (just prior to the "Press Release" on April 19th 2000). The ***check*** command was issued just prior to running the query to illustrate the query components that had been specified. Because the query was saved containing the command ***date previous to latest***, when the query is read in after Press Release on the 19th April data is produced shown in figure 3.8.

```
                | read unempl
----------------+-------------------------------------------------------+

Title       : Unemployment Counts and Rates
Source      : ONS Crown Copyright Reserved [from Nomis on April 19 2000]

Item Name   : Wholly unemployed claimants
Sex         : persons

 Feb 2000    Mar 2000      Change    %Cha     region

  133,222     128,035      -5,187    -3.9     South East
   34,859      33,792      -1,067    -3.1     East Anglia
  190,689     187,565      -3,124    -1.6     London
   74,327      70,639      -3,688    -5.0     South West
  116,089     113,488      -2,601    -2.2     West Midlands
   78,128      75,861      -2,267    -2.9     East Midlands
  122,306     118,579      -3,727    -3.0     Yorkshire and Humberside
  145,609     141,781      -3,828    -2.6     North West
   90,784      89,077      -1,707    -1.9     Northern
   63,455      61,761      -1,694    -2.7     Wales
  133,516     130,564      -2,952    -2.2     Scotland
   44,018      43,156        -862    -2.0     Northern Ireland
1,227,002   1,194,298     -32,704    -2.7     Total

----------------+-------------------------------------------------------+
```

Figure 3.8. Reading in the saved Nomis query

It is worth noting that the system has automatically selected the 'new' previous to latest dates (this time February to March 2000 instead of January to February 2000 shown in figure 3.7.)

3.3.9. Simple geography selection

The way that 'Old Nomis' handled geographical codes was efficient and reasonably easy for basic areas. For example :

ua91=1-10

would select the first ten unitary authority areas.

It was decided that the method for selecting simple geographical areas would not change significantly, but greater functionality should be provided. This improved functionality is possible due to a new intelligent geographical management system (see Chapter 4) which is aware of the relationship between different geographical types. The improvements made are explained below.

3.3.9.1. Geography aliasing

When using 'Old Nomis' a user had to specify the full name of the geography type that they were interested in. If a user was looking at the data set USW91 and were interested in the data for County Durham, they would have to use the command

county91=44

The "91" is necessary because 'Old Nomis' could not distinguish between different geography versions and, therefore, required the full name to be specified. The new system has a details file specific to each data set which contains geographical information such as which geography types are valid, their geography version and also a geography alias. This information makes it possible to alias geography types, so now for USW91 a user can select County Durham by typing

county=44

The main difference between the two data sets ***USW81*** and ***USW91*** is the fact that the first one uses 1981 frozen ward based geographical areas and the second uses 1991 frozen ward based geographical areas. If these data sets are combined (which is planned for late 2000) the new system can distinguish between the 1981 and 1991 geographic areas using the date specified. This is beneficial to the user because they do not have to worry about specifying which frozen ward base they are interested in. It will also allow users to retrieve "time-series" data for geographies across frozen boundaries using aliases where possible.

3.3.9.2. Different geographical types available in geography selection

Selecting more than one type of geographical area was very tedious using 'Old Nomis', requiring a user-defined area for each geographical area and then combine those definitions into a file known as a chain file. For the new system a simple and efficient solution is provided, using the ***and*** command. Simply join together the

geographical requests using ***and***. To select County Durham and also the Northern Region the command is :

county=44 and region=9

As many different geographical types can be specified as required, as long as each one is separated with the ***and*** command. For example :

lad=1-10 and county=44-50 and region=1-5

would allow users to select several local authority districts, counties and regions.

3.3.9.3. Selecting geographical areas found within larger areas

This facility allows a user to select a number of small areas within larger ones as long as a relationship exists between the two geographies. If a user wanted to select all of the local authority districts in County Durham then they would type :-

lad in county=44

Ranges are also supported so selections such as all of the wards in the first ten local authority districts will be handled by the new system. i.e.

ward in lad=1-10

3.3.9.4. User-defined areas

The ***make*** command is used to create an amalgamation of areas not available as standard geographies. These are known as user-defined areas. Only geographic areas available for the data set selected can be specified. Different geographical area types can be combined using the + operator. With a ward-based data set an example is :

make "Made up Area" lad=1-10

This command would produce a single area figure for the first ten local authority district areas. ***Make*** is the standard command for creating aggregate selections. It can also be used to aggregate query components such as occupations, industries, age/durations, sizebands and cells.

3.3.9.5. Improved geographical lookup facilities

Two commands have been introduced to assist geographical querying

1. ***identify string***
2. ***list geogtype=nn***

Identify is a simple geographical search tool where the name of the area required is typed in and the system reports any areas which match with that string. Figure 3.9. shows the result of typing the following command :

identify reading

Found 18	
Geog Command	Geography Name
Lad91=7	Reading
Pca91=8	Reading East
Pca91=9	Reading West
Pca95=10	Reading East
Pca95=11	Reading West
Ttwa8491=30	Reading
Ualad91=7	Reading
Aa9391=30	Reading
Lad81=7	Reading
Pca81=8	Reading East
Pca81=9	Reading West
Ttwa84=30	Reading
Ubonet=179	01763 Reading ESJ
Ubonet=180	01764 Reading B ESJ
Uboesd=17	Reading

Figure 3.9. Search results for "reading"

So if interest is in the local authority district area for Reading the command is :

lad91=7

The list function is equally simple to use but has a few more optional parameters than the ***identify*** function. Type in ***list*** followed by a geography type then press ***<return>*** to receive a list of all of the geography numbers and names for that particular geography type.

list region

Figure 3.10. shows the results from the list command

Code	Name	Type
1	South East	Region91
2	East Anglia	Region91
3	London	Region91
4	South West	Region91
5	West Midlands	Region91
6	East Midlands	Region91
7	Yorkshire and Humberside	Region91
8	North West	Region91
9	Northern	Region91
10	Wales	Region91
11	Scotland	Region91
12	Northern Ireland	Region91

Figure 3.10. A list of Nomis regions

This can be broken down further, for example :

list region=1-5

It is also possible to list small areas within larger ones if there is a relationship between the two geographies. List all of the counties in the South East Region by typing :

list county in region=1

Figure 3.11. shows the output that is produced from the previous list command

Code	Name	Type
1	Bedfordshire	county
2	Berkshire	county
3	Buckinghamshire	county
4	East Sussex	county
5	Essex	county
6	Hampshire	county
7	Hertfordshire	county
8	Isle of Wight	county
9	Kent	county
10	Oxfordshire	county
11	Surrey	county
12	West Sussex	county

Figure 3.11. A list of Nomis regions

3.4. Data models

Most of the limitations of 'Old Nomis' hinged around the access and storage of the data. A suitable method of data storage and retrieval had to be established assuming that the following was known about the data :

1. The date is known.
2. The geography is known.
3. The storage order is not important.
4. Quick, simultaneous retrieval is important at press release.

The method identified to achieve the solution is to use hash tables.

3.4.1. Hash tables

Scott Robert Ladd (1996) states that hash tables contain a fixed number of buckets. Each bucket is either empty, or contains data associated with a key. A function, known as a hashing function, is used to calculate the bucket number using the key. The hash function will always generate the same bucket for a given key, so that the key always retrieves the same data from the same bucket. For example:

Bucket	Contents
0	Empty
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty

The table to the left represents an empty hash table.
If the hash function ***F*** is

$$F(\text{key}) = \text{key} \bmod 7$$

This function will always produce a bucket number between 0 and 6.

Bucket	Contents
0	Empty
1	Empty
2	Empty
3	367,data
4	Empty
5	Empty
6	Empty

If the first record we store has the key value of 367 then the function

$$F(367) = 367 \bmod 7 = 3$$

places the record at bucket 3 in the hash table.

Bucket	Contents
0	Empty
1	Empty
2	Empty
3	367,data
4	Empty
5	Empty
6	1014,data

If the next record has the key value of 1014, we get

$$F(1014) = 1014 \bmod 7 = 6$$

which places the record at bucket 6 in the hash table

Bucket	Contents
0	Empty
1	Empty
2	Empty
3	367,data
4	Empty
5	Empty
6	1014,data

Suppose we now try to insert a record with the key value of 255. Then

$$F(255) = 255 \bmod 7 = 3$$

A problem has occurred, bucket 3 already contains another record. This is known as a collision, i.e. when two different values hash to the same location

It is important to design a collision resolution strategy because collisions are a fact of life when hashing. The solution implemented in the new system was to use a single linked list structure. If we use the above collision example and concentrate on bucket number 3, the new system would handle the collision in the following way.

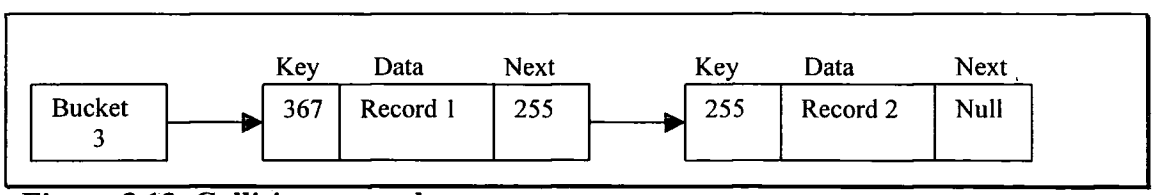


Figure 3.12. Collision example

Bucket 3 points to the first element in the linked list, in this case that is record 1, which has the key value of 367. A second record would now be inserted into the list. The next pointer for record 1 would be changed from null (because there was initially no other records in the list) to point to the second record in the list and consequently its next pointer would be set to null. Bucket 3 now contains two data records that can be easily identified, and therefore retrieved, by using their unique key values.

A quick and simple way of optimising data access for the new system was to store the latest data at the head of the linked list. The assumption was made that generally users would be more interested in the most recent data available on the new system.

As data records are stored in compressed format, they are variable in length. Each new record is appended into the file, so this presents no problems until deleting is required. The reallocation of deleted record space is handled using a technique called free listing. A free list is essentially a list of available locations in a file.

Speed of retrieval is important when accessing the new system but storage limitations are not an issue. Expensive free-listing strategies are not required as most existing data in Nomis are rarely modified. Therefore, a simple free-list strategy was developed using a Last In First Out (LIFO) stack system. Each free list element contains a location and a size. Every time a new record needs to be added, its size is known, so the system checks the first element of the free list to see if it can accommodate the new record. If the new record is too large then the system moves to the next free list element and tries again. Eventually if none of the free list elements are large enough for the new record then it is simply appended to the end of the file.

Using hash tables allows the 'New Nomis' system to store data in one file per data set. This results in a faster retrieval of "time-series" data than with the 'Old Nomis' system. Data for these files are stored in compressed form using the same compression techniques as the 'Old Nomis' system as these were found to be suitable efficient and storage constraints are not as tight as they were in the 1980s on the 'Old Nomis' system.

3.5.Conclusion

The new Nomis system is being developed using object-oriented programming technology, using the C++ language. Initially a command driven system was developed first and then a WWW front-end using CGI was designed at a later stage. A SWOT analysis of 'Old Nomis' was used to prioritise new developments while retaining many of the excellent characteristics of the 'Old' system. This both built on achievements and also helped to manage the transition for users from 'Old' to 'New'. Another important part of the redesign was to introduce ways of improving the geographical limitations of 'Old Nomis' and investigates ways of improving all areas of the geographical management system. This will be the focus of chapter 4.

Chapter 4 Geographical Management and Metadata

4.1. Introduction

The geographical management within Nomis is based upon 'building blocks' which comprise the smallest available geographical unit. With Labour Market statistics such as employment and unemployment the electoral ward is the geographical base, with some 12,000 individual wards covering the UK. Two other building blocks exist – postcode sectors and job centres. None of the three basic blocks intersect when overlaid. An additional challenge is that all three basic geographies are dynamic. They change frequently and not at the same time.

The geographical structures which are built using the building blocks are 'higher aggregations'. For example Unitary Authorities, Training Enterprise Councils, Regions can be built up from wards. It is also possible to 'best-fit' postcode sectors and job centres to some higher aggregations. The importance of the 'best-fit' process is in the need for local and regional government in particular to use coherent statistics for all labour market variables, and vacancy statistics in particular are not based on wards.

The technical challenge in managing geographical definitions involve tasks such as the maintenance of a complex, dynamic and growing geographical 'dictionary', but the crucial performance issue hinges on the need for speed. The release of data once a month at "Press Release" (currently 0930 on a Wednesday) leads to significant customer demand in the hours immediately following release. Customers may want any data from the 130 gigabyte (as of late 2000) holdings, so there is a need to ensure that the processing task of 'reading' data is as short as possible.

Under the 'Old Nomis' system, data for larger administrative areas (such as districts or counties) were pre-aggregated and stored to speed up access time. This was due to the legacy of hardware limitations in the 1980s and presented no immediate problem then as Nomis only had a small number of geographies that generally remained static.

4.2. 'Old Nomis' Geographical management

Although 'Old Nomis' had a geographical management system in place it was limited in 'intelligence'. There were only two basic hierarchical structures stored in two separate databases:

1. Ward or postcode definitions of all larger administrative areas.
2. Definitions of standard statistical regions in terms of all smaller administrative areas.

The relative stability of the geographies during the 1980s allowed a software strategy to focus on speed of delivery. Rather than provide data for areas at retrieval time as demanded by users, data were pre-built into the higher aggregations during the data loading process. In this way the disk access times were minimised (access time minimisation also being a key feature of the file and directory structures discussed earlier) and more users could access more data at Press Release.

Since the early 1990s geographic areas have changed much more frequently, and these changes must be handled more efficiently by the new system. It was decided to design a new Nomis geographical management system to allow easy maintenance of constantly changing geographic areas.

4.2.1. 'Old Nomis' - building higher aggregate areas

The majority of Nomis data are received at ward level. Statistics for other administrative areas (such as districts or counties) are built up by adding appropriate wards together. The system first needs some way of establishing which particular wards are added together to build a particular higher aggregate area. These definitions are stored in a series of master files. One master file exists for each region. Each ward within that region has a single line entry consisting of its ward number then the code for the appropriate higher aggregate area to which it belongs. A section of the South East region file (using 1991 Census of Population wards as the base) is shown in figure 4.1.

```
  1       2    3   4   5   6   7   8   9  10  11
0101001 00001 001 009 001 114 003 037 201 137 Biscot
0101002 00002 001 009 004 114 002 037 201 137 Bramingham
0101003 00003 001 009 004 114 002 037 201 137 Challney
0101004 00004 001 009 001 114 003 037 201 137 Crawley
0101005 00005 001 009 001 114 003 037 201 137 Dallow
0101006 00006 001 009 001 114 003 037 201 137 Farley
0101007 00007 001 009 001 114 003 037 201 137 High Town
0101008 00008 001 009 004 114 002 037 201 137 Icknield
0101009 00009 001 009 004 114 002 037 201 137 Leagrave
0101010 00010 001 009 004 114 002 037 201 137 Lewsey
0101011 00011 001 009 004 114 002 037 201 137 Limbury
0101012 00012 001 009 001 114 003 037 201 137 Putteridge
0101013 00013 001 009 001 114 002 037 201 137 Saints
0101014 00014 001 009 001 114 003 037 201 137 South
0101015 00015 001 009 001 114 003 037 201 137 Stopsley
0101016 00016 001 009 004 114 002 037 201 137 Sundon Park
.
.
.
.
.
etc
```

Figure 4.1. Part of South East region master file

Each line holds several columns of data :-

1. Data address – a unique number used for direct access within the Nomis code
2. Ward91 code
3. Local authority district
4. Training Enterprise Councils 'frozen' at their initial boundary state
5. 1991 parliamentary constituency
6. Unitary authority
7. 1995 parliamentary constituency
8. 1984 travel-to-work area (based on 1991 wards but released for use in 1984)
9. Training Enterprise Councils – current state
10. 1993 assisted area
11. Ward name

These files can be used to generate simplified lists of wards in each larger administrative area. A simplified ward list is shown in figure 4.2.

```
16 21012010 Luton
0101001 0101002 0101003 0101004 0101005 0101006 0101007 0101008 0101009 0101010
0101011 0101012 0101013 0101014 0101015 0101016
```

Figure 4.2. Wards within the unitary authority Luton

The list file above informs the system that 16 wards make up the unitary authority area Luton (which has the data address 21012010). The ward list was generated by scanning the master file (shown in figure 4.1.) for all of the occurrences of the Luton Nomis code (which is 114) and adding the corresponding ward codes into the list.

These lists are used to build larger areas from ward data. Figure 4.3. indicates how each of the files interact to produce the higher aggregate data for unitary authority areas.

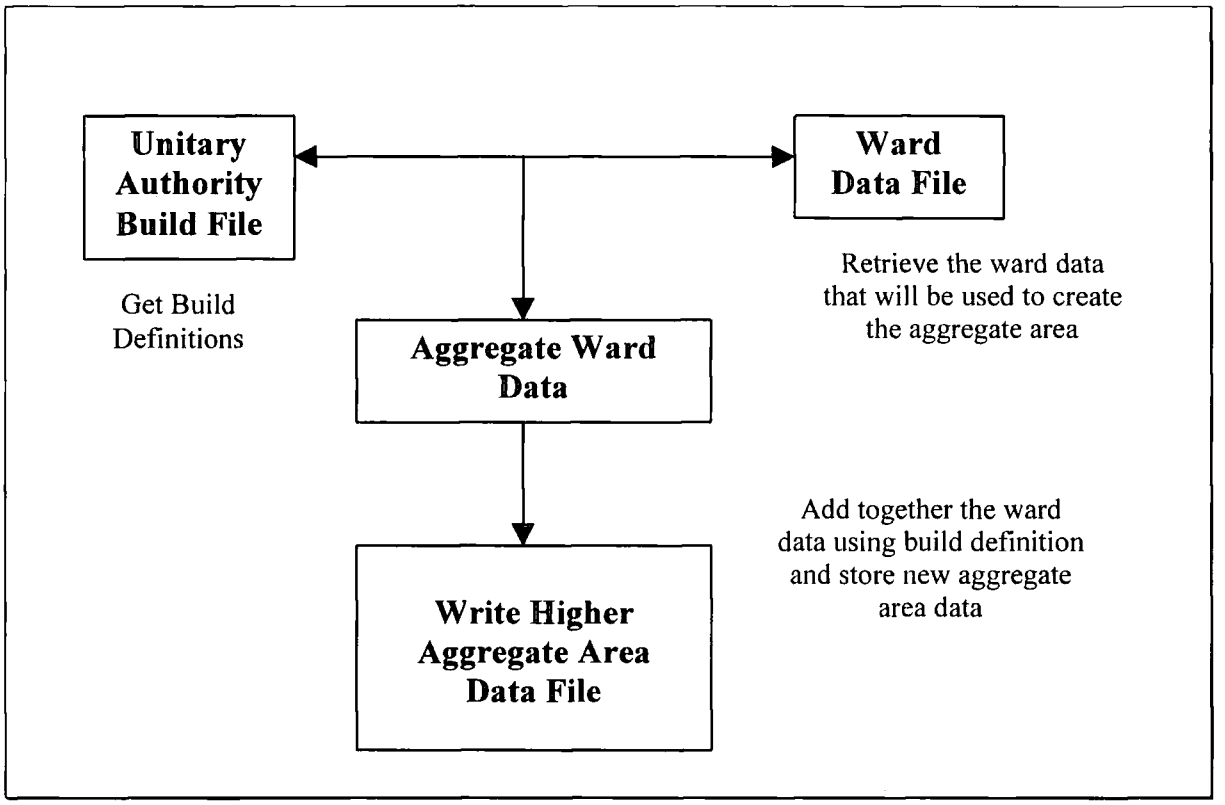


Figure 4.3. How 'Old Nomis' built higher aggregate areas

4.2.2. 'Old Nomis' - defining geographical relationships

Although the Nomis system does not store all possible relationships between compatible geographic areas, some relationship information is available. Each different geographical type has an information file containing specific information about that particular geography.

When the following command is entered into Nomis

county91=44

the system resolves the left hand side argument "county91" to determine the geography files it is interested in. The right hand side argument determines the line address in the ".dir" file for that geography. For the above example line 44 of the ".dir" file returns an address into the ".pag" file which contains the following record shown in figure 4.4.

```
44  48  9  6 Durham  30004400 30004401 30004402 30004403 30004404 30004405
30004406 30004407 30004408
```

Figure 4.4. Part of a geography information file

From left to right the fields are as follows

1. Key – also the county number
2. Record length (in bytes)
3. Number of geography codes to follow
4. Length of Name
5. County Name
6. County Nomis code
7. to 14. Make-up geography Nomis codes (Lad's)

Using the above information the software can determine that the county "Durham" is made up from eight local authority district areas (codes 30004401 to 30004408). If these local authority district records are looked up in the appropriate geography file the wards that go to build up these areas can be determined. Each geography type knows which smaller geography type it is made up from. This is how the hierarchy was defined in the 'Old Nomis' system.

The problem 'Old Nomis' had was that two databases were used, both based on master files but created at separate times. One database was used to build up data and another was used to look up definitions of areas in terms of other areas. As the

geographies used in Nomis became more complicated it was possible for the two to become out of step.

'Old Nomis' was not programmed to be map-oriented – for example it does not store adjacency data for areas (topology), nor does it provide geographical searching (for example all areas within 10 km of a ward). The reasons during the 1980s were a combination of development priorities set by the Department of Employment and the fact that boundary availability for our customers was limited. By the end of the 1990s PC-mapping and boundary availability put such techniques onto the desk-tops of users. Our priority still is one of a highly efficient data engine that delivers data rapidly for desk-top processing.

The 'Old Nomis' geographical structures, while remaining computationally efficient and delivering data rapidly, increasingly became an administrative burden. As geographies changed, and in particular as certain data series went through consistency problems (errors, for example, in the Census of Employment/Annual Employment Survey) frequent data re-loads were needed and the task of keeping track of the re-load was tedious and time-consuming.

The design of 'New Nomis' took the increasing volatility of geographies into account, and the geographical management system combines all of the geographical information into one database which is the focus of the next section.

4.3. The new geographical management system

The 'New Nomis' system needed to focus on two main areas for its new geographical management system:

- Information required by users. Providing more information about hierarchies. This is important because many new users come afresh to nearly 20 years of data and they do not have a detailed understanding of the legacy of past geographies.
- Areas used to build up other areas. This is for computational efficiency.

4.3.1. Information required by users

This first comprises basic geography information, for example area names. These are stored in a binary tree enabling fast access to basic geographical information for users.

4.3.1.1. Geographical information binary tree

(Horowitz 1976) defines a binary tree as a finite set of nodes which is either empty or consists of a root and two disjoint binary trees called the left subtree and the right subtree. The geographical information tree contains information for each individual geographical area stored in a binary tree. Figure 4.5. shows and example of a binary tree.

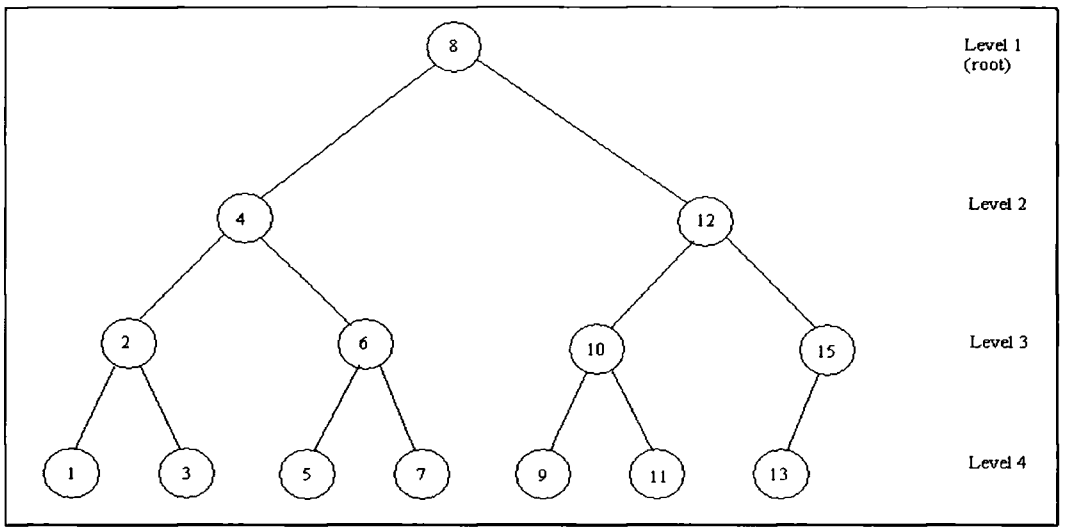


Figure 4.5. A binary tree

The tree is navigated by starting at the root and determining whether the key required is greater than, less than or equal to the node value. If the key required is six, then this is less than eight (the root) so continue to the left subtree (four). The key required (six) is greater than this so continue to the right subtree which is equal to the key value. This is the location of the record required. It has only taken three lookup attempts to locate the required value. The above tree only has four levels, meaning that the maximum lookup attempts to locate any record (one to fifteen) is four. 'Old Nomis' used a sequential file and a binary search to locate the required geographical information records which, using the same example, would also require a maximum of 4 lookup attempts. However, the advantage of using a binary tree becomes more

apparent when there is a need to add new records. Each node of a binary tree contains a pointer to the next left and right subtree, so insertion is just a matter of rearranging the pointers. Figure 4.6. shows the effect of adding a new record with key 14 into the example binary tree.

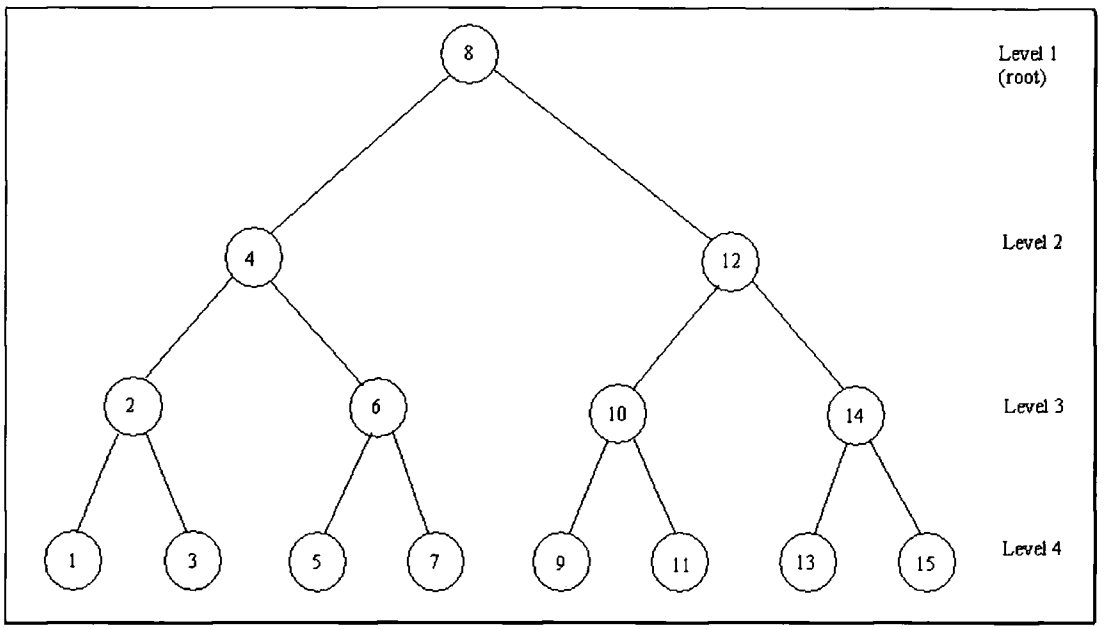


Figure 4.6. Adding record 14 to the binary tree

A new node for record 14 had been added at level 3 and the node for record 15 has been moved to level 4 and is now the right subtree for record 14. Record 13 is now the left subtree for record 14.

If a binary search is used, after a new record was added, all of the records would have to be sorted in order for the search to work. This would be inefficient as the geographical information file contains thousands of records.

The geographical information binary tree stores one record for each area and associates a unique key to each to enable the navigation of the tree. The table below shows the geographical information contained for standard statistical regions.

Key	Code	Type	'Old Nomis' key	Name
50331649	1	12	30300100	South East
50331650	2	12	30300200	East Anglia
50331651	3	12	30300300	London
50331652	4	12	30300400	South West
50331653	5	12	30300500	West Midlands
50331654	6	12	30300600	East Midlands
50331655	7	12	30300700	Yorkshire and Humberside
50331656	8	12	30300800	North West
50331657	9	12	30300900	Northern
50331658	10	12	30301000	Wales
50331659	11	12	30301100	Scotland
50331660	12	12	30301200	Northern Ireland

Figure 4.7 Geographical information for standard statistical regions

From left to right the columns are described as follows

Key - a unique key is given to each individual area (the formula is given below).

Code - corresponds to the Nomis code given in the Nomis manuals. For example region 1 (South East) is code 1 in the Nomis manual.

Type - each geographical type has a unique number to identify its type. For example standard statistical regions are type 12.

'Old Nomis' key – this is used when translating 'Old Nomis' geographies to new.

Name – that of the geography.

A unique key is calculated using the following formula

type* left shifted 22 places + *code

This formula is used so that all possible geographic codes can be represented in a 4 byte integer which identifies both the type and code and space even for very detailed geographies such as enumeration districts which number over 100,000[1]

[1] While it has previously been acknowledged that wards are the building blocks we do need to accommodate the possibility of storing Census of Population data to enumeration district level should ONS require it.

For example the "Newbury" local authority district is type "4" and code "6" (in the Nomis manual), and using the above formula

4 left shifted 22 places = 16777216

16777216 + 6 = 16777222

When the command

list lad 1-5

is typed the system uses the information binary tree to deliver the output shown in figure 4.8.

Code	Name	Type
1	Luton	lad91
2	Mid Bedfordshire	lad91
3	Bedford	lad91
4	South Bedfordshire	lad91
5	Bracknell Forest	lad91

Figure 4.8. The first five local authority districts

4.3.2. Areas used to build up other areas

To support users better 'New Nomis' needed full knowledge of the 'nesting' of all administrative geographies for example :

local authority districts in a county
counties in a government office region
government office regions in a country

This functionality is provided using two further components; first a geographical hierarchical binary tree; second a geographical reference table.

4.3.2.1. Geographical hierarchical binary tree

Each node of a binary tree contains a single geography code (the key), the geography type of the components and a list of keys which show how that individual geography is made up. Figure 4.9. shows the components that make up England.

Key	Type	Components
8388611	12	50331649
		50331650
		50331651
		50331652
		50331653
		50331654
		50331655
		50331656
		50331657

Figure 4.9. Country component file for 'England'

The value 8388611 is the key for England. The components are also listed by their key values. The components listed correspond to the nine regions in England.

This tree is accessed when using the geography "in" command and also to create higher aggregation areas as requested at retrieval time.

4.3.2.2. Geographical reference table

This is a matrix table that defines the relationship between different geographical types. It is used in conjunction with the geographical binary tree to avoid retrieving information for geographical relationships that do not exist. It also means that every relationship possible does not have to be defined. Figure 4.10. is a simplified table containing some of the cells from the real reference table.

	1:Ward	2:Country	3:County	4:Lad	5:Region
1:Ward	1	5	4	1	4
2:Country	0	2	0	0	0
3:County	0	5	3	0	3
4:Lad	0	5	4	4	4
5:Region	0	5	0	0	5

Figure 4.10. Geography relationship reference table

When the following command is typed in :

list ward in county 44

The system must establish if a relationship exists between wards and counties. The system works out the relationship by doing recursive references into the matrix until it receives the geography type value required, or a zero if the relationship does not exist. The first lookup is the cell at row 1 column 3 (to see if wards build into county areas). The value here is 4 (which equates to local authority districts) so the next lookup is row 1 column 4 (to see if wards build into local authority districts). The value here is 1 (which is ward) this indicates that wards build directly into local authority districts and from the previous lookup local authority districts build directly into county areas. As the matrix did not return a zero value, the system has established that a relationship exists between the two area types so it can go on to retrieve the information using the geographical hierarchical binary tree described above.

4.4. Handling frequently changing geography types

Geographic areas can change in two ways. Either the constituents of an area changes (this has happened regularly with Training and Enterprise Council areas) or if a completely new area comes into existence (which happens regularly with postcode sector and job centre geographies).

The 'New Nomis' system provides a facility for handling frequently changing geography types by using two processes; "Quick Build" definitions: Indirection files. Both of these methods are used to ensure that the minimum changes to the data are required and therefore less time is spent reloading data.

4.4.1. "Quick Build" definitions

'Old Nomis' would only pre-build areas that were strictly defined. Constantly changing geographical area types such as Training and Enterprise Council areas were never pre-built because every time an area definition changed all of the data for that area would need to be rebuilt. The solution was to always aggregate such areas at

retrieval time. As wards are the building blocks for Training and Enterprise Council areas then data could be aggregated from wards. However, this was not the most efficient method as these areas may be made up of a hundred wards. If a proportion of these wards made up intermediate areas (such as local authority districts) and Nomis could aggregate a combination of different area types then this would speed up the build. This process is known as a "Quick Build". Figure 4.11. shows part of the "Quick Build" definition file for the Training and Enterprise Council area for North Lincolnshire.

```
#-+-------------------------------------------------------------
2 |         71303234 North Lincolnshire
3 |              511 Quick build definition
4 |                  16777495 Glanford
4 |                  16777499 Scunthorpe
4 |                   4200781 27UBFB Crowle
4 |                   4200795 27UBFR Keadby with Althorpe
4 |                   4200798 27UBFU North Axholme
4 |                   4200801 27UBFY Trentside
#-+-------------------------------------------------------------
```

Figure 4.11. Part of "Quick Build" file for North Lincolnshire

This file informs the system that the Training and Enterprise Council area for North Lincolnshire can be quickly built by adding the figures from the local authority districts Glandford and Scunthorpe to the figures for the wards Crowle, Keadby with Althorpe, North Axholme and Trentside. The system only uses the numbers in the file and is not interested in the area names (included purely as comments). The line beginning with the 2 informs the system the area being built (71303234 is the unique Nomis key value for North Lincolnshire Training and Enterprise Council area). The line which begins with a 3 informs the system how the definition is built (511 is the geography type for quick build areas). Finally, the lines which begin with 4 are the actual area key values used to build up North Lincolnshire (1677495 and 1677499 are local authority districts, the rest are frozen wards).

4.4.2. Indirection files

'Old Nomis' used sequential codes to represent geographies. If these codes were used directly to look up areas then adding a new area would mean completely re-

sequencing the data. An intermediate file was introduced which links the Nomis code to the geography key known as an indirection file. This means that Nomis only needs to re-sequence the indirection file each time new areas come into existence and not the data.

Areas such as postcodes change every 6 months on Nomis when a new Central Postcode Directory is published (CPD). This results in new postcodes being available to Nomis users. Figure 4.12. shows the Nomis codes for the postcode sectors which begin "DH7".

Code	Name	Type
8106	DH 7 0 - Stanley	ps
8107	DH 7 6 - Durham	ps
8108	DH 7 7 - Durham	ps
8109	DH 7 8 - Durham	ps
8110	DH 7 9 - Durham	ps

Figure 4.12. Postcode sectors which begin "DH7"

As the Nomis command interface uses consecutive numbers to allow users to easily select ranges of areas (in this case ***ps=8106-8110***) if a new postcode sector was introduced "DH7 5" this would cause a problem. If the next available code for postcode sectors was 9300 then Nomis could allocate this to the new postcode sector. This would make a consecutive range selection a lot more difficult for users (i.e. ***ps=8106,9300,8107-8110).*** The other solution is to reorder all of the Nomis codes so that the areas beginning "DH7" would correspond with figure 4.13.

Code	Name	Type
8106	DH 7 0 - Stanley	ps
8107	DH 7 5 - Whatever	ps
8108	DH 7 6 - Durham	ps
8109	DH 7 7 - Durham	ps
8110	DH 7 8 - Durham	ps
8111	DH 7 9 - Durham	ps

Figure 4.13. Revised postcode sectors which begin "DH7"

'New Nomis' stores all geographic areas with a unique key value. This bears no resemblance to the actual Nomis code used at the command prompt. If a completely new area is added as in the example above the system creates an indirection file to reorder the Nomis codes. This file is essentially an array of size equal to the number of Nomis codes for that area type, each element containing the actual key value corresponding to that Nomis code. When the system does any geographical lookup for any area type it first checks for the existence of an indirection file. If the file does not exist then it is assumed that the geography has remained constant and had no new areas added.

4.5. NmGeography database

All of the facilities above are facilitated from a single geographical database called NmGeography. The programming solutions are written in C++. Binary trees, reference tables and indirection files are all readily available using C++. Figure 4.14. shows how the NmGeography database interacts with the different components.

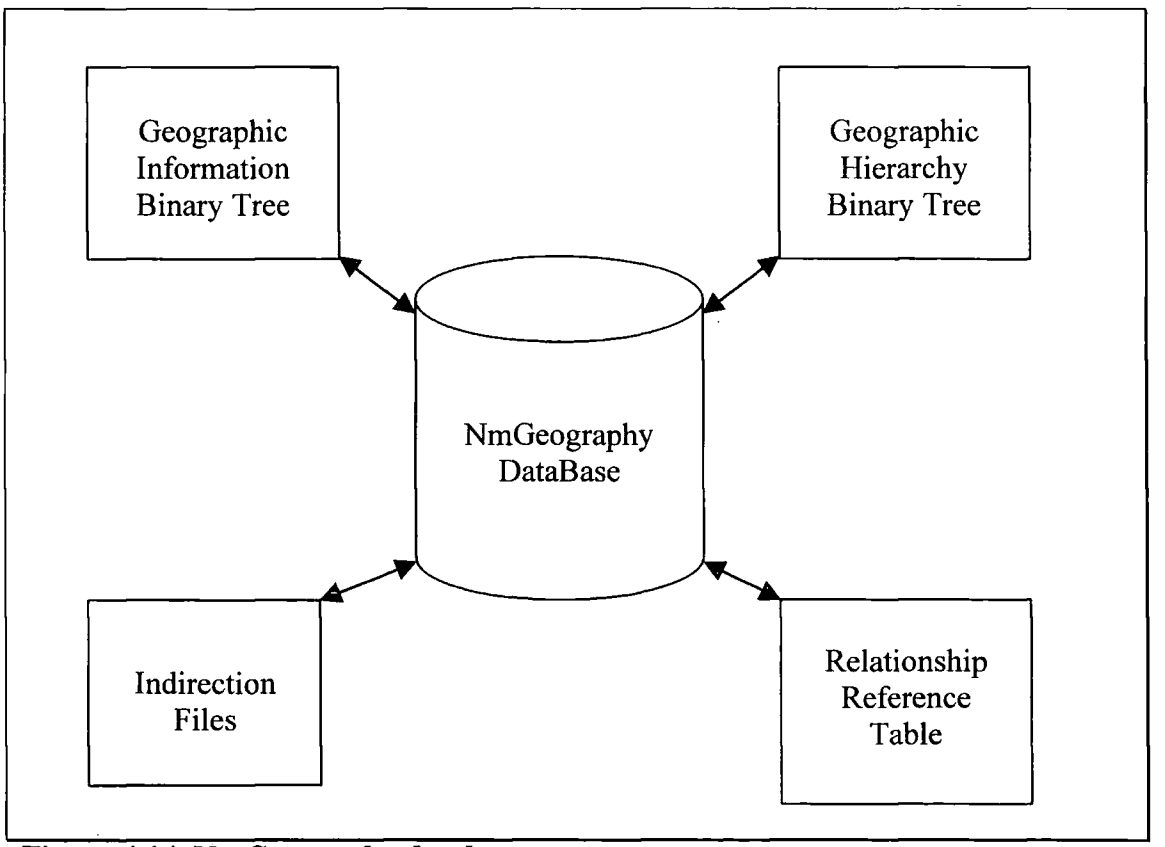


Figure 4.14. NmGeography database

4.6. Conclusion

'New Nomis' has improved the overall handling of geographical areas by developing a new geographical management system. This system improved the functionality of the command interface by allowing users more flexible lookup facilities. This system also made it far simpler to handle constantly changing areas such as teclecs and postcode sectors by means of indirection files and the "Quick Build" facility. The consistency of the geographies is always maintained because all of the information is stored in one database. It is intended that there will be further improvements to the geographical management system such as implementing unique geography mnemonics to all areas which will mean that Nomis codes remain constant over time. However, other improved facilities had a higher priority such as adding a "menu-style" interface to the Nomis system. This is discussed in chapter five.

Chapter 5 Adding a Menu-style Interface

5.1. Introduction

With the advent of Windows™ based applications on desktop computers, users expectations of a systems user interface have increased dramatically. Even during 1999, the Nomis command driven interface remains very popular with experienced users but can be difficult for the inexperienced to master. If Nomis had a Graphical User Interface (GUI) even novice users could retrieve data without too much training or documentation. At each stage of any query the appropriate choices for a user could be presented in a 'point-and-click' style giving the information they require on the screen to complete a Nomis query and retrieve the data they requested. Furthermore, the expansion of the user base has introduced users who do have a background in labour market data. Consequently a GUI could also embed more metadata and guidance as part of a strategy to develop efficient and effective usage of official statistics. With the development of the new Nomis system using Objected Oriented Programming with its standardised command interface, adding a GUI was seen as a realistic option. This chapter will discuss the different choices of technology available to provide a GUI. It will also show how the appropriate choice has been implemented within the new Nomis system.

5.2. Technologies

There are several technologies available to provide a menu-style interface. Many of them have different names but can be generally broken down into the following three categories; Character based menu interface; PC client/server interface; WWW interface.

5.2.1. Character based menu interface

This type of interface could be developed using the 'Curses' facility within the screen handling packages of UNIX. 'Curses', allows the development of full screen terminal-independent applications. The front end would be in constant communication with the background system so that validation could be done in real-time. This type of system

would work relatively well even with slow modem connections using 'vt100' emulation available with all major communications software packages. Figure 5.1. below shows an example of a character based menu interface used in the r•cade on-line systems for European statistical data developed from 1995 onwards at the University of Durham (http://www-rcade.dur.ac.uk)

Figure 5.1. Taken from the r•cade system

The main problem with this method is the fact that a character based menu interface is an intermediate step between a command driven system and a Graphical User Interface (GUI). Since the character-based system is in constant contact with the database there is a very real limit placed on the number of simultaneous accesses to the database – this number is fixed at the number of logins allowed. The rapid growth of the World Wide Web both allowed a more flexible interface to be created and would allow more flexibility in database access because the interface on the Web is separate from the database.

5.2.2. PC based client/server interface

A 'point-and-click' interface written specifically for PC's for use with Microsoft Windows™ operating systems could be developed. The facility for building up a query would use a method of storing 'up-to-date' metadata in a file stored locally on the client machine. This would allow command sequences to be built up into a command file that could then be sent to the Nomis system for processing once the query was complete. The system would process the command file locally, generate the output and transfer the results back to the client machine.

Several problems result from using this method for a 'front-end' to Nomis

- As the command file is being built up, the front end has no communication with the Nomis system directly, so all of the metadata required for query validation needs to be stored locally on the end-users PC. This means that updates to the system needs to be reflected in the metadata which then needs to be pulled across to the PC at regular intervals. An 'out-of-date' metadata file could result in a user not being able to download data for a new data series, incorrect data being returned or in the worst case no data at all.
- Metadata has to be stored locally on the users PC using up local disk space.
- The program that would generate the interface would have to be stored and executed locally. A lot of businesses have a policy of not allowing distributed software to be loaded onto office machines for fear of 'viruses' or other 'knock-on' effects caused by loading new software onto machines.
- Any updates to the program would need to be distributed each time to over 800 users adding a significant management and administrative overhead to the service.

Consequently this option was disregarded at an early stage.

5.2.3. World Wide Web interface

Three different Web development styles were considered initially; HTML forms; JavaScript; Java.

5.2.3.1. HTML, forms and the Common Gateway Interface (CGI)

Hypertext Markup Language (HTML) was developed as an authoring language for creating and sharing integrated electronic documents over the Internet (Musciano and Kennedy, 1997). At the time of writing most of the WWW pages accessed by users are produced using HTML. As far as the WWW is concerned the Internet connects two types of computers. Firstly, there are Web servers which make available (or "serve up") documents for retrieval. Secondly, there are "Clients" which are the machines which retrieve and display the documents for users. Clients access and display HTML documents by running browsers such as Microsoft Internet Explorer or Netscape Communicator.

If Nomis incorporated a WWW interface then basic HTML would not be enough as it is not truly interactive without the use of forms. Using forms it is possible to create documents that can collect and process user input and respond with suitable replies. Forms can be generated through many different programming languages. The Web server achieves this using the Common Gateway Interface (CGI). Gudavaram (1996) describes CGI as the part of a Web server that can communicate with programs running on the same server. Using CGI, the Form data can be passed to programs called up by the Web server. The data are processed, then the response is returned back to the Web browser in HTML. Figure 5.2. shows a simple model of CGI taken from Gundavaram (1996).

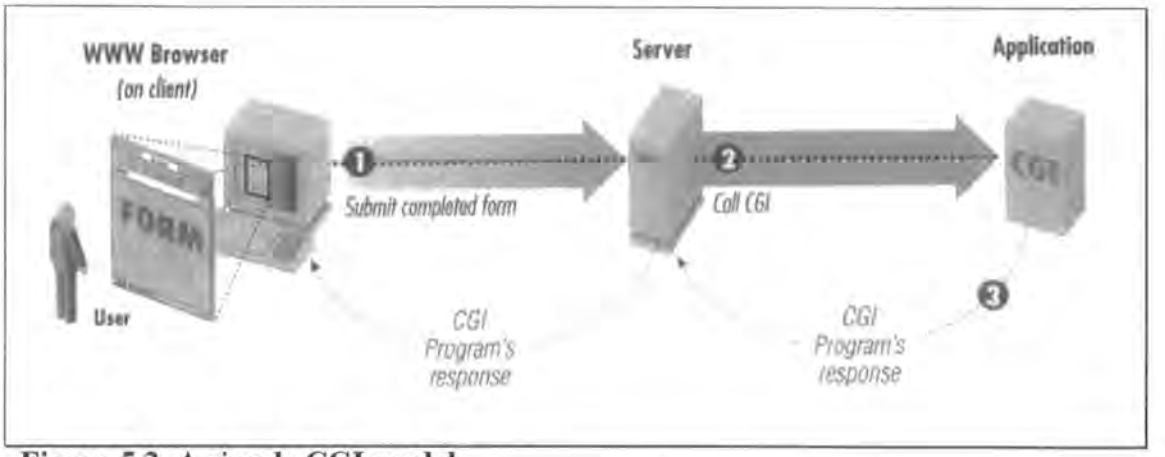


Figure 5.2. A simple CGI model

The Nomis front-end could be developed using HTML with forms. Nomis system forms could be developed allowing the user to specify the data set, date, geography, etc. by re-using existing output routines available for the command interface. The only difference is that these output routines would generate HTML as opposed to standard output.

The major advantage of using forms is that it is supported by all WWW browsers and that it would be possible to re-use existing code from the new command driven system.

5.2.3.2. JavaScript

JavaScript is an interpreted programming language written specifically for Web programming (Flanagan, 1997). Its constructs are similar to C, C++ or Java, allowing the 'if' statement and 'while' loops etc. However, JavaScript is entirely different syntactically as it is an "untyped" language, meaning that variables do not need to have a type specification.

JavaScript has no long-term storage or file access capabilities. JavaScript is designed to develop small applications, such as a calculator program. Larger applications such as the Nomis front-end would not be a feasible solution using this method. Also JavaScript produced Web pages can vary quite significantly depending on the Web browser being used.

5.2.3.3. Java

Not to be confused with JavaScript, Java is an Object Oriented Programming language developed by Sun Microsystems (Flanagan 1997). Java is not compiled like other programming languages. Java programs are translated into an architecturally-neutral format known as 'byte-code'. Each 'byte-code' instruction is then interpreted using a Java interpreter. Java has a strong relationship with the WWW because it allows the use of applets. A Java applet is a piece of source code whose 'byte-code' is executed when viewing part of a WWW page.

Although Java could be used to develop the Nomis front-end, several problems were identified:

1. Performance: Java is interpreted, not compiled and as such is noticeably slower. With the introduction of Java workstations (machines with built in Java interpreter processors) speed will not be a problem. But we had no way of mandating that all Nomis users implement Java on their machines.
2. Web browsers must have Java support. Many Nomis users have low-specification PC's and may only have WWW browsers that don't support Java, or have Java disabled.
3. Java security – the Nomis system contains sensitive, confidential government data. System security issues cannot be taken lightly. Weber (1996) states that a number of flaws have been identified with Java since its release. These identified the ability of users to access supposedly secure parts of systems. Even if newer releases have addressed these security 'loop holes' having confidence in using such a language is difficult.

5.2.4. Decision

Taking everything into consideration the decision was made to produce the GUI using the Web technology of HTML and forms[1]. The main advantages identified were

- Accessibility. Many Nomis users already have access to the WWW.
- No distributed software. Web browsers are freely available and supported by their producers.
- Unlike Java or JavaScript, HTML and forms are supported by all browsers.
- No need for updating metadata locally. HTML forms are generated from the Nomis system so the metadata is always up-to-date.
- Security. Users can only fill in HTML forms so the ability to get 'system side' is removed.
- Familiar 'point-and-click' interface. Most users are accustomed to using this type of interface unlike a character based menu interface.
- Code re-usability. With minor alterations the new Nomis command driven system can produce the HTML forms needed in the implementation of the WWW front-end.

However, there is a particular downside to using this method. Any mouse click submissions have to be sent back to the server for processing and appropriate responses returned. This puts a large load on the server and makes this sort of application significantly slower than equivalent Windows™ based applications. A new Sun Ultra 2 UNIX Server was introduced in mid 1999 to resolve any performance issues.

Another issue worth noting is Web browser compatibility. The appearance of a Web page on one browser may be different to the same page viewed on a different browser. This problem is purely cosmetic and can be addressed by testing the new interface on the most popular browsers.

[1] Although Java was not considered in mid 1998, it may still be a possibility later as the technology develops.

5.3. Implementation

The first stage of implementation was to decide on an appropriate domain name. The domain name is the part of the WWW address that uniquely identifies that particular site. The preferred choice was "www.nomis.co.uk" as adding "co.uk" to the end of an organisation name (such as Nomis) was the standard that all large organisations used. Unfortunately this address was already registered to a computer company known as 'Nomis Microsystems', so an alternative had to be found. After discussion "www.nomisweb.co.uk" was agreed upon and registered. The next stage in implementation was to decide on the order of information flowing between the user and the system.

Information from the user is vital for the WWW interface to succeed. The information received from the user and what the system returns will essentially determine the order of the query. At each stage of user input a new screen will be displayed as each HTML Form is completed and the new HTML returned to the screen. The forms required are shown in figure 5.3 below.

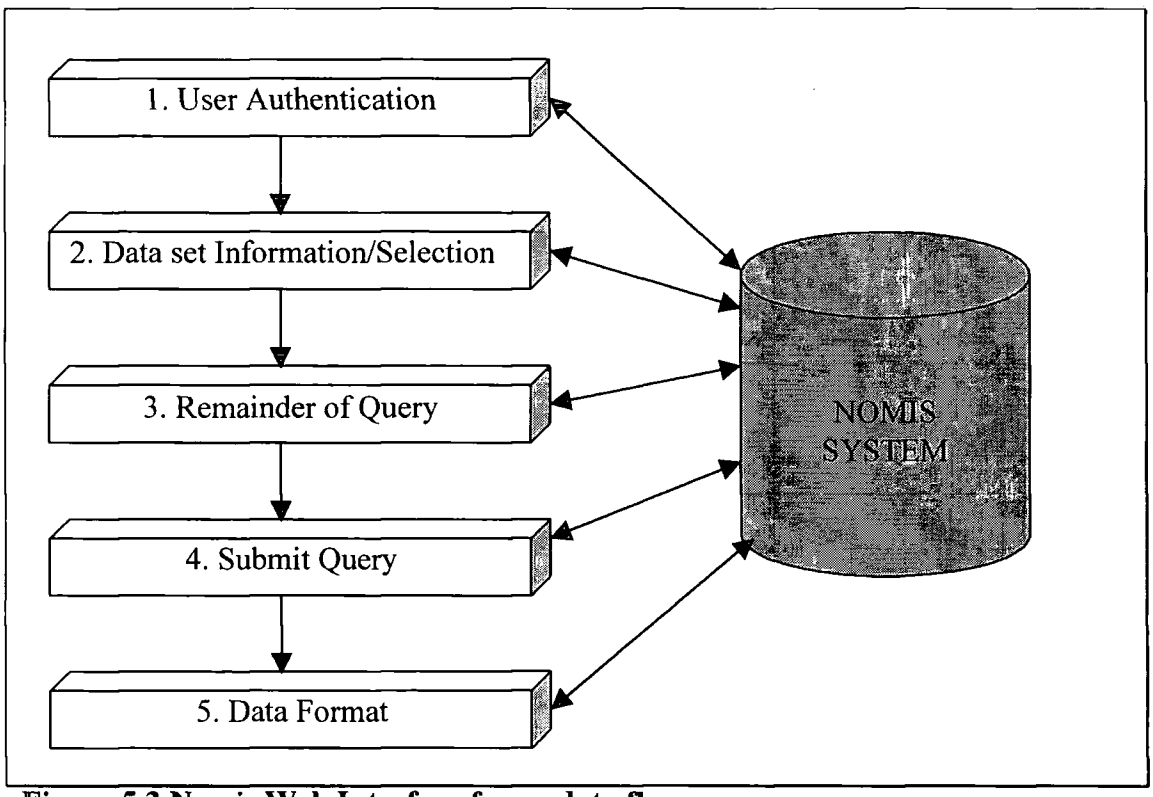


Figure 5.3 Nomis Web Interface forms data flow

The user input has been split into five stages:

1. User Authentication. For security purposes, a username and password are required before the system initialises. The username and password variables are passed to a server script that verifies that a given user exists and that the password is correct. This process is done via HyperText Transfer Protocol (HTTP). Without a user name the system has no means of determining what type of data the user has access to or their charging regime. The user specific details can be retrieved from the user details file used with the command interface.

2. Data set information/selection. A query cannot be initiated without a data set being selected as the remainder of the query depends on the data set that has been selected. General information regarding the different data sets should be available at this point. This information can be generated in HTML using the same data set details files used with the command interface.

3. Remainder of Query. Once the data set is "known" by the system a screen can be displayed which will allow the user to complete the query. The components required to complete the query (date, geography etc.) will be known by the system and can be "filled in" by the user by splitting them into sub-screens selectable as a series of buttons. Other features such as preview, save and new query could also be available as clickable buttons.

4. Submit query. Once the query has been built up it must be sent to the command system to be processed and the appropriate data returned. Before this is initiated the user should be informed as to the cost of the run and then the data can be retrieved.

5. Data format – WWW browsers can handle a number of data file formats so it would be useful to offer the user their data in several different formats for flexibility.

The WWW pages where broken up into six main sections. The "home" page and the five other user forms discussed above.

5.3.1. The WWW interface home page

The "home page" for the WWW interface would contain basic user information appropriate to the interface. This would include a simple welcome message, an E-mail link to the "Nomis Team" to make it simple for users to report any problems they might discover, a link to a "latest news" page and a link to the actual Web interface. Figure 5.4. shows the WWW interface home page at the time of writing. It is worth noting that the diagram also contains unimplemented links to data availability and charging structure pages.

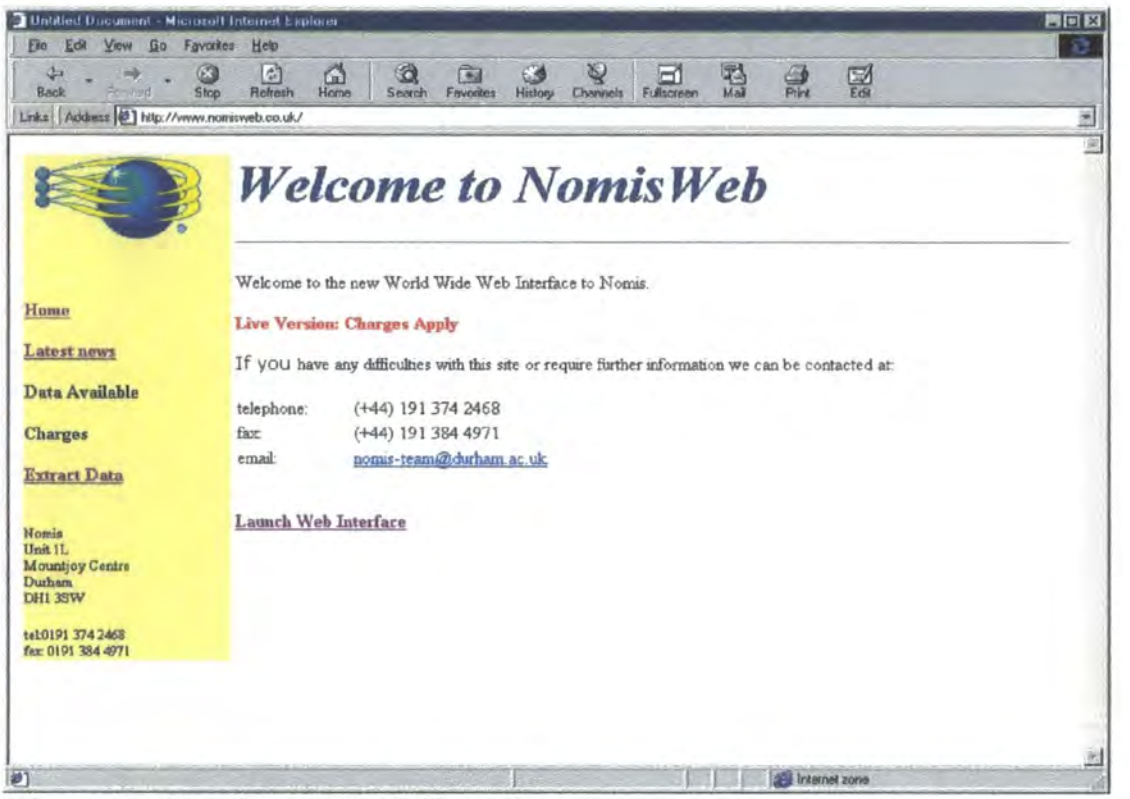


Figure 5.4. The "NomisWeb" Home Page

5.3.2. The user authentication form

Once a user clicks on the link to launch the Web Interface, the first of the HTML forms is displayed in order to authenticate the user. The user must enter a username and password in order to proceed any further.

This is shown in figure 5.5.

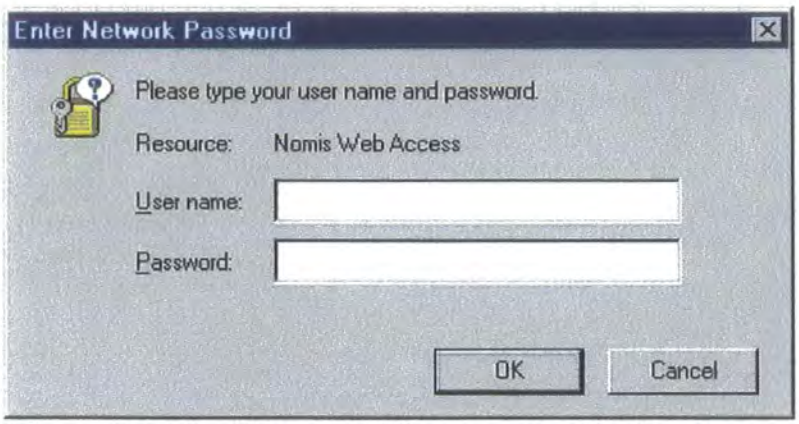


Figure 5.5. The "NomisWeb" user authentication form

5.3.3. The data set information/selection Form

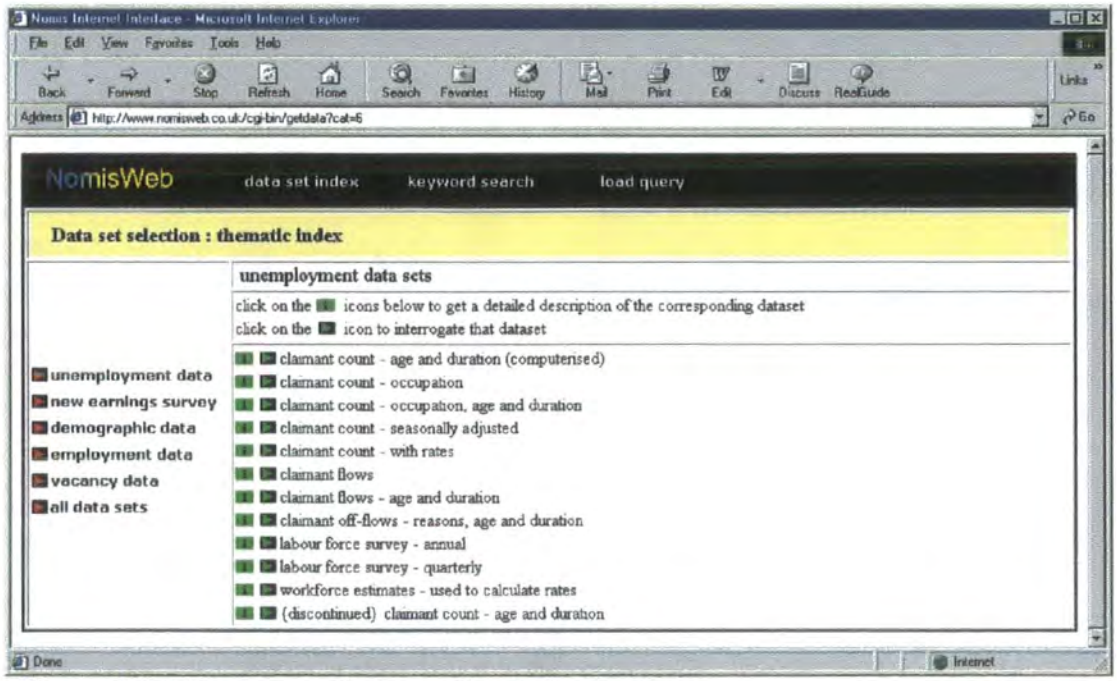
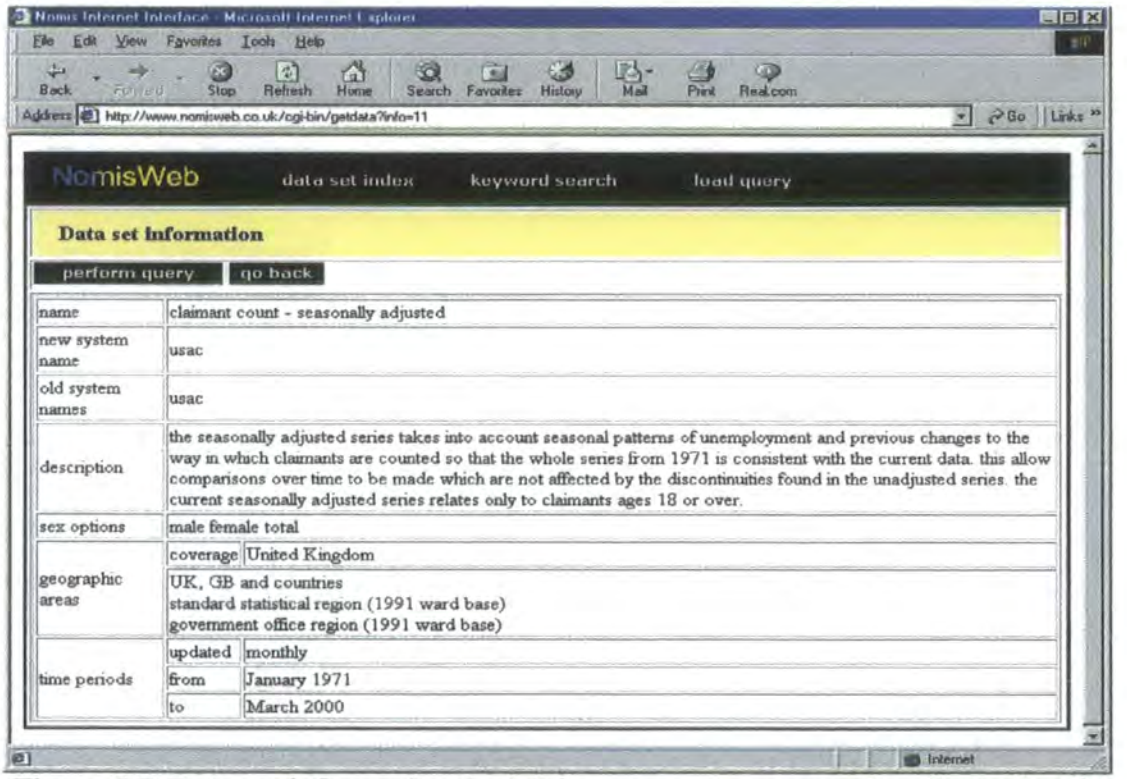


Figure 5.6. The data set information/selection form

Once a user has "logged in" successfully the system can look at the user details file and decide which data sets they are allowed to access and then present a "data set information/selection" form (figure 5.6.). Some users have access under licence to confidential data series. Others are subject to data charging for certain data series. Data set categories are presented down the left-hand side of the screen. Clicking on the red buttons will cause the system to display the appropriate data sets available on the right-hand side of the screen, by submitting the request to the Nomis system and returning the appropriate response as another HTML form. Users can also retrieve a previously saved query (described below) by clicking on the "load a saved query" button. By clicking the "i" icon next to a particular data set the system returns more information regarding the data set in question (figure 5.7.). This information is returned from the Nomis system using the same technique as for the data set categories discussed earlier. Clicking on the "arrow" button next to a data set initiates a user query and proceeds to the next stage of the user input, the query pages.



name	claimant count - seasonally adjusted	
new system name	usac	
old system names	usac	
description	the seasonally adjusted series takes into account seasonal patterns of unemployment and previous changes to the way in which claimants are counted so that the whole series from 1971 is consistent with the current data. this allow comparisons over time to be made which are not affected by the discontinuities found in the unadjusted series. the current seasonally adjusted series relates only to claimants ages 18 or over.	
sex options	male female total	
geographic areas	coverage	United Kingdom
	UK, GB and countries standard statistical region (1991 ward base) government office region (1991 ward base)	
time periods	updated	monthly
	from	January 1971
	to	March 2000

Figure 5.7. Data set information for 'claimant count – seasonally adjusted' series

5.4. The remainder of query

The remainder of query has been grouped together into one section because although there are several different pages used, they all follow a standard pattern which helps to speed up the user learning curve. There is an example of a query page in figure 5.8.

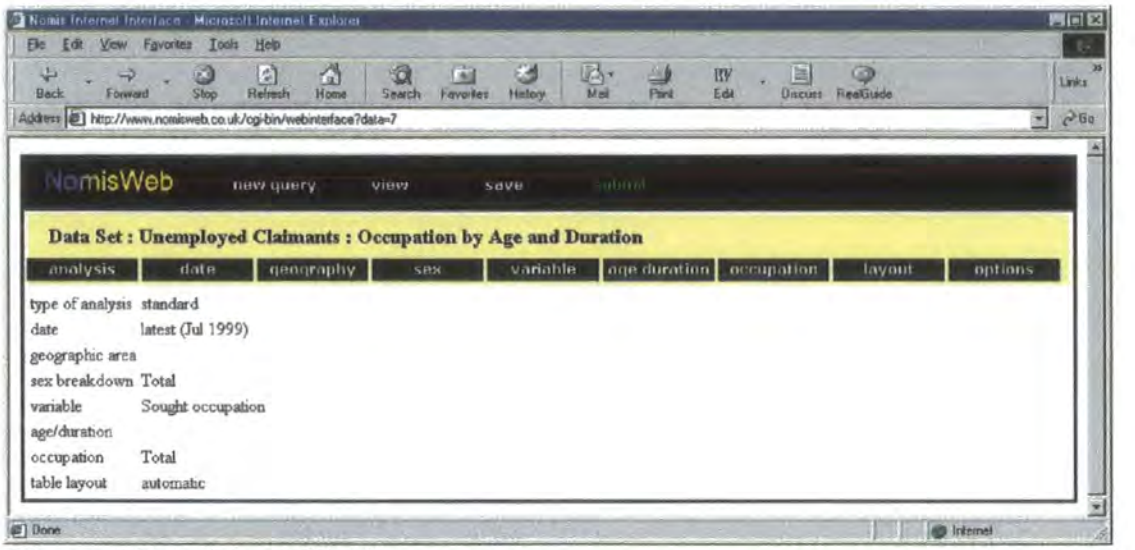


Figure 5.8. A "NomisWeb" query page

The top menu is split into two sections.

- The query menu (black section) – general buttons that can apply to any data set
- The query component menu (beige section) – used to select various breakdowns available for a data set.

The options available in the query menu are consistent regardless of which data set has been selected. To begin a completely new query users click on the "new query" button. Clicking on the "view" button provides details of the current query. The "save" and "load" query buttons respectively save the current query for re-use, and allow a saved query to be re-loaded. Once a query is complete clicking "submit" sends the query for processing.

5.4.1. Query component menu.

The general appearance of this menu will be similar for all data sets, but the buttons available will vary depending on the data set selected. In the example shown the data set is "Unemployment Claimants : Occupation Age and Duration" and the breakdowns available are date, geography, sex, variable, age/duration, occupation and layout. All of these breakdowns have corresponding buttons in the query component menu, which once clicked will take the user through one or two more structured selection screens depending on the component. The "sex" button displays the gender options screen (figure 5.9.), and boxes can be clicked to select the gender breakdowns that are required. All gender breakdowns are obtained by clicking the text "select all genders on page", and are removed for re-selection using "clear anything selected on page". This principle applies to most of the simple query component selection screens eg date, variable, flow, table. However, the more complex query components such as geography have further menu page breakdowns prior to the selection screen. It is logical to split such selection processes up as there are too many geographical areas available to represent on a single page.

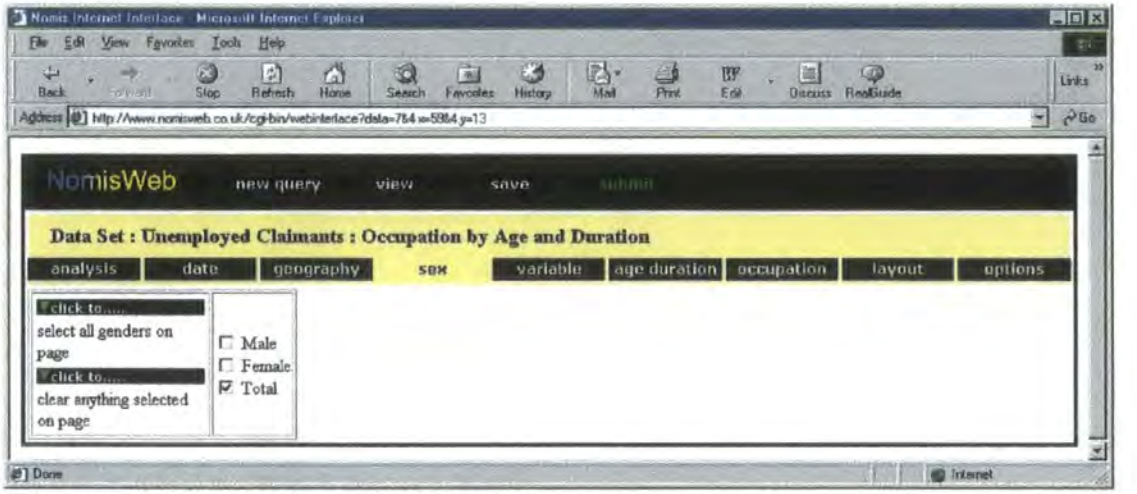


Figure 5.9. The gender options screen

5.4.2. Geography selection

By clicking on the "geography button" a series of submenu options appear as shown in figure 5.10.

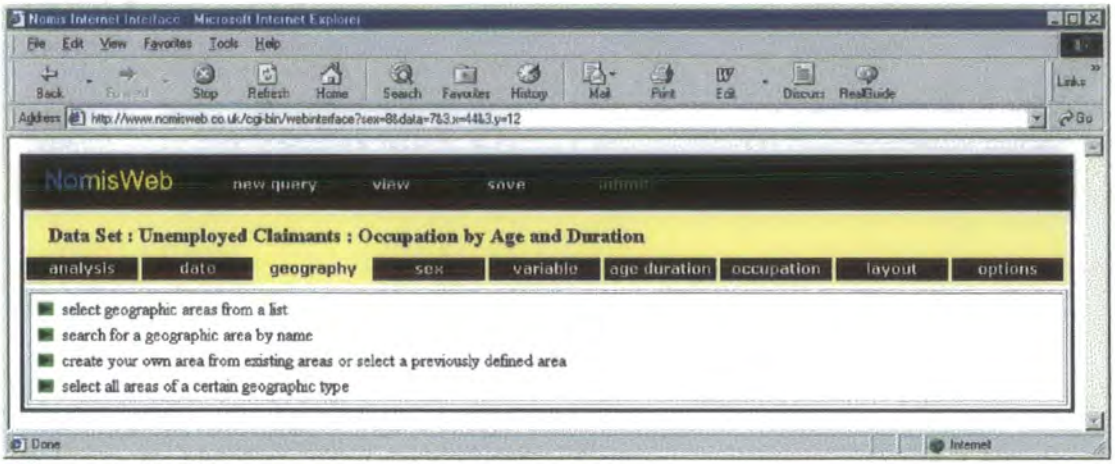


Figure 5.10. The geography submenu screen

Four options are provided. These are; selecting a number of geographies from a list; searching for all geographies matching a name; creating aggregate geographies; selecting all available geographies of a particular type – such as counties. Clicking on the first option produces the detail in figure 5.11.

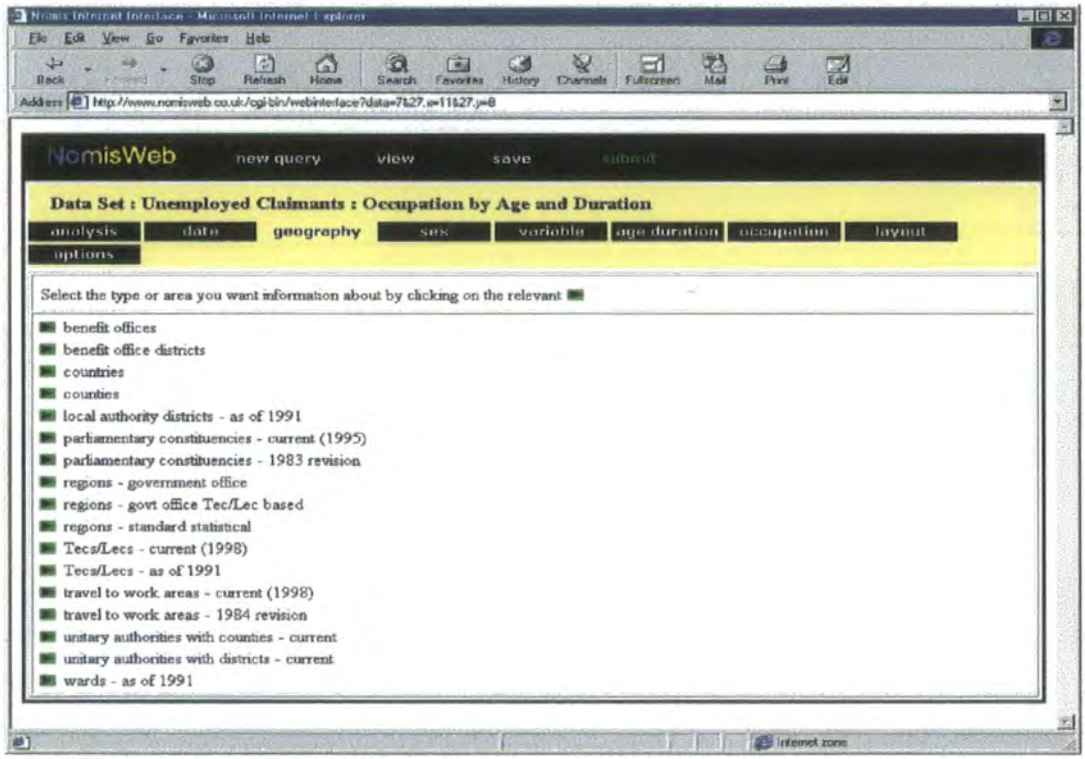


Figure 5.11. Geographical selection screen

The next stage is to decide which area type is required and click on the appropriate button. Figure 5.12. shows the screen after selection of "travel to work areas – current (1998)". There are 308 of these areas which cannot be represented on a single page. Therefore, each page is broken down by region (12 sub-pages). If the requirement is for the ttwa98 areas in the Northern region as opposed to the South East, then the drag down menu which follows the text "travel to work areas - current (1998) within the **South East** region - standard statistical" can be used. This is achieved by highlighting "Northern" (see figure 5.12. below) and clicking select.

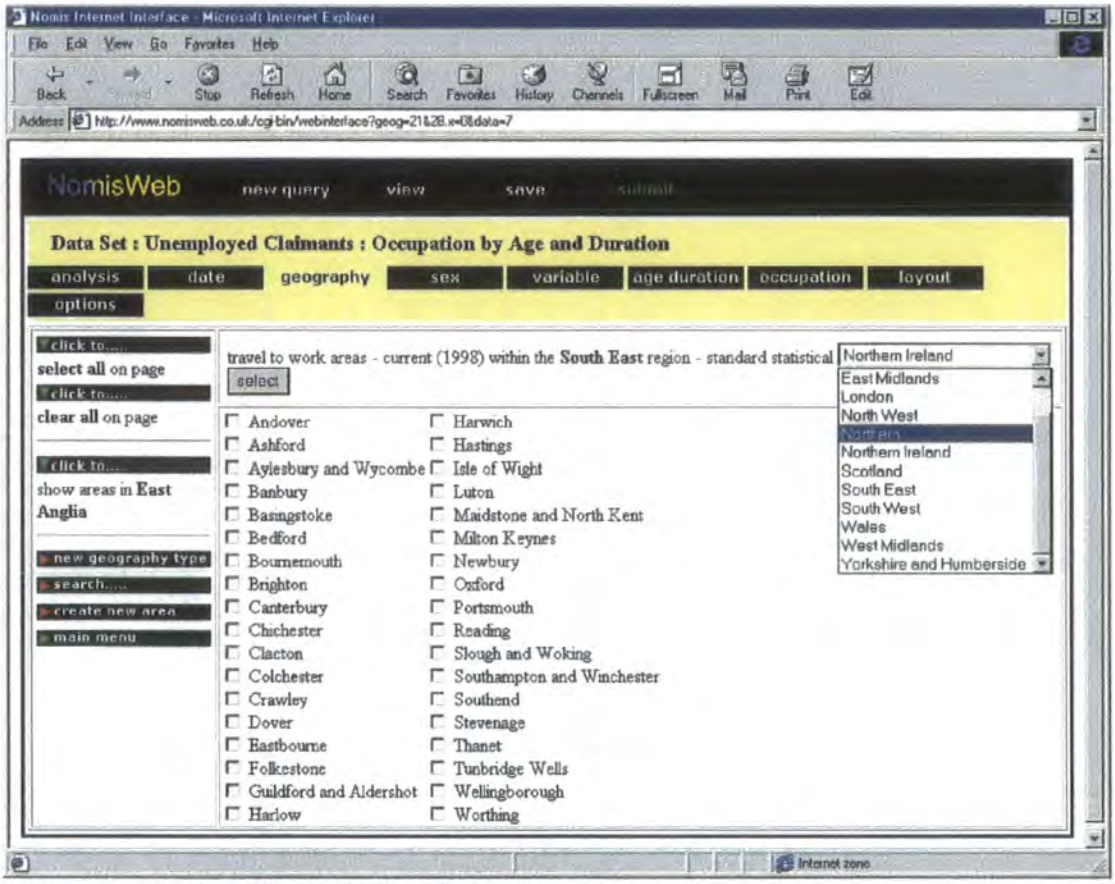


Figure 5.12. ttwa98 within South East region

An individual area can be selected using the case insensitive "search for a geographic area by name" option. For example, identifying areas matching the name "Durham" is achieved by typing in the name "Durham" and clicking on search to produce a screen shown as figure 5.13. below. It is then a simple case of clicking on the appropriate match of "Durham" that is needed.

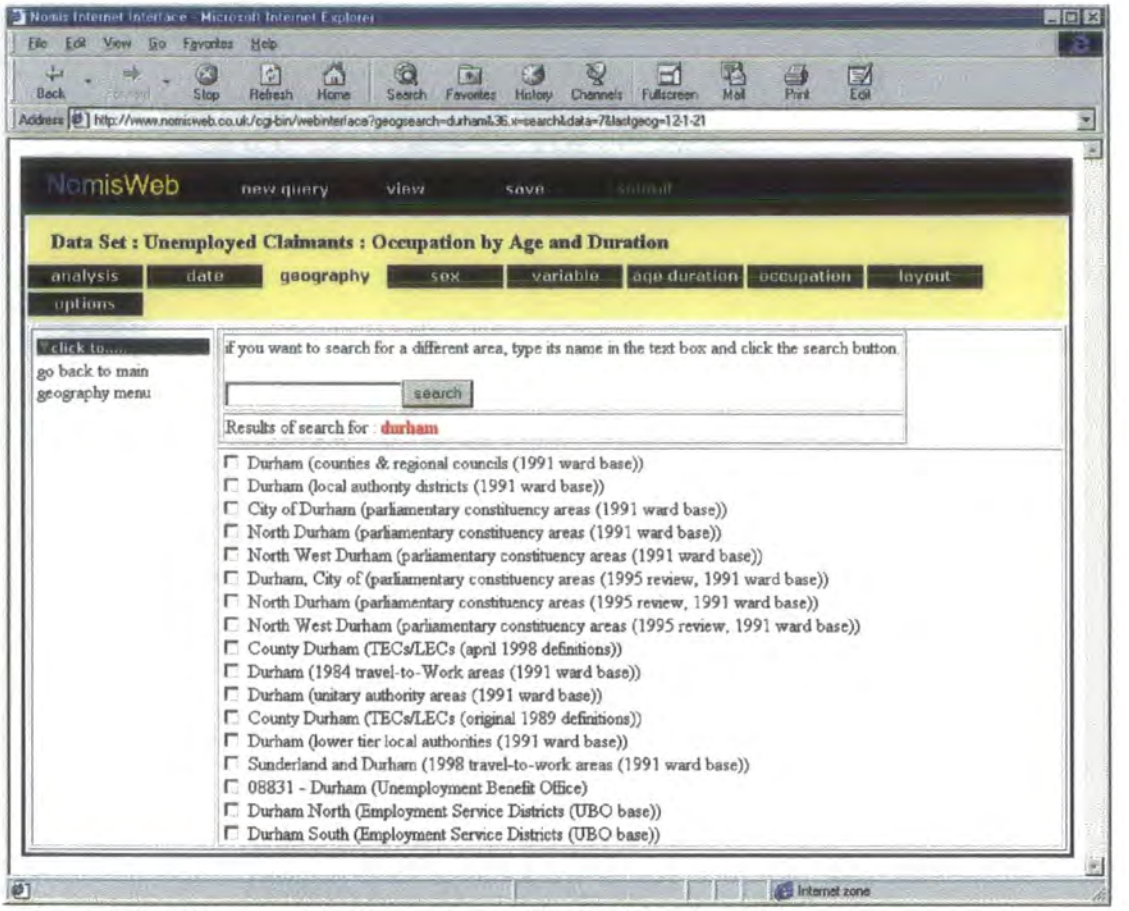


Figure 5.13. Geographical search results for "Durham"

The option "create your own area from existing areas or select a previously defined area" is similar in principle to the standard geographical selection process. The difference is that the areas selected are added together when the data is retrieved. The area definition can be saved for future use. Figures 5.14. to 5.17. show the creation process of a new area "Real North West Gor" which aggregates the data for "North West" and "Merseyside".

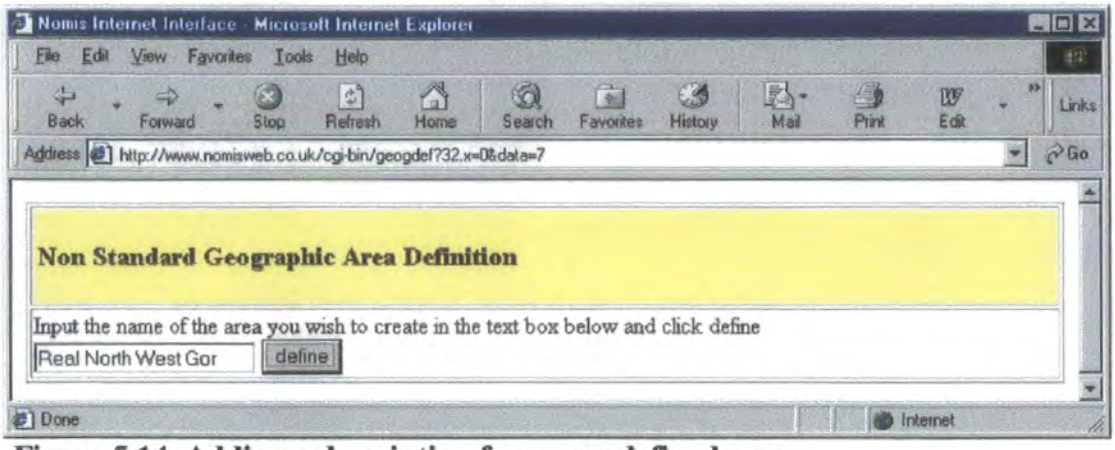


Figure 5.14. Adding a description for a user defined area

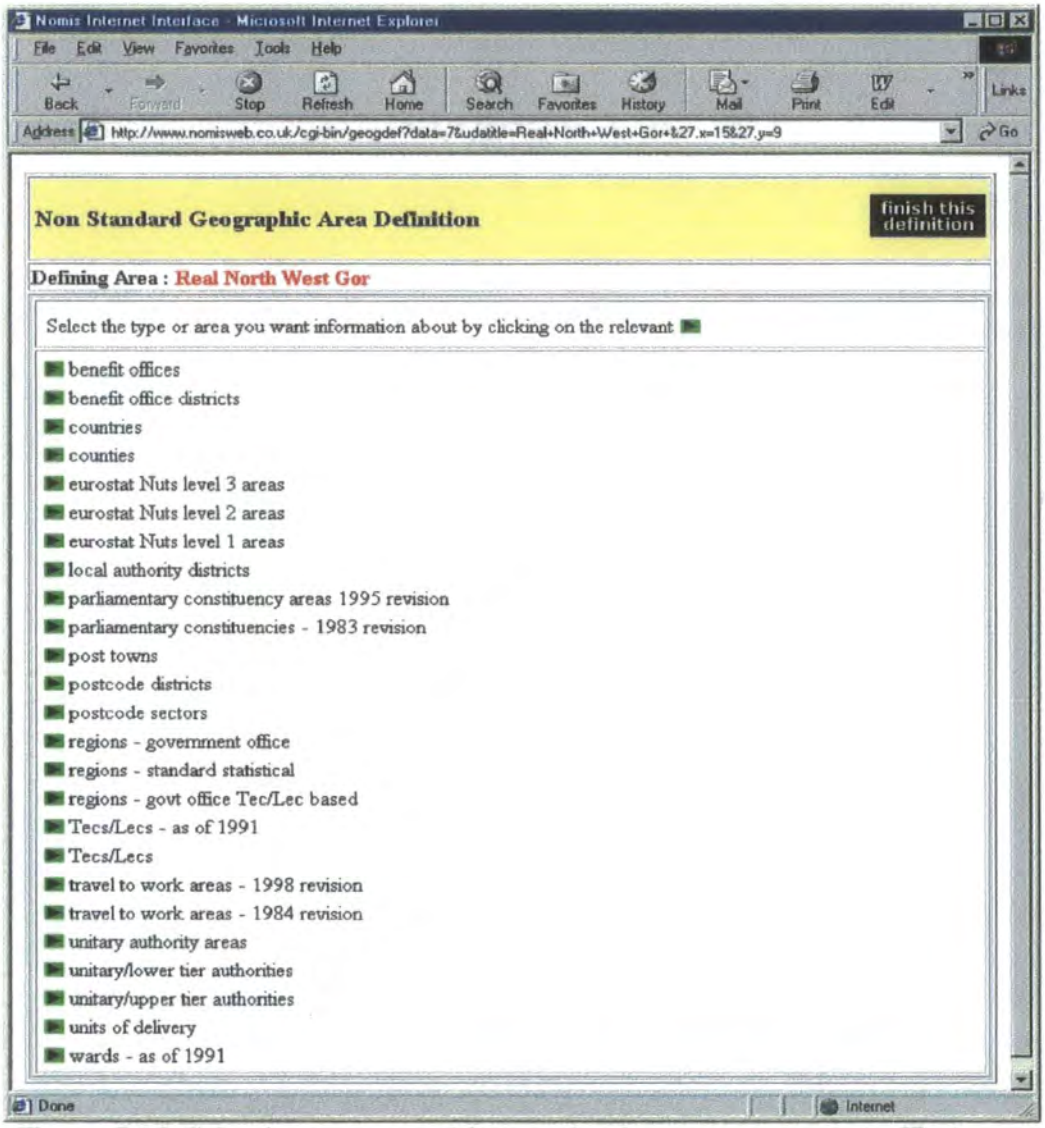


Figure 5.15. Selecting a geographic type (regions – government office)

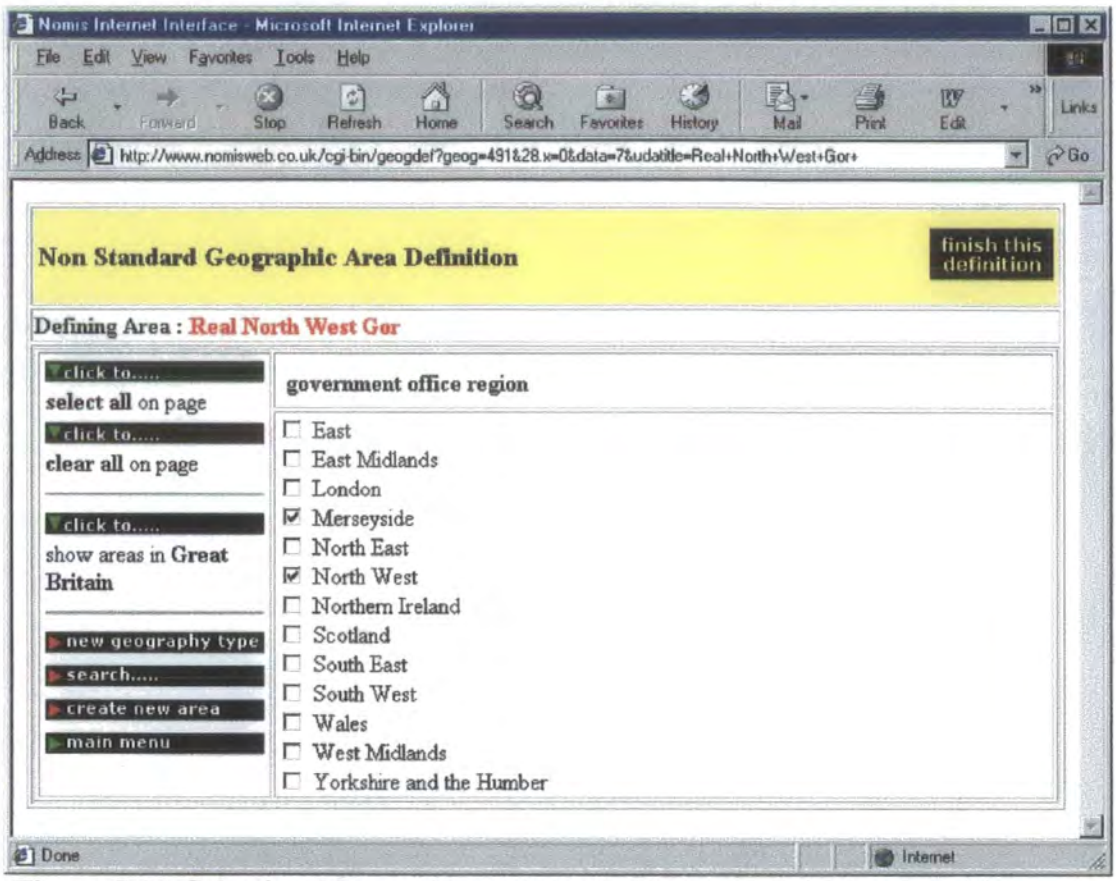


Figure 5.16. Selecting Merseyside and North West and finishing definition

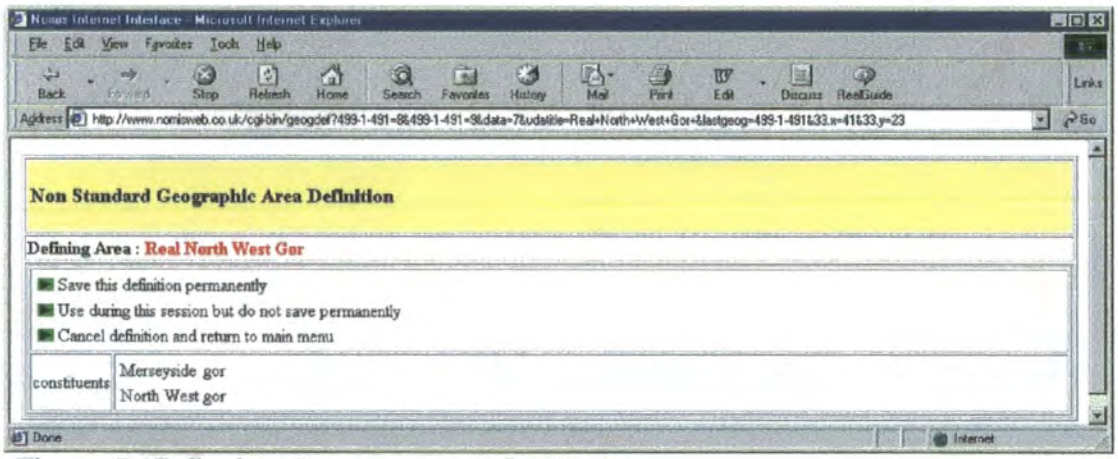


Figure 5.17. Saving the new user defined area

Figure 5.18. shows how the selection of all areas of a certain geographical type is achieved (i.e. all regions or all counties).

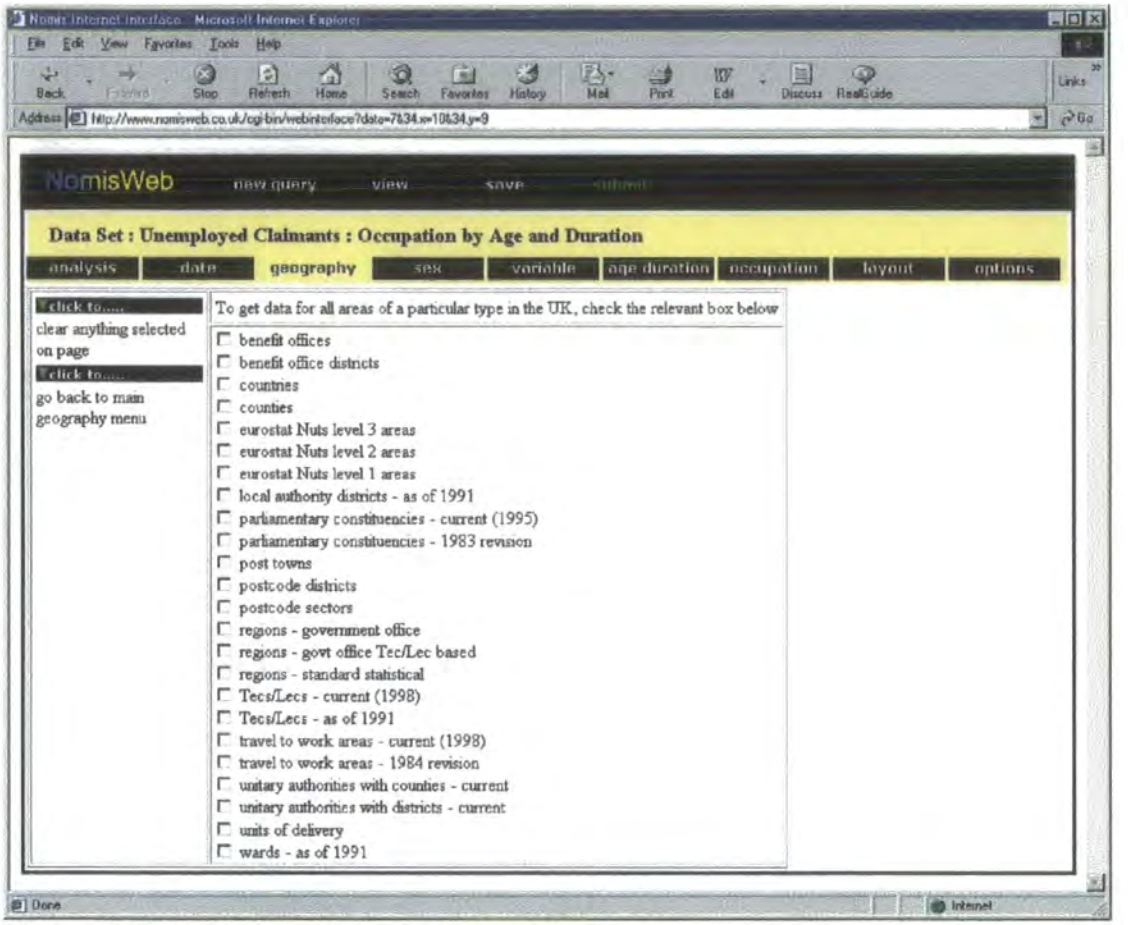


Figure 5.18. Selecting all areas of a certain type

The process of geographical selection is 'additive', in that a series of selections from different geographies will be added to a list that will be used for output. There is an option to clear all selections and to start again.

5.5. Finalising the query

Once selection is complete the "submit" button is clicked. The processes needed in order to complete the query can be separated into three sections.

5.5.1. Query checking

The interface must determine that all of the relevant components have been "filled in" for submission to the Nomis system. If any components are still required then a relevant warning message will be provided. The system will never continue to the next stage until all of the necessary components have been selected. An example of an incomplete query is show in figure 5.19. where a geographic area has not been selected.

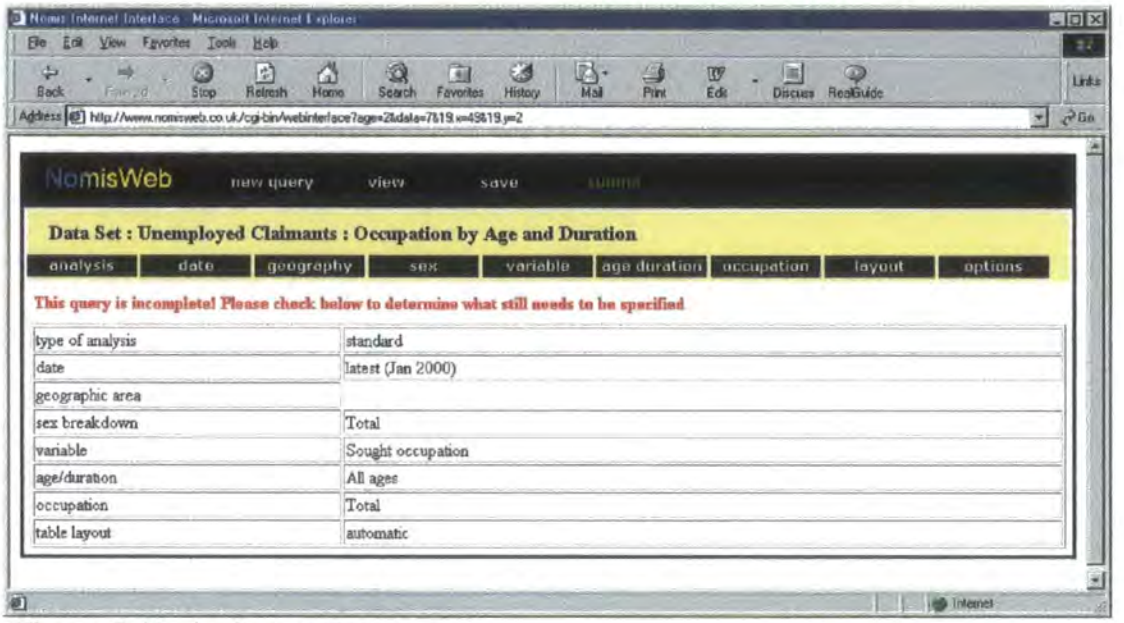


Figure 5.19. An incomplete query

5.5.2. Query processing

Once the system has established that the query is complete, it calculates by means of a cell based charging regime, how much the query will cost. The Web has no concept of elapsed time in user sessions, and time historically has been the mechanism by which users have paid for access to Nomis. Without time the only option has been to revert to a 'data value' approach, which is the least unsatisfactory charging mechanism that could be identified. Nomis is not suited to flat session charging since users may extract significant volumes of data in a single query, or may simply want a few cells of data. It would not be equivocal either in terms of system load, or data owner relations, to provide a mechanism where an entire time series could be extracted for a flat cost. The information is presented as a Form, which can be accepted or declined.

An example of this is shown in figure 5.20. below. Cost is provided as a price (for pay-as-you-go users) or in charging units. Charging units are pre-purchased amounts of usage, working on the basis that the higher value of yearly subscription taken by a user the lower the price of a charging unit.

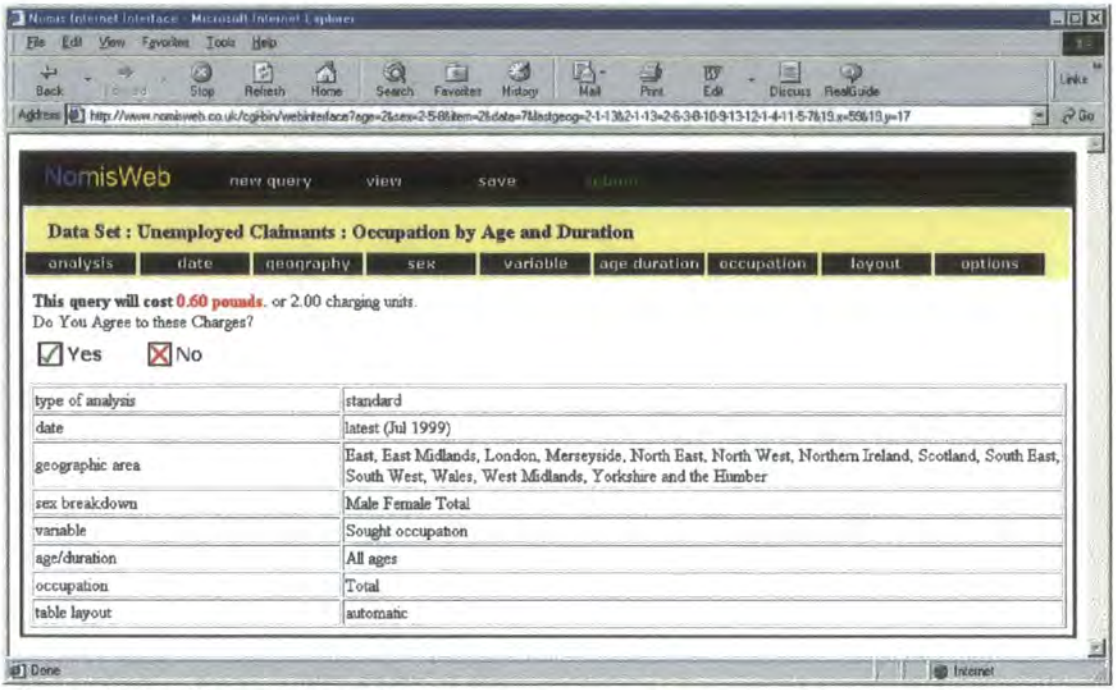


Figure 5.20. Query submission form

On clicking the "Yes" button the real processing begins by means of a program sitting on the Web server called "gocgi".

5.5.2.1. The "gocgi" program

The "gocgi" program, like all of the other programs used to generate the NomisWeb pages resides in the cgi-bin directory on the Web Server. Felton (1997) describes the cgi-bin directory as a special directory associated with a Web Server. Programs placed in this directory are used to provide CGI capabilities which extend well past plain HTML.

The first stage of this program converts all of the components that have been selected into a Nomis command file ready for execution. The Nomis program is then run in the background and the newly generated command file is processed. The resulting output is stored in a temporary area in three different formats. Users are presented with

another Form to choose which format is required. This form is shown as figure 5.21. below.

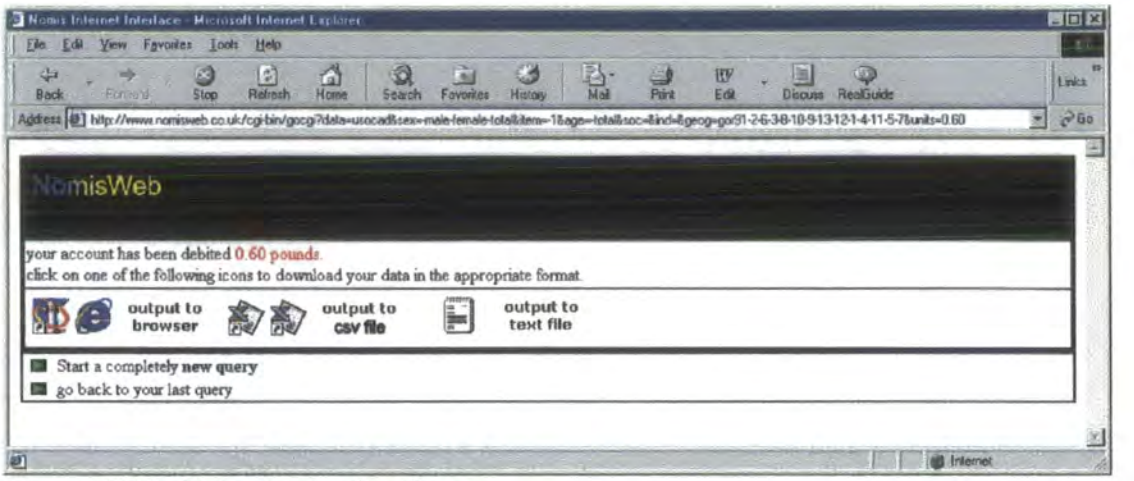
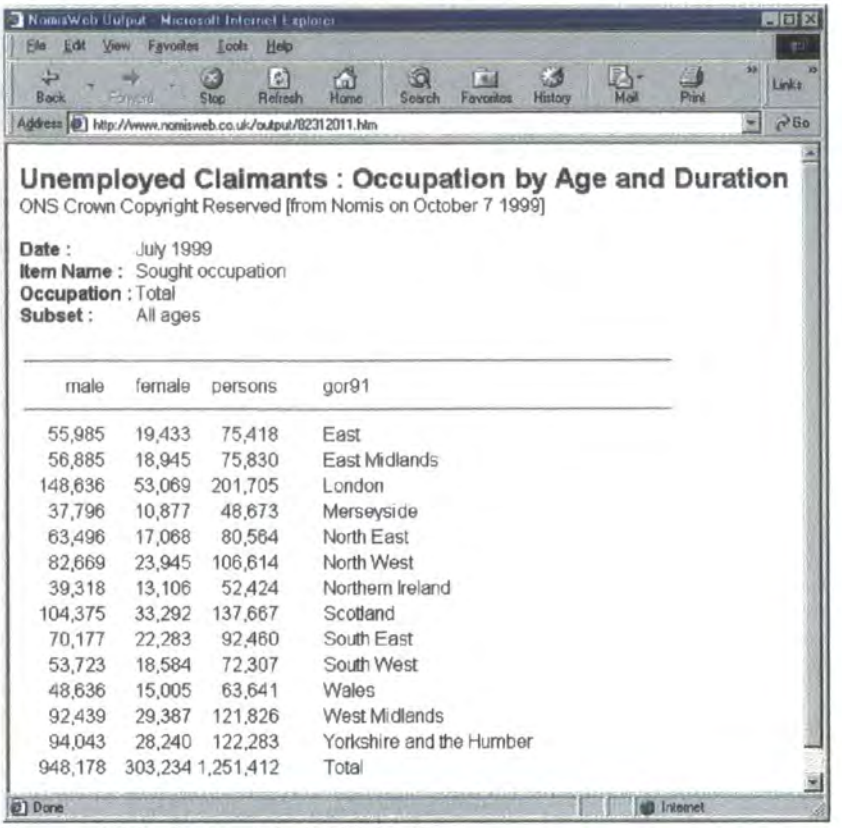


Figure 5.21. Output selection form

5.5.3. Data delivery

There are three choices of output type

1. Output to browser – HTML format displayed on the browser for printing (figure 5.22.)
2. Output to CSV (Comma Separated Format) – used to import data into a spreadsheet (figure 5.23.)
3. Output to text file – for Import to a Word Processor (figure 5.24.)



NomisWeb Output - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Links

Address http://www.nomisweb.co.uk/output/82312011.htm Go

Unemployed Claimants : Occupation by Age and Duration

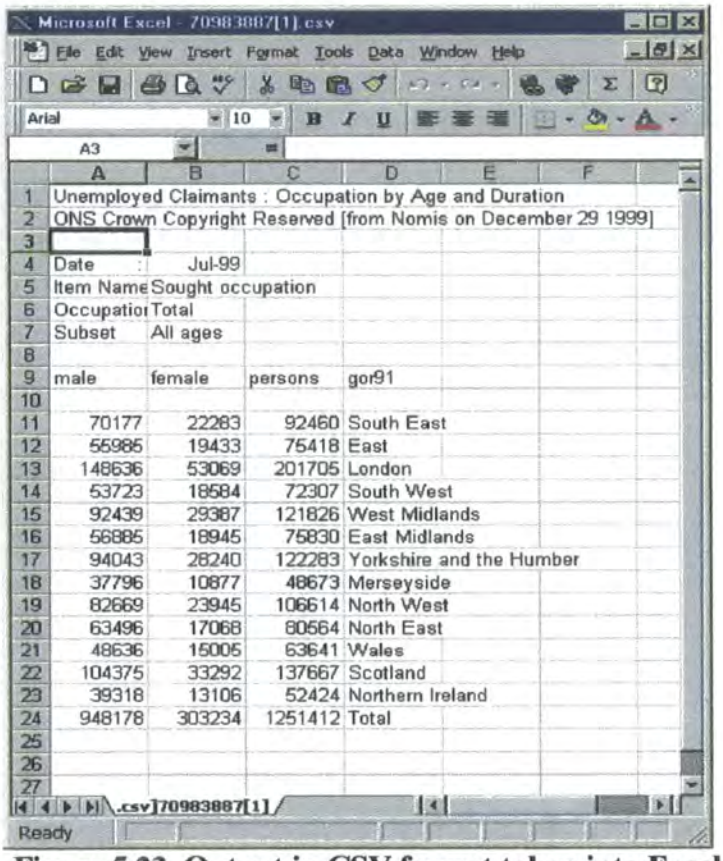
ONS Crown Copyright Reserved [from Nomis on October 7 1999]

Date : July 1999
Item Name : Sought occupation
Occupation : Total
Subset : All ages

male	female	persons	gor91
55,985	19,433	75,418	East
56,885	18,945	75,830	East Midlands
148,636	53,069	201,705	London
37,796	10,877	48,673	Merseyside
63,496	17,068	80,564	North East
82,669	23,945	106,614	North West
39,318	13,106	52,424	Northern Ireland
104,375	33,292	137,667	Scotland
70,177	22,283	92,460	South East
53,723	18,584	72,307	South West
48,636	15,005	63,641	Wales
92,439	29,387	121,826	West Midlands
94,043	28,240	122,283	Yorkshire and the Humber
948,178	303,234	1,251,412	Total

Done Internet

Figure 5.22. Output in HTML format



Microsoft Excel - 70983887[1].csv

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U

A3 =

	A	B	C	D	E	F
1	Unemployed Claimants : Occupation by Age and Duration					
2	ONS Crown Copyright Reserved [from Nomis on December 29 1999]					
3						
4	Date :	Jul-99				
5	Item Name	Sought occupation				
6	Occupation	Total				
7	Subset	All ages				
8						
9	male	female	persons	gor91		
10						
11	70177	22283	92460	South East		
12	55985	19433	75418	East		
13	148636	53069	201705	London		
14	53723	18584	72307	South West		
15	92439	29387	121826	West Midlands		
16	56885	18945	75830	East Midlands		
17	94043	28240	122283	Yorkshire and the Humber		
18	37796	10877	48673	Merseyside		
19	82669	23945	106614	North West		
20	63496	17068	80564	North East		
21	48636	15005	63641	Wales		
22	104375	33292	137667	Scotland		
23	39318	13106	52424	Northern Ireland		
24	948178	303234	1251412	Total		
25						
26						
27						

.csv]70983887[1]/

Ready

Figure 5.23. Output in CSV format taken into Excel

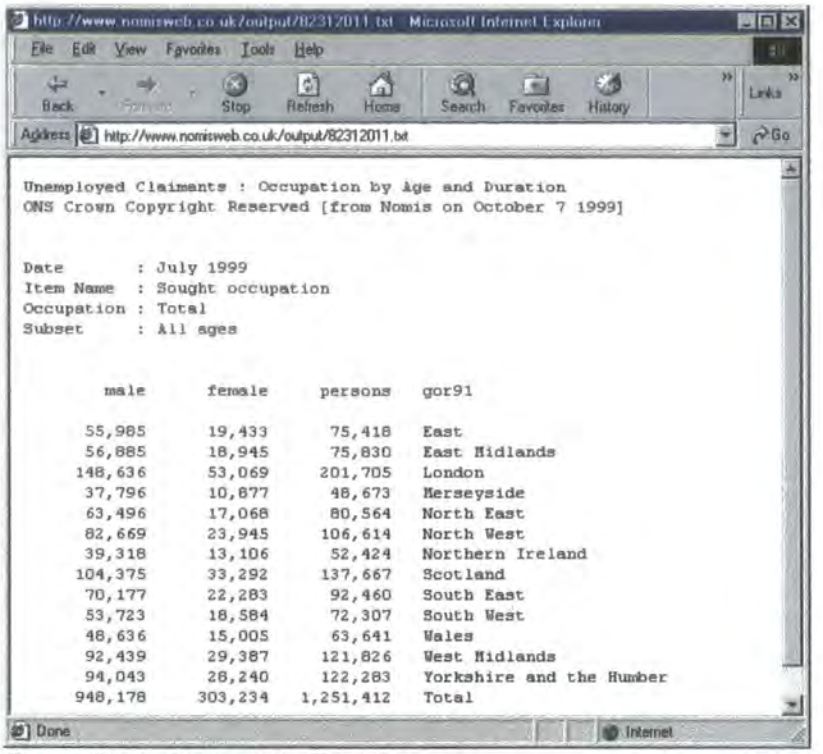


Figure 5.24. Output in Text Format

5.6. Emergent Web Technologies

NomisWeb, like a lot of Web sites, allows users to interact with programs (or scripts) via the use of HTML and the Common Gateway Interface (CGI). These requests (and server responses) are formatted using the rules of the HyperText Transfer Protocol (HTTP) standard. However, HTTP is "stateless" meaning that each time a Web page is loaded the user is effectively disconnected from the server, therefore, the server has no knowledge of what the user is doing. Even after logging on to a site, the username and password is passed across to the server for each new page accessed to verify that the user has permission to access that new page. In other words, there is no direct interaction between the 'New Nomis' Web Interface and the server.

New Web technologies are becoming available which may allow more intelligent client/server interaction. This could provide users with better feedback from a Web Interface (warnings/error messages) and would allow more complicated component selections instantaneously as the Interface would be in constant communication with the server, and therefore also with the actual database.

The choice of using HTML and forms made it possible for all Nomis users with Internet access to retrieve data via the Web, and it was capable of being designed and developed within the resources and time-scales available to us. Once all of the data have been transferred onto 'New Nomis' (by the end of 2000) resources will be released to develop an enhanced interface using Web technologies available such as CORBA, SGML, XML and Java. Users with basic IT resources will still be able to access the original Web Interface, but as PC technology increases even users with moderate IT resources will be able to access such a new Interface.

5.6.1. CORBA

Morgan (1997) describes the Common Object Request Broker Architecture (CORBA) as a specification that defines how distributed objects can interoperate. The specification is controlled by the Object Management Group (OMG), a group of over 700 companies that collectively define open standards for object computing.

CORBA objects can be written in any language supported by a CORBA software manufacturer such as JAVA, C or C++. The objects can exist on various platforms including UNIX (which is the 'New Nomis' system platform). Language independence is made possible by constructing interfaces to the objects using the Interface Description Language (IDL). CORBA objects communicate with each other using an Object Request Broker (ORB).

A new NomisWeb interface could be developed using CORBA. The CORBA objects could be written in C++. For example: a gender selection screen. With the current NomisWeb interface when a user clicks on the "sex" button, the system (in the background) sends a text list of the available sex breakdowns to the browser and presents them as a series of 'checkboxes'. If a CORBA object was written to represent the different sex breakdowns then this could be sent to the browser. As objects contain information regarding themselves, validation checks can be performed at each stage of a query. Objects can also communicate with each other so checking query component interdependencies (which was previously unavailable) may also be possible.

5.6.2. SGML

Bosak (1997) notes that Standard Generalized Markup Language (SGML) allows documents to describe their own grammar. HTML itself is based on SGML but it uses a small frozen set of tags that conform to a single SGML specification. The freezing of these tags makes it simple to develop applications but result in the following limitations:

1. Users cannot specify their own tags or attributes used to qualify their data.
2. No support exists for deep structures needed to represent database schemas or object-oriented hierarchies.
3. No support exists for data validation.

Generic SGML applications have none of the above limitations and makes it possible to define personal document formats, handle large and complex documents and manage large information repositories. The full SGML contains many features not required for Web applications and as a result is over complicated. However, a more straightforward, simplified set of specifications has been set up by the World Wide Web Consortium (W3C). This specification set is known as the Extensible Markup Language (XML).

5.6.3. XML

There are three main differences between XML and HTML:

1. New tags and attribute names can be defined.
2. There is no limit to nesting level of document structures.
3. XML documents can contain a description of its grammar for use in applications that must perform structural validation.

NomisWeb currently uses HTML, which is limited by its frozen tags. A new interface could be developed using XML with new tags defined for individual query components. Because XML documents can contain descriptions of grammar, this

would also allow query validation checking. However, XML does not use objects so would be an intermediate step between the current NomisWeb and using CORBA.

5.7. Conclusion

A WWW interface to 'New Nomis' has been implemented and was officially released to users in April 1999. Whilst the initial release of the interface is simplistic, it does not offer the full functionality of the command interface. The next stage of development will be to do a SWOT analysis of the latest Web technologies and develop a new Web interface that does offer the full functionality of the command interface. Chapter six illustrates ways in which the success of 'New Nomis' were evaluated.

Chapter 6 Evaluating the New System

6.1. Introduction

The 'New Nomis' system was officially released to users in January 1999 via the command interface. Thereafter the Web interface was designed, tested, and released to all users in April 1999. In January 1999 the only data made available to users was the current unemployment series. Because of the time scale involved in transferring all of the data sets to the new system, 'Old Nomis' and 'New Nomis' had to run in parallel. To help encourage more use three new data sets were made available only on the new system – Unemployment Stocks by Occupation Age & Duration (USOCAD), Unemployment Destinations by Age and Duration (UDAD) and Labour Force Survey - Annual (LFSA) with the promise of the New Earnings Survey (NES) once ONS decided on a charging policy. By Christmas 1999 the Annual Employment Survey (AES) and the vacancy series were added to 'New Nomis'.

On 16th February 2000 the "Press Release" unemployment and vacancy data was only made available to users on the 'New Nomis' system. By that stage established users had been provided with nine months of access to the new system, including some free resource for experimentation. New users had been trained primarily on 'New Nomis' during Introductory Training Courses since late Summer of 1999. New documentation had been provided, and the period of system transition was, therefore, nearly finished.

Adding new data sets and releasing the latest data only on the 'New Nomis' system resulted in increased usage levels. Monitoring these usage levels formed part of the overall evaluation process of the 'New Nomis' system. This chapter will discuss the performance and further ways of evaluating 'New Nomis' including reviewing the user interface and beta testing.

6.2. User Interface

The first part of the evaluation process was to examine the 'New Nomis' user interface and establish if the new system met with all of the design criteria :

- Standardise commands to achieve consistency

There is only one command type used to specify age/duration definitions regardless of the data set selected this was not the case with 'Old Nomis'. The ***output*** and ***format*** commands (used to manipulate table output) can be used in any query for any table format, this was also previously unavailable.

- Simplify the commands where possible

With the introduction of the ***make*** command it is possible to create aggregates for geographic areas, occupations, industries, age/durations, sizebands and cells using the same command structure. 'Old Nomis' achieved this using an assortment of commands:

user	- used to create a single aggregate area
add	- used to aggregate an occupation/industry
subset	- used to aggregate age/duration definitions
&sbselect	- used to aggregate sizebands (Annual Employment Survey –Workplace Analysis)
lbsratio	- used to aggregate cells (Local Base Statistics 1991)

If users wanted more than one area defining they would have to create a file which contained each individual ***user*** command. 'New Nomis' allows unlimited aggregations each definition separated with the ***and*** command.

- Maintain reasonable backward compatibility with the previous Nomis system. It was accepted that with a large 'legacy' user base the cost of re-training should be kept to a minimum.

The 'New Nomis' command syntax although not exactly the same as 'Old Nomis' does still have some similarities. The ***data*** command always initialises a query and commands such as ***sex***, ***item***, ***date*** follow a similar syntax as before. 'Old Nomis' command files, which established users have saved, will need changes to be successful on 'New Nomis'. Various site visits and calls from the 'Help Desk' team ensured that this was as smooth a transition as possible.

- Provide a 'menu-style' interface

NomisWeb was released to users in April 1999 and (as indicated later) by April 2000 was the most popular retrieval platform for users.

- Provide increased functionality of commands

With the introduction of selections such as ***all*** users have the ability to select all values of a particular query component such as 'all items'. This can be done without any prior knowledge of how many items exist for a particular data set. Use of the selection ***latest*** in conjunction with the ***date*** command will allow the system to automatically select the latest available data for a particular data set. Users can then save their query and then reload it in subsequent release periods and always receive the latest figures without having to edit their command files.

- Provide a context sensitive help system which could assist users at any stage during a query

The ***check*** facility is available at any time during query to allow users to see what has been selected and what is still required to complete a particular query.

6.3. Beta Testing

As with any major system development there was a period of 'beta testing' prior to the official release of 'New Nomis'. From October 1998, the new system was made available for testing to a selected group of users. This included 60 ONS users, 12

Employment Information Units (EIUs) and 17 other key users from the following sectors: Training and Enterprise Councils (Tecs), City and County Councils, Commercial users, Academic users. All of the initial 'beta test' users were sent documentation which contained 'test account' details (for logging into the system) and a basic getting started guide. This guide was split into three sections: the first section introduced the new command interface; the second section referenced the data sets available and the third section introduced NomisWeb.

The initial response (although a low overall percentage) was generally very positive. Most of the useful comments and suggestions came from the 17 key users. Generally people found the Web interface easy to use – giving reasons such as the reduced need for referral to manuals. They also liked the fact that because the Web charging structure is based on the data cells downloaded, they could experiment with the Web system before having to pay for the data. One user suggested that the Web interface would be used for small specific queries, which in the past would have taken unacceptable time to work out the command sequence.

The command interface generated some minor problems. One user commented on the "inflexibility of the command interface in terms of reviewing the structure of a query". It was pointed out that using the ***check*** or ***?*** command would list the current query or ***review*** would list any previously typed commands. Some users were also unaware that the ***make*** command could be used to combine different occupations, industries, age/durations etc, as well as different geographies.

A lot of these problems were a result of the levels of documentation that was available at the time of testing. The documentation was improved as a result of the beta testing feedback and this reduced the number of future queries received regarding command problems.

The 'beta testing' of the 'New System' is a continuous process. Almost every month twelve users attend 'Nomis' training courses helping them to learn how to use the 'New System' but at the same time helping the Nomis team to test it. Each user is given eight hours of free time to experiment with the 'New System' after attending the course. Giving users their own time to experiment with the system and allowing

them to report back (either by phone or e-mail) is an important way of identifying ways in which we can improve the system as well as occasionally 'ironing out' bugs in the system.

Site visits have also taken place by various Nomis team members in order to assist 'Old Nomis' users in the transition over to the 'New Nomis' system. One of the objectives of the visit was to assist users in the transfer of 'Old Nomis' command files to the 'New Nomis' system. These visits also helped identify ways of improving the system. In the first few months of 2000 several 'focus group' visits were arranged to key users at regional Employment Information Units (EIU's). Comments from these visits are discussed below:

'New Nomis' Command System
Strengths
Most had used the 'New System' before visits and found transition from 'Old Nomis' to 'New Nomis' relatively easy.
Ability to combine different Occupations/Industries using ***make*** command.
Age and Duration subsets a lot easier with ***make*** command.
Pre defined dates e.g. ***date latest*** and ***date prerelease*** in conjunction with ***save/read*** mean no 'file manipulation' necessary every month prior to "Press Release".
Geographical listing and searching such as ***list lad in region*** and ***search geog <string>***
Ability to review query using ***check*** command.
Weaknesses
Automatic layout didn't always generate to required table. Manual layout selection sometimes needed trial and error before the required format was established.
No 'Historical' data were available.
The data tables generated were occasionally more than 132 columns wide, which caused 'data-wrapping' on screen or in the output file.
The lack of some 'Old Nomis' user specific features such as ***wardprint*** and ***&title***

Figure 6.1. 'New Nomis' command interface strengths and weaknesses

'NomisWeb' Interface
Strengths
Liked simplicity of Web interface.
Geography selection much simpler using geographical search.
Outputting data directly into spreadsheets via Comma Separated Value (CSV) format.
NomisWeb preferred interface for ad-hoc queries.
Saving queries quick and simple
Weaknesses
Certain command system options not available on the Web such as ***row*** and ***sort***
Changing data sets on NomisWeb results in component selections being lost – selected geographies, items, etc are not passed to new query.
Users with poor network connections found Web interface too slow to be useful.

Figure 6.2. 'NomisWeb' interface strengths and weaknesses

General Points
Several users had unusable or slow internet connections making Web access impossible. Most users systems were also several years out-of-date. Web access was generally gained by using PCs that were funded for European projects.
All of the EIU's equipment is hired from Electronic Data Systems (EDS) who have a policy of insisting that Nomis can only be accessed on a 'Stand Alone' PC specifically used for Nomis access. No internal office software is allowed on the same PC for fear of virus infection. EDS have been informed that as the Nomis system only delivers the data as 'Text' there is no risk of virus infection but they insist on retaining the 'Stand Alone' policy. This leads to the situation of EIU users downloading Nomis data onto a floppy disk on one PC and then having to move to another before inserting the data into a document or spreadsheet.
Users complained about being logged off the system automatically after 15 minutes – This was a result of a program running at Durham designed to disconnect any Nomis users that forgot to logout of the system. The program was only supposed to logout users who had been 'idle' for 15 minutes but seemed to just log off all users after 15 minutes regardless. This problem was fixed as soon as it had been notified to us.

Figure 6.3. General points

Generally users thought that the 'New Nomis' command system and NomisWeb were good in terms of ease of access and the time taken to get used to a new database. They liked the concise nature of the command interface and good feedback was received for new commands such as ***make, check, list and search***. The manuals were well received. EIU users are generally very competent and had little trouble in using the new systems compared with 'Old Nomis'. One minor criticism was regarding the many different table layouts available with 'New Nomis' made it difficult to know which one would be appropriate. The 'automatic' layout feature did not always select the desired layout – although it always generated the fewest tables possible these often contained too many columns so wrapping occurred.

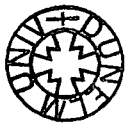
Most EIUs used the command interface rather than NomisWeb purely for the reason that Web access was unavailable (policy reasons) or too slow on their systems. However, most believed that Web access would become available soon.

The weaknesses were considered individually to see if improvements were possible or feasible. For the command system weaknesses, it was decided that more intelligence needed to be built into the layout routines so that it generates larger, fewer tables, which also are not too wide for the screen (less than 132 columns wide). Historical data will be added in time and the 'Old Nomis' specific features need to be considered individually to see how well used they have been in the past and determine how vital they are for the 'New Nomis' system.

For the NomisWeb system the lack of certain command line functionality could not be avoided as the main purpose of the interface was to make accessing Nomis data easy for novice users. The next main development phase will be to write an enhanced Web interface which offer such functionality as the resources becomes available and are prioritised by ONS.

6.4. Performance of 'New Nomis' command interface and NomisWeb

The performance levels of 'New Nomis' have been monitored using the daily, weekly and monthly accounting records. Figure 6.4. compares the percentage usage of 'Old



Nomis', 'New Nomis' (command system) and NomisWeb counting back from November 2000. Figure 6.5. compares the unit usage of 'Old Nomis', 'New Nomis' (command system) and NomisWeb. One unit is the equivalent of 1 minute online for 'Old Nomis' and 'New Nomis' (command system) or a number of cells downloaded for NomisWeb.

The graphs in figure 6.4. and figure 6.5. suggest the increased popularity of the Web interface (yellow) since its release and Nomis2 (red) has become as popular as Nomis1 (purple). As more data sets are added to Nomis2 it will overtake Nomis1 which will eventually be phased out (at the end of 2000). If Nomis2 and NomisWeb combined are considered 'New Nomis' then both figures suggest that on 1st April 2000 'New Nomis' was accounting for approximately 80% of online income. Of this 80%, the Web interface was generating 60% and the command interface another 20%. As both of these graphs contain only 18 months of data, it is difficult to determine if this usage level will continue. It also does not indicate whether or not the figures are due to the popularity of the interface or is in some cases just users trying out the new interfaces.

Figure 6.6. compares the number of data extractions on 'Old Nomis' with 'New Nomis' daily. Figure 6.7. compares the length of time spent online using 'Old Nomis' and 'New Nomis' daily. Figure 6.8. compares the number of extractions of 'Old Nomis' with 'New Nomis' weekly. These three Figures (6.6. through 6.8.) suggest that by mid January 2000 'New Nomis' had become more widely used than 'Old Nomis' and has remained so since then. Again these figures may not be due to the popularity of the new system but down to the idea that post February 2000 unemployment and vacancy data would only be available on the 'New Nomis' system.

Figure 6.9. shows 'Old Nomis' against 'New Nomis' daily income in units. This shows that by mid January 'New Nomis' was generating more income than 'Old Nomis'.

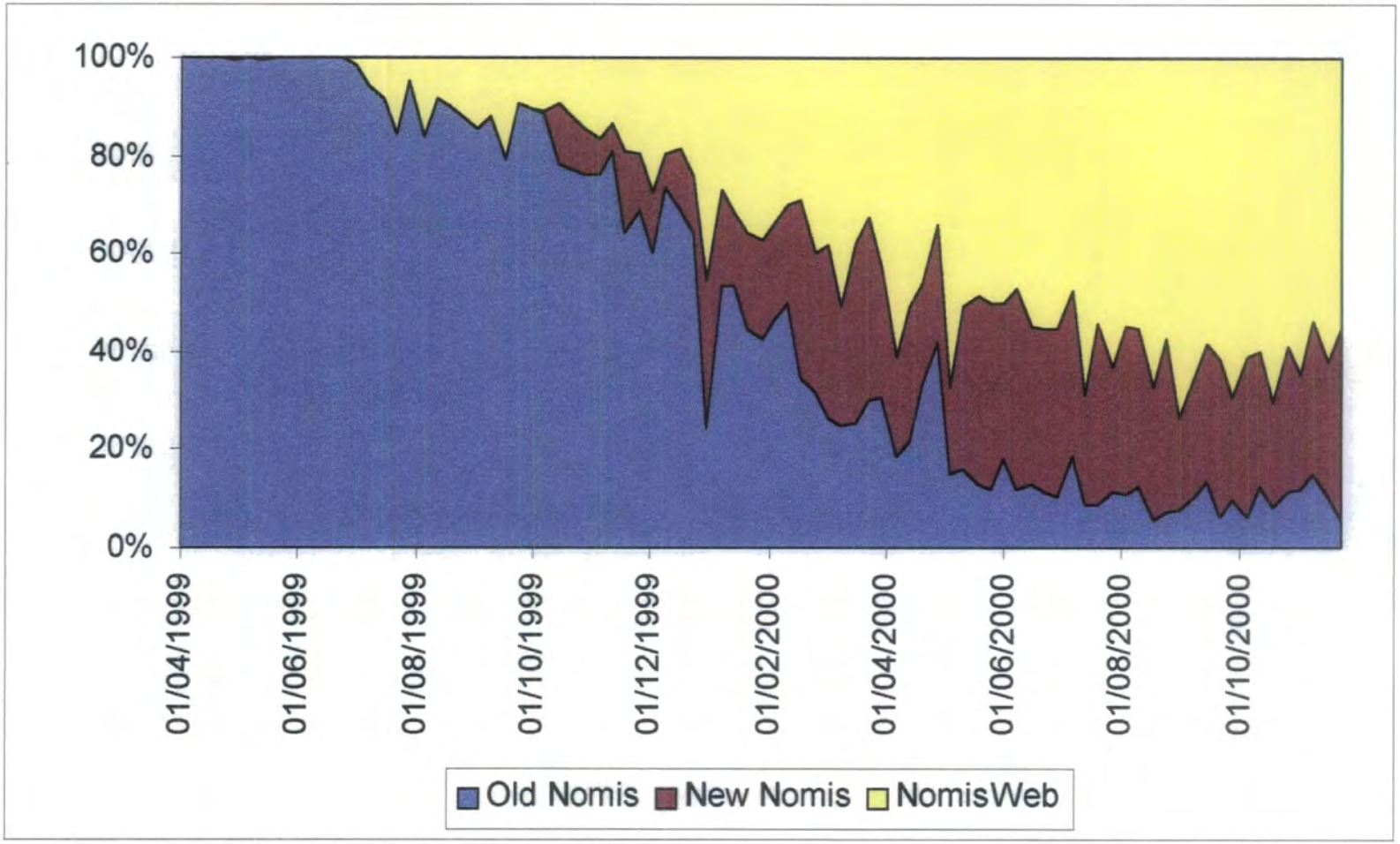


Figure 6.4. 'Old Nomis', 'New Nomis' and NomisWeb weekly performance as a percentage

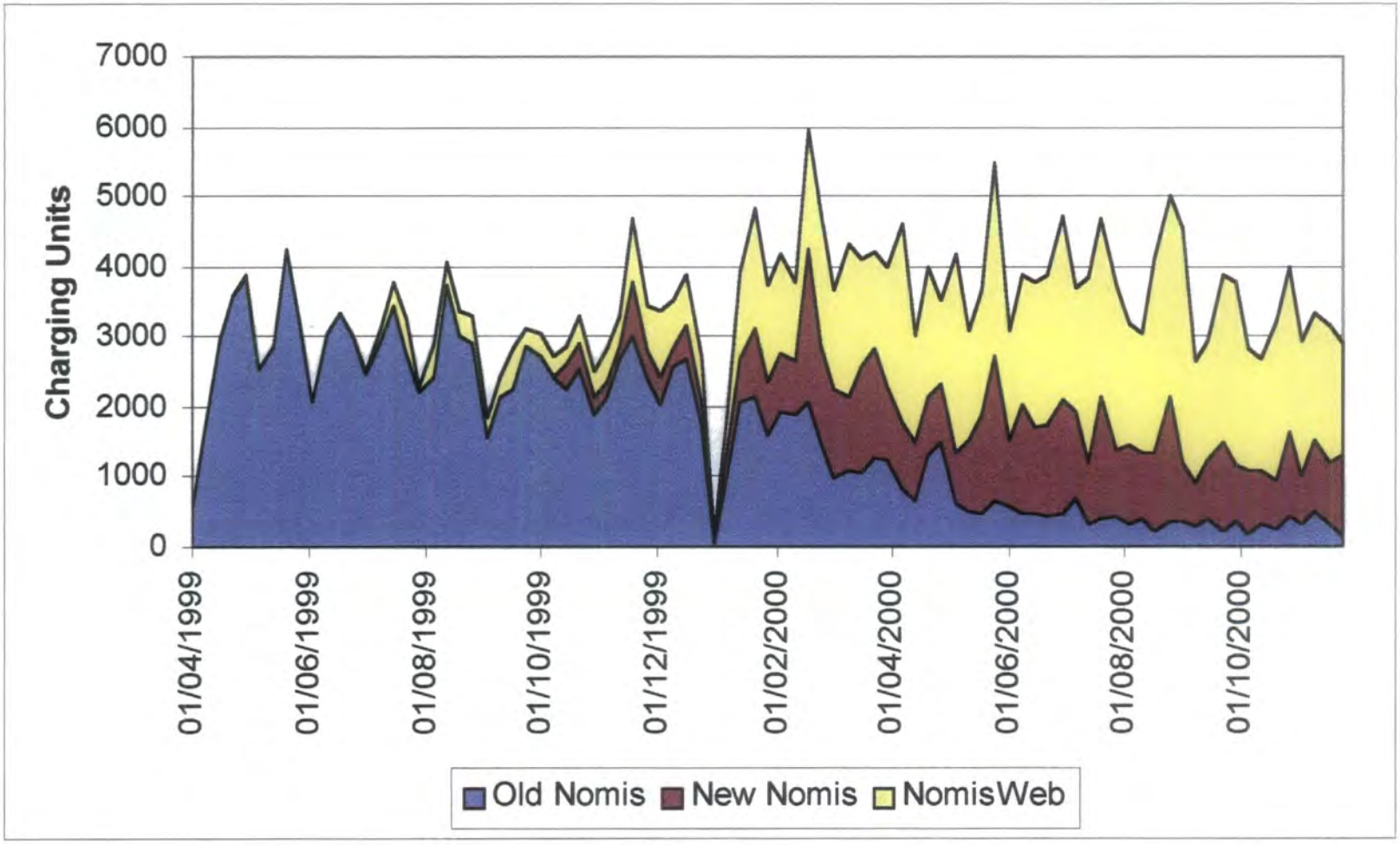


Figure 6.5. 'Old Nomis', 'New Nomis' and NomisWeb weekly usage

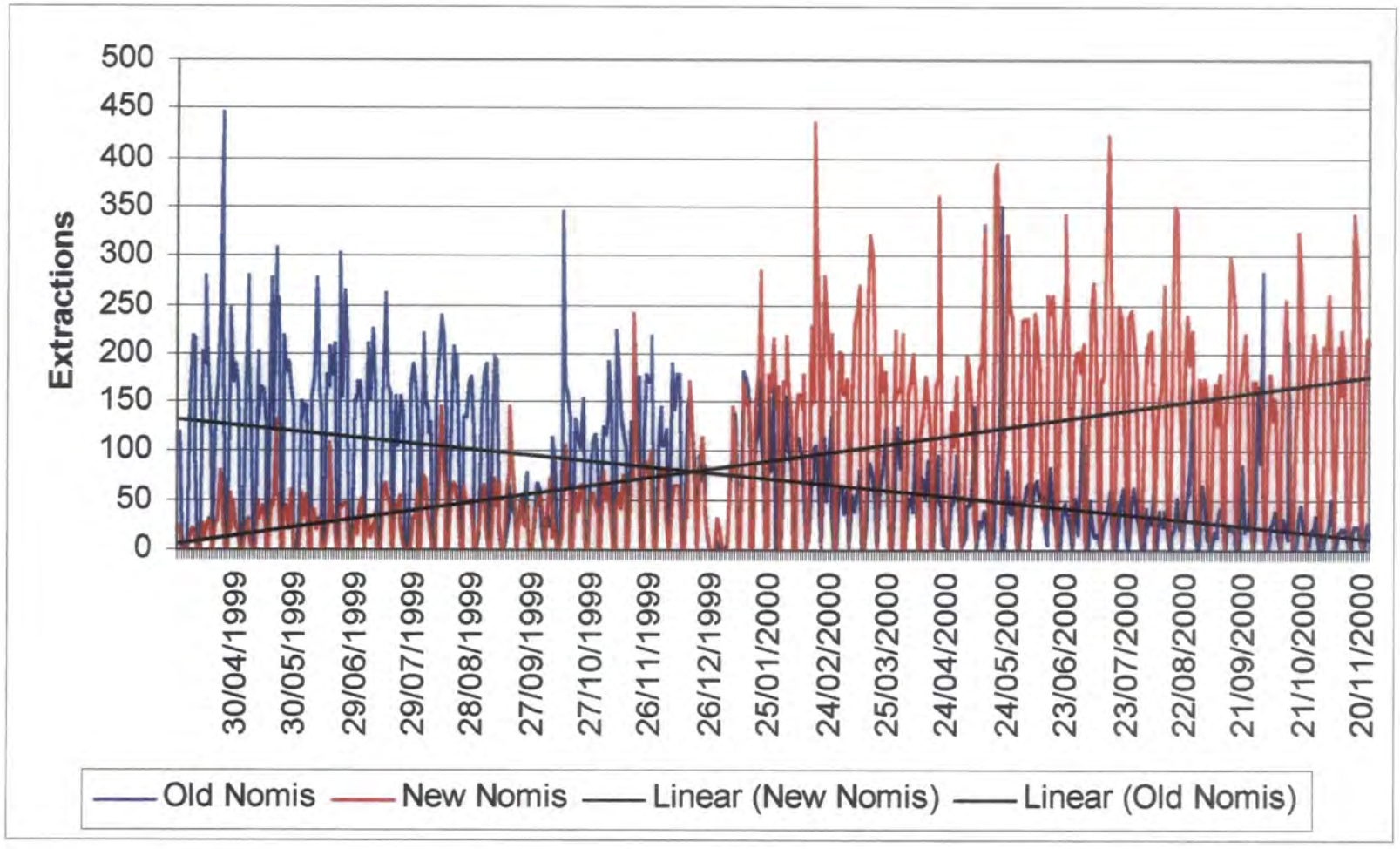


Figure 6.6. 'Old Nomis' and 'New Nomis' extractions (daily)

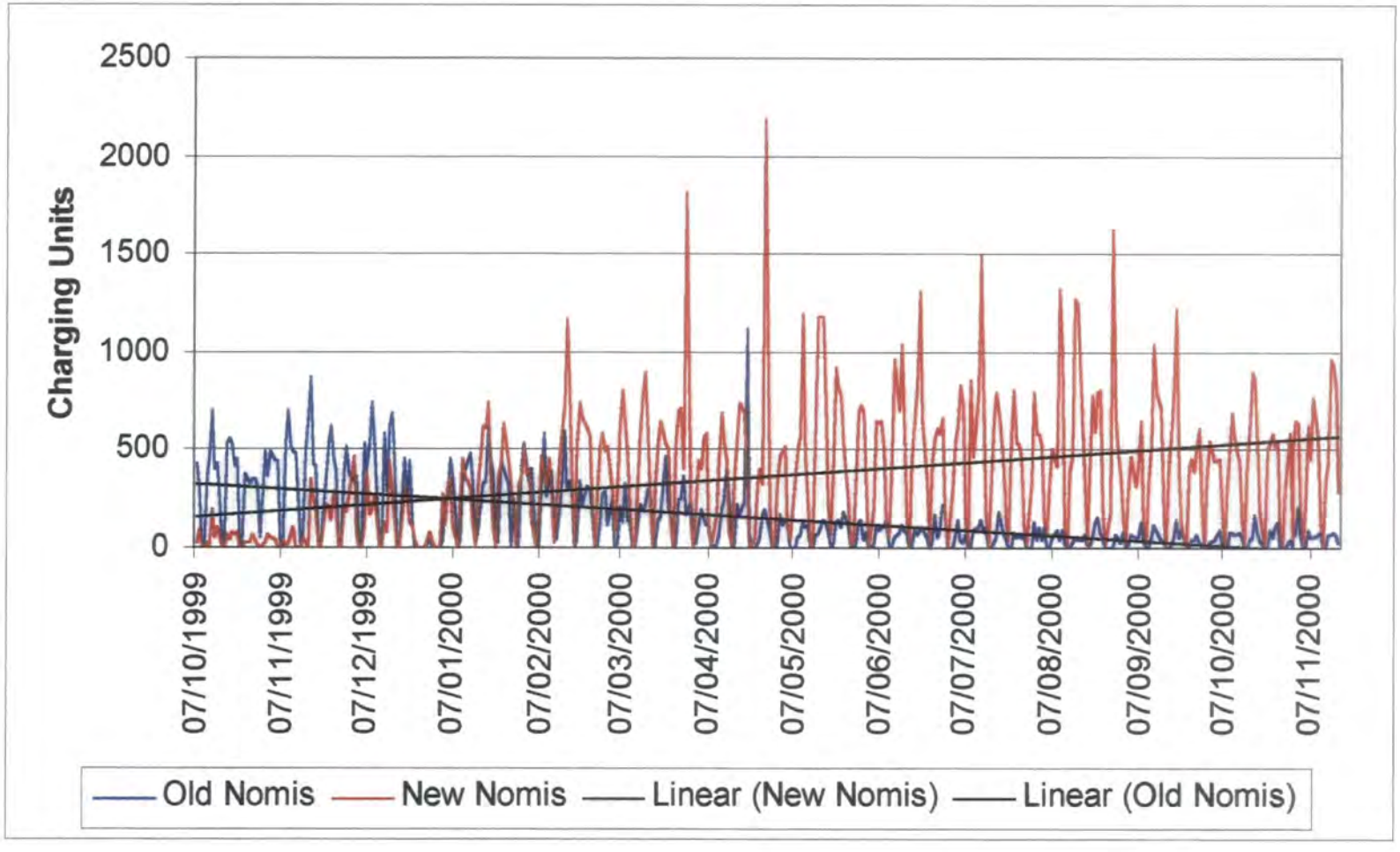


Figure 6.7. 'Old Nomis' and 'New Nomis' time spent online

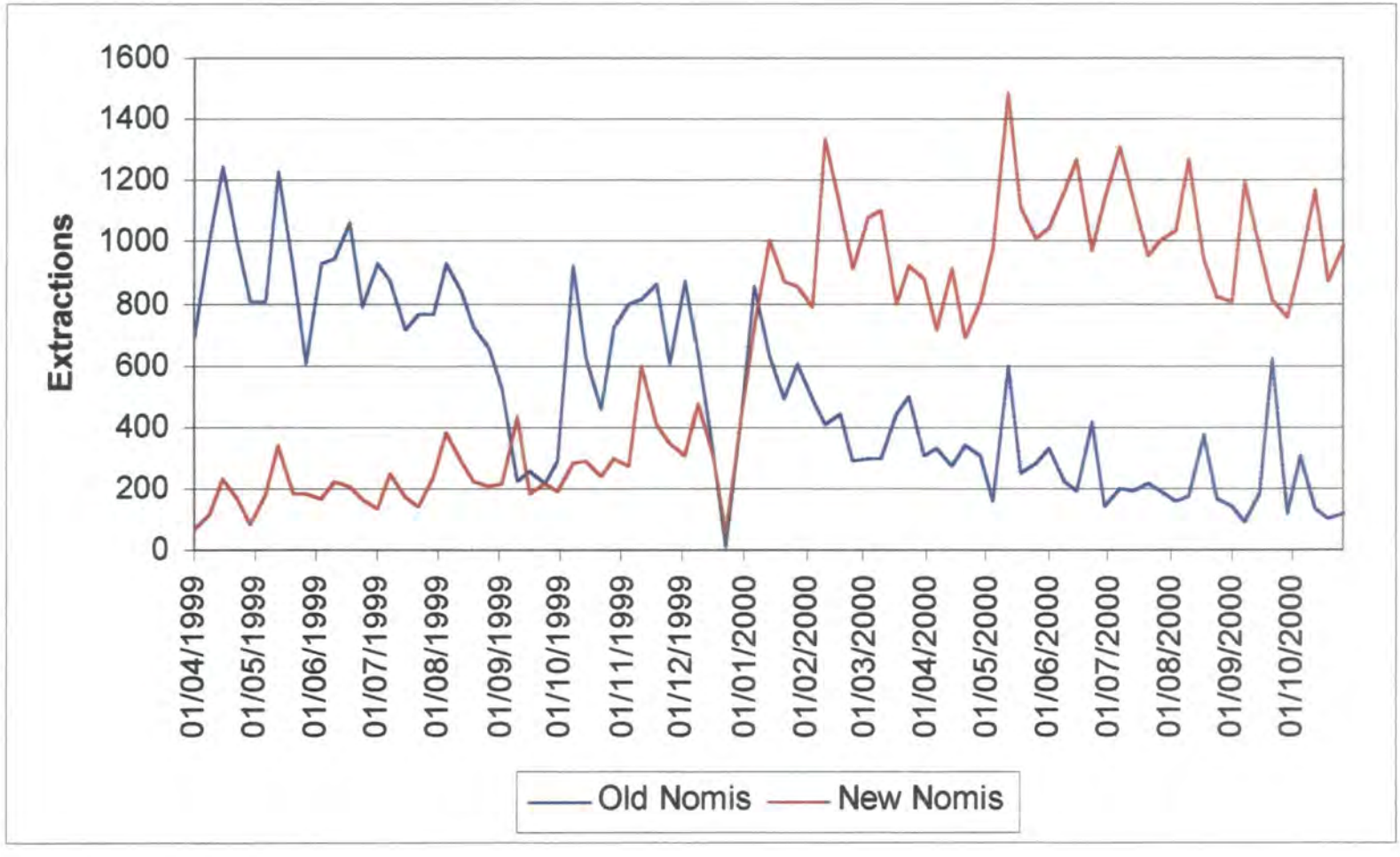


Figure 6.8. 'Old Nomis' and 'New Nomis' extractions (weekly)

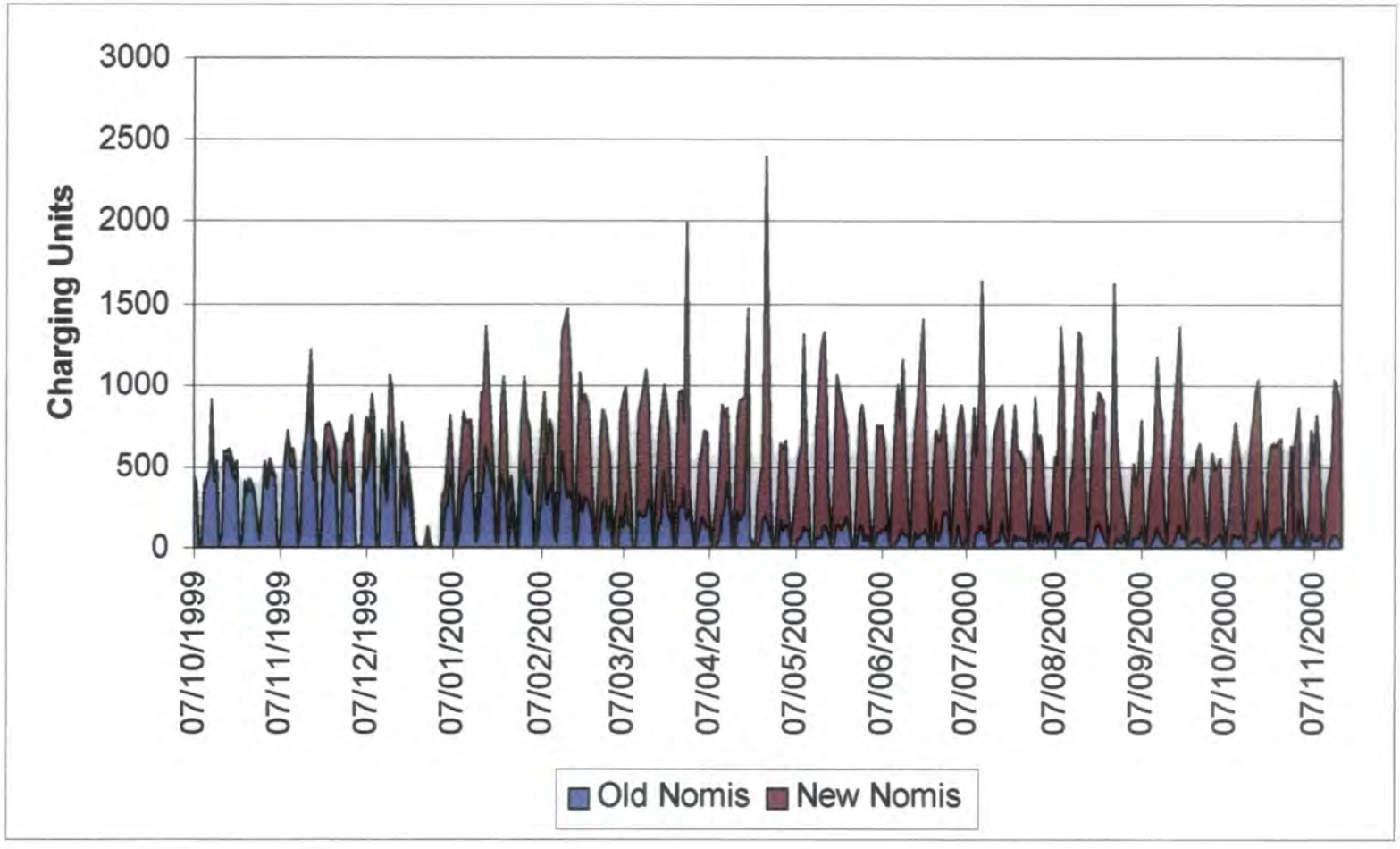


Figure 6.9. 'Old Nomis' and 'New Nomis' daily income

6.5. Conclusion

The transition from 'old' to 'new' Nomis has been a relatively smooth one for users. Help-desk calls have been low, given the task of moving 800 accounts and several thousand users onto a new system. The policy of transition, rather than a 'big-bang' sudden change, has also worked with the implementation of new data series. We were very concerned that the area for maximum confusion would be with new users, who would need to learn two command syntaxes, but overall the training and support strategies minimised problems there.

Overall the 'New System' has been successful in terms of the responses received from users and in the amount of use. Minor bugs are occasionally reported, but many are fixed immediately. With a development team of three and a relatively small team at Nomis (ten people) there was a limit to the amount of testing that could be done. As soon as we regarded the system as stable it was released to users. This was the best means of fully testing 'New Nomis'. In early May 2000 there are still several data series to be transferred to the new system (including our historical data). Once this has been completed 'Old Nomis' can be taken offline and the next generation of Nomis services will be fully in place.

Chapter 7 Conclusion

By 12th July 2000 all of the current employment, unemployment and vacancy data were withdrawn from the 'Old Nomis' system. This was one of the final stages of the transition from the 'Old Nomis' system to 'New Nomis'.

This study has described the development of the 'Old Nomis' system as it evolved from a small, simple database with a single data set, to a larger, complex model with over 90 data sets and a user base of over 800. The strengths and weaknesses of the 'Old Nomis' system as it developed over 15 years were discussed. The results suggested that a complete redesign of the system was required. The next stage of the study was to investigate the different methodologies that could be incorporated to implement the redesign. This resulted in a 'home grown' system written in C++ using object-oriented programming technology.

Next the development of the 'New Nomis' system which started early in 1997 was discussed. The initial task was to develop an enhanced version of the 'Old Nomis' command interface with the focus on producing a consistent yet simplistic command syntax. Other improvements were added such as context sensitive help and the ability to save and reuse popular queries. Another important part of the initial development was to decide on an appropriate data storage and retrieval structure for the system. Hashing was identified as being the most suitable method and this allowed the system to use only one file per data set (as opposed to one file per data set per time period with 'Old Nomis'). This resulted in far more efficient retrieval of time-series data as the system only needed to open a single file.

The next stage of development was to improve the overall geographical management of Nomis replacing the two existing databases (one for building higher aggregates and the other providing lookup definitions) with a single integrated database that was constantly up to date. The new system included facilities for housing and managing frequently changing geography types which was a major downfall of the 'Old Nomis'

system. 'Quick Build' definitions and indirection files ensure that less time is spent reloading data.

With the command system in place development could start on a GUI for the 'New Nomis' system. Various options were considered and it was decided to use the World Wide Web to provide the interface using HTML and forms. The Web interface was released to users in April 1999 and within a year was the most popular way to access Nomis data. An article titled 'Nomis via the Web' featured in the July 2000 edition of Labour Market Trends (produced by National Statistics).

The evaluation of the system suggested that the 'New Nomis' system met the design criteria as set out. It also showed that users response to the 'New Nomis' system was generally very positive. This was backed up by the performance graphs which illustrated the increased usage and income generated by the 'New Nomis' system.

The system is by no means complete. There are always improvements that can be made in terms of new facilities for users (more output formats), more efficient data retrieval (at present if a user requires several items of data from the same data set the system will retrieve each item individually rather than all at once). The Web interface was developed to allow users with all levels of IT equipment access to Nomis data via the Web. It is, therefore, a very simplistic model which does not offer the full functionality of the command interface.

My initial role during the development of the 'New Nomis' system was to develop the command interface. I was then involved in the development of NomisWeb along with the system manager. Working on the Web interface proved to be both challenging and rewarding. Given the response to "New Nomis" and to NomisWeb in particular the key task for the future is to start work on the design and implementation of an enhanced Web interface taking advantage of the latest Web technology. I believe that the future of Nomis, like so many other statistical dissemination services, lies in the World Wide Web.

References

Avison, D.E. and Fitzgerald G. 1995: **Information Systems Development: Methodologies, Techniques and Tools.** Berkshire, McGraw-Hill

Booch, G. 1991: **Object Oriented Design with Applications.** California, Benjamin/Cummings

Bosak, J. 1997: '**XML, Java, and the Future of the Web'.** (Web document) http://metalab.unc.edu/pub/sun-nfo/stadards/xml/why/xwlapps.htm

Bourne, S.R. 1982: **The UNIX System**. Canada, Addison-Wesley

Coad, P and Yourdon, E. 1991: **Object Oriented Analysis.** New Jersey, Prentice Hall

Date, C.J. 1990: **An Introduction to Database Systems, Volume 1, Fifth Edition.** New York, Addison-Wesley

Felton, Mark 1997: **CGI Internet Programming with C++ and C**. New Jersey, Prentice Hall

Flanagan, D. 1997: **Java in a Nutshell, Second Edition**. Sebastopol CA, O'Reilly & Associates

Flanagan, D. 1997: **JavaScript the Definitive Guide**. Sebastopol CA, O'Reilly & Associates

Gundavaram, S. 1996: **CGI Programming**. Sebastopol CA, O'Reilly & Associates

Horowitz E. & Sahni S. 1976: **Fundamentals of Data Structures.** London, Pitman

Ladd, S.R. 1996: **C++ Components and Algorithms, Third Edition.** New York, M&T Books

Martin, J and Odell, J. 1992: **Object Oriented Analysis and Design.** New Jersey, Prentice Hall

Morgan, B. 1997: **'CORBA meets Java'.** (Web document) http://www.javaworld.com/javaworld/jw-10-1997/jw-10-corbajava.html

Musciano C. & Kennedy B. 1997: **HTML the Definitive Guide, Second Edition.** Sebastopol CA, O'Reilly & Associates

NOMIS, 1994**: Tender for the Department of Employment's National Online Manpower Information System (NOMIS)**

Parsons, D. 1994: **Object-Oriented Programming**. London, DP Publications Ltd

Sutherland, S.A.W. and Blake, A: 2000: 'Nomis via the Web'**. Labour Market Trends** (National Statistics), Vol 108, No 7, July 2000 (pp 349 –352)

Glossary

AES

The "Annual Employment Survey" data set contains data from an annual survey of employees in employment providing detailed employment counts for local areas down to ward level.

Bucket

Hash tables are divided into buckets. Each bucket can contain one or more record.

Byte-code

Java programs are converted into byte-code. Byte-code is a platform-independent code allowing Java programs to be executed on any machine.

C

A general-purpose programming language originally designed by Dennis Ritchie.

C++

An object-oriented programming language originally developed by Bjarne Stroustrup

CGI

The Common Gateway Interface is the part of a Web server that can communicate with programs running on the same server. It is used to pass data between the end user and the server.

CORBA

The Common Object Request Broker Architecture is a specification that defines how distributed objects can interoperate.

County

A county is a geographical area used in Nomis. In England and Wales these areas are local authority district based and in Scotland region councils. For Nomis purposes there are 66 counties in Great Britain.

CPD

The Central Postcode Directory defines which postcodes exist in the UK. It is updated twice a year.

CSV

Using Nomis output can be generated into Comma Separated Value format. Fields are separated by commas. Most modern spreadsheet packages can import CSV files directly.

EIU

Each region of Great Britain has an Employment Information Unit.

Forms

Forms is part of the HyperText Markup Language that makes it possible to create documents that can collect and process user input and then form suitable responses.

FORTRAN 77

A programming language designed to solve problems involving numerical computation. Although first designed in the 1960s, FORTRAN 77 is the 1977 standard of the programming language.

FTP

The File Transfer Protocol is used as a way of transferring data files across the internet.

Gb

A Gigabyte of data is 1024x1024x1024 (1,073,741,824) bytes of data.

GUI

A Graphical User Interface is a graphical based interface allowing users a simple way of selecting information on a screen.

HTML

HyperText Markup Language was developed as an authoring language for creating and sharing integrated electronic documents over the Internet.

HTTP

Using the World Wide Web, user requests (and server responses) are formatted using the rules of the HyperText Transfer Protocol standard.

IDL

An Interface Description Language is used in conjunction with CORBA.

Ingres

Ingres is a commercial Relational Database Management System (RDBMS).

JANET

The Joint Academic Network is a countrywide network linking together various Universities.

Java

Java is an Object Oriented multiple platform programming language developed by Sun Microsystems.

JavaScript

JavaScript is an interpreted programming language written specifically for Web programming.

Local Authority District

A local authority district is geographical area. For Nomis purposes there are 485 lads in the UK.

LFSA

The "Labour Force Survey - Annual" data set contains data from an annual survey carried out by interviewing people about their personal circumstances and work.

LIFO

Used in programming terms when referring to a "Stack" system it stands for Last In First Out.

Lmis

Labour market information system is the name given to the Sun Microsystem machine which houses the Nomis system.

MIGR

The "Migrations" data set contains data from a quarterly analysis of migrations derived from the National Health Service Central Register (NHSCR).

MSC

The Manpower Services Commission.

MTS

The Michigan Terminal System was an operating system that ran on mainframes housed at the University of Durham from 1982 – 1992.

NES

The "New Earnings Survey" data set contains data from an annual survey carried out to determine average wages/hours worked etc for areas across the UK.

Nomis

Nomis is an on-line database providing remote access to a comprehensive range of official labour market related statistics.

NUMAC

Northumbrian Universities Multiple Access Computer. This was an agreement drawn up between Durham and Newcastle University to share the computing resources based at Newcastle.

OMG

The Object Management Group is a group of over 700 companies that collectively define open standards for object computing.

ONS

The Office for National Statistics is the organisation that owns the Nomis system.

OOP

Object-Oriented Programming allows the representation of objects found in the real world as objects in software.

Oracle

Oracle is a commercial Relational Database Management System (RDBMS).

ORB

An Object Request Broker allows CORBA objects to communicate with each other.

Parliamentary constituency area

A parliamentary constituency area is geographical area.

PEST

The "Population Estimates" data set contains mid-year estimates of population based on results from the latest Census of Population.

PPBD

The "Population Projections Births and Deaths" data set contains population projections produced by the Office of Population Censuses and Surveys (OPCS), for Wales by the Welsh Office, and for Scotland by the General Registrar's Office (GRO).

Press Notice/ Press Release

This occurs every month (usually on the second Wednesday) when the new unemployment figures are released to the public.

RDBMS

A Relational Database Management System is a suite of packages allowing the creation and manipulation of a Relation Database system. Two such systems are Oracle and Ingres.

Region

For Nomis purposes regions refer to standard statistical regions or standard economic planning regions. There are 12 regions in the UK.

SGML

Standard Generalized Markup Language allows documents to describe their own grammar.

SQL

Structured Query Language is used to formulate operations that define and manipulate data in relational form. SQL forms part of a RDBMS such as Oracle or Ingres.

SWOT analysis

SWOT stands for Strengths Weaknesses Opportunities Threats. The idea is that you identify the strengths and weaknesses of your service, pinpoint opportunities and note threats.

TEC

Training and Enterprise Councils provide a range of services for individuals and organisations, which help to improve the economic prosperity in their local area. They also define their own geographic areas for use on the Nomis system.

Unitary Authority

A unitary authority is a geographical area used in Nomis. It represents unitary authorities that were operational on 1 April 1997.

UDAD

The "Unemployment Destinations by Age/Duration" data set contains destinations data.

UNIX

A family of computer operating systems originally developed at Bell Laboratories.

USOCAD

The "Unemployment Stocks by Occupation, Age and Duration" data set contains a quarterly count of unemployed claimants analysed by their sought and usual occupation. It also contains an age and duration breakdown.

USW91

The "Unemployment Stocks – 1991 frozen ward base" data set contains a monthly count of claimants who are claiming benefit on the count date.

W3C

The World Wide Web Consortium was formed in order to define the standard versions of HTML. It is also responsible for standardizing and technology related to the World Wide Web such as the HTTP standard.

Ward

Wards form the building blocks for most administrative areas used on Nomis.

WWW

The World Wide Web is the open community of hypertext-enabled document servers and readers on the internet.

XML

Extensible Markup Language is an enhanced version of HTML which allows users to define their own tags.