DESKTOP DOCUMENTATION: INDIVIDUAL INCOME TAX MICRODATA, SOI

Martin David, Economics Department, University of Wisconsin - Madison
1180 Observatory Dr., Madison, WI 53706

KEY WORDS: Metadata, information facilities, text and image retrieval

1 Metadata and Statistics of Income (SOI)

The SOI Individual Income Tax Return Sample. The SOI program of collecting statistical data from individual income tax returns dates from 1916 (Coleman 1988). Major aspects of accounting for tax liability are captured in a probability sample of returns. Microdata available from this program are called the Tax Model. The name captures the capability of the data to quantify impacts of changes in tax law on the revenue yield and distribution of tax payments. The tax model was made famous by Pechman (1985).

Metadata are information about statistical data. Paper documentation of past years was often incomplete, inaccessible, or unavailable. The metadata capability described remedies these problems by incorporating electronic documents created in the production of SOI data. Metadata system encompass more than a data dictionary and a published scientific design. More information is necessary because variables are tested for integrity and consistency to the logic of tax forms. A relational database (RDBMS) organizes metadata and reveals implicit links among types of documentation available. It also points to electronic documents, to publications, and to graphic images of annotated tax forms. The metadata system can be integrated with production of data. It can be augmented as analysis is carried out. It provides a growing record of the scientific findings in the microdata (David, 1993).

Organization of the paper. Statistical data are generated within a conceptual framework. The operations that sample a universe, that capture reality in quantitative data, that test the integrity of data, and that reduce data to parameters of statistical interest are implemented by protocols that ensure reproducibility and minimize variability of the parameters estimated. Meaningful inference from the resulting data requires knowledge of these operations as well as access to the sample data realized. DMS refers to a data and software system that contains and organizes information needed by data analysts. Sections 2-3 answer questions about the value and capabilities of a DMS. Section 4 presents capabilities of a DMS for the 1992 Tax Model, *InfoSys*, which I developed. Section 5 evaluates what we have learned from assembling and testing the prototype.

2 Content of a Desktop metadata system (DMS)

Data are the objects of statistical analysis. Metadata are information about the data. That information pertains to all phases of a statistical research project: Design governs data collection. Execution of the design captures data, including measurement errors, and processing errors. Analysis locates deficiencies in the design and execution. Corresponding documentation will be called outcomes. Analysis also estimates parameters of interest. Analysis provides information in two senses. Tabular aggregates, including "control totals" are convenient for many analytical purposes. Documents that discuss and evaluate parameter estimates are not only the end product; they also, very importantly, become the references that a continuing community of analysts need to prepare the foundation for further analysis.

Integrating documentation from design to outcomes. Capturing all aspects of information pertaining to statistical data and presenting that information, or metadata in a convenient manner has eluded most data producers in the past. Impediments to complete metadata have several aspects. The *volume* of material is daunting. metadata are stored on several *media*. Some information is text, some is numerical, and some is graphic. *Timing* of "documentation" is inappropriate. Typically, public documentation is undertaken after preliminary outcomes from the data are known. Knowledge is *distributed*. Large data collection activities involve a team of people, none of whom commands encyclopedic knowledge of all processing decisions; none can anticipate all the complex statistical questions that will be posed.

Common computational capabilities can eliminate each of these problems. The volume of pertinent information can be condensed on optical storage media. Different types of metadata — text,

graphics, and numbers — can each be stored electronically. They can be retrieved using computational algorithms that are adapted to different datatypes. The timing problem in producing documentation can be eliminated. Specifications for design, execution, and analysis can be made legible to non-programmers through the use of auxiliary software. Disciplined use of the word processor, spreadsheets, and common database languages, such as SQL can greatly increase the power of specifications to be read by novices. Electronic mail, network servers, and bulletin boards facilitate assembly of information that is spread over a team of workers.

The diversity of material that is encompassed by metadata, is illustrated by tools used in the SOI and the Treasury OTA in producing and analyzing the Individual Tax Return sample (Table 1). Clearly, books, journals, images, databases, electronic documents, and code for data processing algorithms all contribute to documentation. It is now possible to keep all of these items in an electronic format.

Most publications and text material can be kept in electronic documents; more complex pages in a book can be stored as images. Spreadsheets can be stored as such, or easily transformed to databases. Portability of documentation in these forms requires relatively universal access to the software that retrieves images and manages electronic documents.

Algorithms that transform variables, maintain data integrity, and verify data consistency constitute the most difficult class of metadata. Use of many aliases to refer to the same variable and the illegible character of many programming languages create the difficulty. Many programming languages are cryptic, unstructured, and lacking disciplined naming conventions. If many users are to understand such algorithms, they must have access to a concordance of all aliases for a variable. They must also know the relevant programming language. In addition, the algorithms must be cataloged and preserved in a library.

Query languages associated with relational databases reduce these problems. The naming of variables is controlled by the database, and the logic of data manipulation is easily parsed. Lastly, Codd's insistence on data independence and integrity assures a standard of performance that is not present in lower level programming. See Date (1988: 15, 245, 444).

Inputs to the SOI DMS Individual Income Tax Return data are captured via a relational database (ORACLE). Specifications for design, integrity tests and consistency tests are maintained in WORDPERFECT. Desktop publishing of *SOI Bulletin* also begins with WORDPERFECT. Spreadsheets are embedded in LOTUS. Creating an information system from these capabilities requires four steps:

- The electronic media must be collected.
- The versions of the media must be controlled.
- The material must be integrated.
- The material must be maintained as corrections, supplements, and extensions of scope cause earlier versions to become obsolete.

3 Why do we need desktop metadata?

The need for DMS has seven dimensions.

Design is not fully published. Table 1 indicates that information on design, execution, and outcomes is not fully published. Errors in the data and related processing minutiae can not economically be incorporated into journals and books. They can be preserved electronically.

Interest in many datasets peaks at the time of their first release for analysis. That is certainly true of the Tax Model, which is released in preliminary form less than two months before the President presents budget proposals (and corresponding revenue effects) to the Congress. Precisely at that time, it is most difficult to publish complete, error-free, and up-to-date metadata.

The DMS can be accumulated during data collection and processing. The care that must be exercised to provide all members of the data collecting team with up-to-date instructions and specifications can be used to provide bibliographic control on the archival version of the DMS. The DMS is therefore always current and up-to-date. It can be issued as soon as preliminary versions of the data are available.

Execution is summarized in aggregates; important detail is lost. The volume of information about execution of a scientific design makes it impossible to publish all details. In the best of published documentation (Jabine, King, and Petroni 1990, Brooks and Bailar 1978), summary statistics on error rates are published. Many relations between the size of errors and covariates are permanently lost. The solution is to include arrays of information on errors conditioned on relevant variables in a DMS.

Archival versions are not identified. The *InfoSys* prototype DMS which is described below does not contain the array of sampling probabilities — That array was available in electronic form during processing. At the time that the *InfoSys* prototype was assembled, no electronic version could be found. The array had already been modified for the processing of the next tax return year.

Text, images, and databases must be combined. See Section 2.

Metadata are dynamic. On-going studies of complex data generate much new knowledge that can be referenced through a bibliographic database that is made part of the DMS. Because such knowledge appears first in ephemera, it is important for the data collector to collect the references. (This is being done by many data collections. It is particularly important for SOI, because much analysis is unpublished.) Weaknesses of particular designs or data collections also surface in this continuing analysis. Lastly, the scope of questions that may be asked of an existing data set expands as the collection is replicated over time and on different samples.

Need to computerize search of common language descriptions. Materials that were gathered for a prototype DMS for the Tax Model sample of 1992 returns *InfoSys* comprised 1500 pages of paper documents, corresponding WordPerfect documents, and approximately two megabytes of spreadsheets. Finding information in this collation is difficult. Different aliases are used for variables at different stages of processing. The organization of the *Editing Manual* does not conform to the order in which tax forms are completed or filed. The numbering of integrity tests and consistency tests does not offer clues to the variables involved. No indices exist for any of the documents. The spreadsheets contain large redundancies, and at the same time do not encompass the entire data collection.

A novice needs to be able to search these material electronically using English language descriptions or analogies from the tax return document. Otherwise she needs to be an apprentice to an expert who can adjudicate the meaning of variables with similar descriptions. Anecdotes from the US Treasury suggest that novices fear using the Tax Model because variable descriptions are incomplete and algorithms for the calculation of transformations are not easily obtained.

Each of these needs can be satisfied with a DMS.

SOI needs automation in updating processing from year to year. The principal cost of replicating the Tax Model from year-to-year is the labor involved in adjusting variable names used in integrity tests, consistency tests, and the data product. These adjustments take three forms: (a) additional variables are captured from the tax form; conversely, variables used in the preceding year are deleted. (b) Each variable captured from the preceding year and the current year is analyzed to assure that the meaning has not changed. When meanings are significantly different because of change in tax code, new variable names are assigned in the current year. (c) Changes in tax code force changes in the algorithms for integrity and consistency testing. The processes involved in the first two steps can be automated when metadata are in a database (David and Robbin 1992). Furthermore the database makes it possible to generate the discordance — the list of homonyms which represent distinct concepts in Tax Models for different years.

Adjustments (a)-(c) occur over a period of time and are prone to errors. Three factors contribute. Substantive content of the Tax Model varies from year-to-year, depending on policy thrusts. Tax legislation is often not passed until the end of the Congressional session; tax documents can not be designed and printed until after new provisions are enrolled in law. The logic of changes in law are subtle, and synergism with existing provisions is often unforeseen. (For example 1987 tax returns included no entry for investment tax credits, although businesses were entitled to carry-over unused credits from prior years. Designing the tax collection from the forms led to the omission of an important variable for analysis purposes, which had to be retrofitted into the collection design.)

4 Creating a DMS for SOI/Individual returns

Principal concepts organizing *InfoSys*. The *InfoSys* prototype DMS for the 1992 Individual Tax Return sample contains material that describes underlying processing and related data objects. *InfoSys* includes a relational database, documents, and images.

Data can be organized as arrays whose rows reflect entities described by attributes shown in the columns. Labels for rows and columns are

essential to linking data, and manipulating attributes for statistical computations. Labels must be unique, or results will be ambiguous. Arrays must also have unique names.

Labels often are abbreviations or arbitrary combinations of ciphers. To understand each attribute, table, and entity, it is necessary to provide common language explanations of those labels. This description is a meaning for the label. Similarly, when underlying data have been classified with numeric codes, it will be necessary to provide a meaning for the code value. Over and above these meanings, it is useful to provide a semantic principle that explains the logical principles which produce the rows and columns present in the table. In almost all cases such principles are confounded by special cases that must be noted, or analysts will misinterpret the data.

SOI annotated tax forms with the labels for each variable that was edited. These notations were squeezed onto the tax forms, often in odd places. The irregular placement of note and the complexity of the IRS forms led us to scan the documents, producing graphic images. Users' familiarity with the forms makes the images an invaluable reference tool.

Implementing a prototype. Five objectives created priorities for assembling *InfoSys*:

- Obtain a searchable data dictionary for all attributes (called elements) of the output data file, Insole.
- Enable users to scan generic tax forms for information about Insole.
- Establish aliases used to label Insole variables at different stages of processing.
- Create a library of electronic documents and spreadsheets created during data collection.
- Establish bibliographic control of both published documents, electronic documents, and database used in *InfoSys*.

InfoSys includes three types of objects:

Object	Description
Electronic documents	WordPerfect files of principal processing specifications
Images	WordPerfect graphics of Individual Income Tax Forms and Schedules
Relations	PARADOX arrays organized to display stages of SOI processing, the data dictionary, and references to external electronic information, memos, and publications

InfoSys uses the database to facilitate access to other electronic capabilities. The database was derived from text and displays logic implicit in those sources.

Infosys relies on seasoned, off-the-shelf software. This strategy assures: (a) client-users who know how to manipulate the software, (b) reasonable cost and widely available software, and (c) error-free operation of the software. *InfoSys* operates with modest capabilities (IBM-286, 20mb data storage, and 2mb memory). The database software, PARADOX, was chosen because of its interface to spreadsheets and database servers using SQL and its query-by-example capability.

Implementation entailed about four man-months of professional time. Effort concentrated on organizing information and understanding the design and processing. Few "applications" were programmed to make *InfoSys* more user-friendly. That appeared to be a task secondary to the definition of the content and the logical capabilities of the system.

Table 2 gives a bird's eye view of the database. Users can begin their search for information in one of three ways by naming an Insole element, naming a tax form, or initiating a text search of element descriptions. The prototype includes the tax forms with annotations of edited fields. This facility makes it possible to locate most attributes through a visual scan.

5 What have we learned?

Development: Images. Available scanning technology generates satisfactory substitutes for complex paper documents when used as graphic display.

Spreadsheets. Borland has a common data server for spreadsheets and its PARADOX database. Appropriately structured spreadsheets can easily fit in the database. A spreadsheet was the source of the 92DERLMF table.

WP documents. WP documents enter the database in two ways. WP tables and text were incorporated into the database (most notably Section 3 and Appendix G of the *Consistency checking manual*). WP documents also were archived to permit users to read or print chapters.

Portability. *InfoSys* was designed to be exported to other PC environments with PARADOX4.0. The *InfoSys* prototype was installed at the US Treasury and IRS/SOI in May 1994. Duplicating and moving the entire DMS created no problems.

Value-added to existing documentation.

Because *InfoSys* was generated as a global approach to integrating documentation, uniform naming procedures were followed and documented in the database. System tables (RELATION and ATTRIBU3) were created to describe and archive that effort. Those tables provide an inventory of every table and of every column appearing in the database. These tables describe the database and its scope.

FORM-EDI contains information about individual tax forms. No such capability exists in WP documents. The table illustrates the ease with which an index from a published document (*Package X*) can be incorporated into the relational database. It also shows how the database can be used to point to relevant electronic document files.

92DERLMF records variable names used in data capture by PRISM. The table can be more efficiently searched than the spreadsheet from which it was derived.

BUS-FARM establishes an important distinction between similarly labeled fields that may refer to tax return aggregates or enterprise detail. In a relational data structure this difference in unit of analysis is mirrored in separate tables.

PUBLICAT provides for a bibliographic database of documents and analyses.

DOCDIRWP creates a version control for WP documents included in *InfoSys*. It assures that changes to documents can be removed.

Quality of a DMS. A DMS can be generated concurrently with all phases of data collection and dissemination. DMS can support an adequate archive for analysis of successive years of the Tax Model while maintaining version control on all information that they contain. DMS assures uniform use of names in different phases of data development and can create authority lists for all aliases.

Creating a DMS: Numerous tables, spreadsheets, and lists can be more easily updated and checked using the editing capabilities of relational databases than the WP documents and Lotus spreadsheets where updates are currently undertaken.

Cross-references: *InfoSys* makes it clear that indexes and tables of contents to WP documents are doubly valuable when they can be incorporated into the database. Absence of these tools in the current WP documents makes finding specifications for integrity and consistency checks an extremely difficult task.

Ease of use. Startup time: We believe that electronic access to annotated tax forms will aid novices in learning about Insole. Also, searching a complete compilation of memos and documents will aid in locating benchmarks created by others.

Adequacy of pre-existing documentation. We also believe that most user analysts do not have ready access to information on integrities and consistencies imposed on the data.

Responsibility for the DMS. Because SOI has day-to-day responsibility for planning scientific design and executing it, it must take responsibility for preparing the DMS. However, it needs to solicit input from analysts who use SOI data. It should receive copies of working papers. It should solicit information about errors and anomalies from analysts and it should incorporate notes on error into the DMS. As experience with the data accumulates over time, the DMS will grow and become ever more valuable.

Flexibility. Tables in the relational database can easily be reorganized. Columns are easily renamed. This is a tool that can assist in defining classes of similar attributes. The principal constraint on reorganization is the fact that the system tables must also be amended. Mark and Roussopoulos (1986) offer an approach to automating the updating of system tables that can be accomplished within PARADOX.

References

Brooks, Camilla & Barbara Bailar. 1978. An Error Profile: Employment as Measured by the Current Population Survey: Statistical Policy Working Paper #2. Washington DC: Executive Office of the President (US)/Statistical Policy Office/Office of Management and Budget.

Coleman, Michael J. Fall 1988. Statistics of Income Studies of Individual Income and Taxes. SOI Bulletin 8:2, 63-80

David, Martin H. 1993. Systems for metadata: documenting scientific databases. Proceedings of the Twenty-Sixth Annual Hawaii International Conference on Systems Sciences, 3:460-69.

Date, C. J. 1988. An Introduction to Database Systems. Reading MA: Addison-Wesley.

David, Martin H. and Alice Robbin. 1992. Building new Infrastructures for the Social Science Enterprise: Final Report to the National Science Foundation on the SIPP ACCESS Project, November 1984 - December, 1991. Madison, WI: Institute for Research on Poverty. (2 Volumes, ca. 400 pp.)

Jabine, Thomas B., Karen E. King, and Rita J. Petroni. 1990. Survey of Income and Program Participation: Quality Profile. Bureau of the Census. US Dept. of Commerce.

Pechman, Joseph A. 1985. Who Paid the Taxes: 1966-85. Washington DC: Brookings Institution.

Mark, Leo and Nick Roussopoulos. December 1986. Metadata management. Computer, 26-36.

Table 1 Materials used to guide analysis of Individual Tax Return Data

Phase of scientific effort	Data-type	Metadata	Access
Design			
	Book	*Package X*	Published
		SOI Bulletin	Published
	Spreadsheet	Sampling rates	Published
		Concordance: ORACLE and processing labels: Tax form elements	Restricted
	WP documents, or paper text	Sampling memo	Restricted
		Integrity and consistency checks	Restricted
	Algorithms	Integrity and consistency checks	Restricted
	Images	PRISM data editing screens	Restricted
Execution	Database	Error rates on integrity checks	Restricted
Outcomes	Spreadsheet	Analytical tables	Published
Analysis	Book or database	*IRC and Regulations*	Published
	Book	*SOI Bulletin*	Published
	Documents	Miscellaneous pre-prints	Ephemera

Table 2. Tables in the infosys database

Tables	Entities	Semantic principle: keywd
	A. System tables	
RELATION	Tables	Describes each infosys table (relation): structure and links
ATTRIBU3	Variables	Describes label on each table column (or attribute)
SOURCE	Value	Meaning of entries in table column SECTION 3: SOURCE
	B. Tables pertaining to *Insole*	
FORM-EDI	Tax forms, prism tables	Describes (1) individual tax return forms or schedules and (2) related "prism" tables
APPGCODE	Variables	Describes coded insole attribute
APPG2	Values	Meanings of codes for variables in APPGCODE
STATE	Values	Relation between irs districts and states
SECTION3	Variables	Describes money amounts, control elements
BUS-FARM	Enterprise	Describes farm and business enterprise amounts
92DERLMF	PRISM variables	Describes variables captured by "PRISM" processing
INSOLEDF	Variables checked	Relation between aliases used in consistency checking and insole variable names
	C. Tables for bibliographic and file control	
DOCDIRWP	WordPerfect documents	Shows DOS filename and creation date to establish version control
PUBLICAT	Publications	Bibliographic description of insole relevant publications, memoranda, and documents

Acknowledgements

Inspiration for this project comes from Fritz Scheuren. His untimely resignation from IRS/SOI in the midst of this project created a severe setback for creative dialog on ideas and problems outlined in this paper.

Alice Robbin and Tom Flory's collaboration on the SIPP-ACCESS project gave me the insight to understand the cognitive problems of analysts who deal with a complex dataset (David and Robbin 1992, David 1993).

I am indebted to staff of the Statistics of Income Division, for their able assistance and cooperation in developing this schema and the material contained in InfoSys. Special thanks go to Carl Greene, Lori Eckhardt, Michael Strudler, Marty Shiley, and Susan Eastep who were most helpful in supplying material and answering dumb questions.

I owe a special debt to John Czajka without whose knowledgeable help the project could not have been completed. John participated in every stage of this project and installed InfoSys at the IRS/SOI and the US Treasury OTA.

Reference to Internal Revenue Service Documents

IRS/SOI. [Annual]. *Individual Income Tax Returns [Year] (Publication 1304)*. Washington DC: GPO.

IRS. [Unpublished/restricted]. Internal Revenue Administrative Processing Manual.

IRS/SOI. Memorandum: Weighting and Variability Specifications Package for the SOI 1992.

IRS/SOI. Specifications for consistency testing and related processing of tax year 1992 Statistics of Income Individual Income Tax Return Records, Forms 1040, 1040A, and 1040EZ

IRS. *PackageX: Informational Copies of Federal Tax Forms [Year]* (3 volumes). Washington DC: GPO.