Creating Desktop Documentation: Individual Income Tax Return Microdata

Martin David, University of Wisconsin - Madison

The Statistics of Income (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 [1]. 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 [2]. The tax model was made famous by Pechman (1985). Pechman used the tax model to estimate the burden of the American tax system over a period of nearly thirty years.

Unfortunately, documentation for early years of the Tax Model was incomplete. (Experts have reconstructed a portion of the archive of replicated samples of individual income tax returns by analogy to the order of entries in prior or subsequent years and the corresponding tax return forms and schedules.) This fact signals a major problem in the past, the institutional memory for statistical data has failed, with resulting compromises in our ability to study trends.

A public-use version of the Tax Model has been widely distributed since 1961. It is used for research and advocacy of new tax legislation. Use of the data is no longer restricted to a dozen individuals who spend their careers immersed in tax law and policy. Many users are not familiar with the complex data collection process. They are unaware of errors that inevitably exist in the Tax Model. The community of users needs convenient tools to learn about the Tax Model.

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. Desktop Metadata System (DMS) refers to a desktop data and software system that contains and organizes information needed by data analysts. This paper begins with some background on the current status of SOI individual data. The next two sections answer questions about the value and capabilities of a DMS. Then, capabilities of a DMS for the 1992 Tax Model, InfoSys, which I developed, are presented, followed by a section which evaluates what we have learned from assembling and testing the prototype. The paper concludes with a look at some extensions and discusses next steps.

■ Background on SOI

Complexity and Legislative Changes

Witte (1991) documents the increasing frequency of changes to the Internal Revenue Code (IRC) in the period since 1954. Those changes led to complex accounting (e.g., the Alternative Minimum Tax) and complex definitions of relevant units (e.g., units eligible for the Earned Income Tax Credit up to 1992). Changes in prescribed accounting, options, thresholds, and exclusions make it difficult to compare many variables that bear the same name and have different meanings on the tax form from one year to the next.

The 1987 Panel of Family Data

Pervasive changes in the IRC in 1986 and the need to identify the substantial mobility of tax filing units in the income distribution led to a Panel of tax returns sampled in 1987 and continuing indefinitely. The panel sample contains clusters of returns in demographic families, rather than tax filing units (Hostetter et al., 1990). Panel data require a concordance of identical variables measured in successive

years. The panel data also require a discordance that warns users about the pitfalls of comparing identical lines on the tax form in different years.

Archival Documents

The U.S. Treasury and the Congress' Joint Committee on Taxation use the Tax Model to estimate revenue and distributional effects of tax proposals included in the US Budget for future fiscal years. Their analyses concentrate on the most recent data collected. In December 1994, the Treasury and Joint Committee analyzed the Tax Model for 1993. Interest in historical data becomes increasingly limited as time passes. The pressure of current legislative demands makes it difficult to maintain access to older data. These pressures extend to SOI whose focus is always on the year for which tax data are processed.

The result is that processing instructions, data dictionaries, and other information are continuously revised. Archival versions are not consistently identified and preserved. Archival documentation is not maintained on electronic media. SOI is currently creating the concordance of variables for the 1987 Panel. Most outsiders would guess that creating a concordance is a trivial task. In fact, the task is technically difficult because of deficiencies in electronic documentation.

■ What is a Desktop Metadata System (DMS)?

Data are the objects of statistical analysis. Metadata are information about the data. Information pertains to all phases of a statistical research project:

- *Design* governs data collection.
- *Execution* of the design captures data, including measurement error and processing error.
- *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.

The discussion that follows will elaborate the distinctions between data and metadata.

An Integration Tool

DMS integrate documentation pertaining to all phases of the science behind statistical data. 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. Part of the problem has been the volume of relevant material. Part of the problem has been the medium containing the material -- some information is text, some is numerical, and some is graphic. Part of the problem has been the timing of "documentation" activities -- typically public documentation has been undertaken after preliminary outcomes from the data are known. Part of the problem has been that large data collection activities involve a team of people, none of whom commands an encyclopedic knowledge of all the processing decisions and outcomes that are pertinent to the complex statistical questions that can be posed to the data.

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 SOI and the Treasury's Office of Tax Analysis (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 [3].

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. To create an information system from these capabilities four steps are required:

- 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.

These same principles apply to other data.

Conclusions

DMS collect information necessary to understand data. DMS encompass the scientific design for data collection and documentation of that collection. DMS can integrate a variety of electronic media into a coherent information resource. DMS can deal with images, databases, spreadsheets, and documents. They can provide bibliographic control on versions and a capability for updates.

Although metadata are documentation, the usage of that term is narrowly construed to mean a static information system that is limited by past concepts of how to compile and disseminate the information system. Documentation compiled after data collection implies inability to capture material pertinent to the process of data collection. It also implies a system that fails to incorporate knowledge created by analysis of the data. Documentation that is conceived of as a publication lacks the facility of logical organization and rapid retrieval that can be attained in databases.

DMS can be compiled concurrently with the several phases of scientific data collection. DMS can comprise a number of software capabilities. And DMS can be constructed to accept revisions and extensions of scope.

DMS Links to Other Text Databases

Table 1 points out that tax analysts use the *Internal Revenue Code* and the *Income Tax Regulations* as a desk reference. They are available as a text da-

Table 1.--Materials Used to Guide Analysis of Individual Tax Return Data

Phase of scientific effort	Datatype	Metadata	Access
Design	Book	*Package X [Year]*	Published
		SOI Bulletin	Published
		SOI Individual Income Tax Returns	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 check	Restricted
Outcomes	Spreadsheet	Analytical tables	Published
Analysis	Book or database	*IRC and Regulations*	Published
	Book	*SOI Bulletin*	Published
	Documents	Miscellaneous methodology reports and papers	Published & Ephemera

tabase (using the folio software). Ideally, a DMS should integrate these two systems, facilitating cross-references between variables in the dataset and the statutory language and administrative code that govern tax compliance.

DMS Provides a Concordance to Other Statistical Data

Pechman's (1985) earliest use of the Tax Model illustrates another capability that can be contained in a DMS. The Tax Model contains little data about sales and excise taxes paid; however, these taxes clearly burden taxpayers. Pechman appended simulated sales and excise tax payments to individual tax returns to arrive at a more realistic representation of tax burden; that practice continues today (Cilke and Wyscarver, 1990; Congressional Budget Office, 1987).

The simulation requires that analysts define the closest analog to a "total income" variable that can be computed in both the Tax Model and the Consumer Expenditure Survey. Once the analogy is defined, it is important to document it in a DMS. The analogy is then available for use by other analysts (who would otherwise face the daunting problem of understanding two complex data collections). When the analogy is documented, it can be studied and criticized by the general public.

When DMS containing similar semantic structures exist for both the Tax Model and the Consumer Expenditure Survey, it is possible to combine them.

Analogies can be systematically documented. The synergy of this combination is ease in creating analogies to variables embedded in other scientific design.

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 et al., 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 [4].

- *Archival versions are not identified.*--The on-going attempt to compile a concordance of variable names used in the Tax Model between 1987 and 1993 has been more difficult because archival versions of electronic documents are not identified at SOI. 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.*--The need for this feature is clearly established in the previous section.

- *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 [5]. 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 dataset expands as the collection is replicated over time and on different samples.

- *Computerized searches of common language descriptions are needed.*--Materials that were gathered for a prototype DMS for the Tax Model sample of 1992 returns InfoSys comprised 1,500 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 con-

form to the order in which tax forms are completed or filed. The labeling 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 materials 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 U.S. 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.

- *Automation would facilitate 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:

  - Additional variables are captured from the tax form; conversely, variables used in the preceding year are deleted.

  - 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.

  - 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 different Tax Models.

All of these adjustments 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 is 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.)

How is the Desktop System Implemented 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 several kinds of material — a relational database, documents, and images. (See Table 2.)

Table 2.-- Architecture for *InfoSys*

DATA			
	Electronic documents obtained from SOI	Images of tax returns 1992	Tables capturing relationships between variable names, forms, and aliases established in SOI processing
SOFTWARE			
	WordPerfect5.1	WordPerfect5.1	Paradox4.0
CAPABILITY			
	Search text	Display form and annotations	Query-by-example Pointers to documents Index captured from documents
	Print	Print	

Because data are organized as arrays for statistical processing, they can be thought of as tables with rows and columns. The rows reflect entities whose attributes are shown in the columns. Labels for rows and columns are essential to linking data and manipulating attributes for statistical computations. Obviously, labels must be unique, or results will be ambiguous. It is also essential to have unique names for tables, or other data objects, so that the desired arrays can be retrieved and operated on. Thus, an inventory of all labels used must appear in the information system.

Labels often are abbreviations or arbitrary combinations of ciphers. To be useful, each label must be unique. 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 [6].

SOI annotated tax forms with the label of each variable that was captured in the editing process. These notations were squeezed onto the tax forms, often in odd places. The irregular placement of notes and the complexity of the IRS forms led us to scan the documents, producing graphic images. The image can not be searched by widely available computer software. However, users are familiar with the forms (and related instructions). That familiarity makes the images an invaluable reference tool.

Implementing the InfoSys Prototype

Several objectives created priorities for the assembly of 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 developed during data collection.

- Establish bibliographic control of both published documents, electronic documents, and the database used in InfoSys.

InfoSys uses a relational database as a keystone that supports and facilitates access to other electronic capabilities. The relational database was derived from text and displays logic implicit in those sources. It is easy to count repeated instances in the database and exploit a uniform nomenclature in organizing material that has logical similarities. The database permits easy identification of aliases used for the InSole variables. It records the process of information flow from the tax return, through administrative processing, and the capture of tax returns by SOI. References to images of tax returns, titles of documents, and word-processing files are embedded in the database.

InfoSys relies on seasoned, off-the-shelf software. This strategy assures:

- client-users who know how to manipulate the software,
- reasonable cost and widely available software, and
- error-free operation of the software.

InfoSys operates with modest capabilities (IBM-286, less than 20 mbytes of hard disk data storage, and less than 2 mbytes of memory). The database software, Paradox, was chosen because of its interoperability with spreadsheets (particularly QuattroPro) and database servers using SQL [7].

The four man-months of professional time used to develop the prototype were concentrated on organizing information and understanding the scientific design and processing. Few "applications" were programmed to reduce the level of Paradox knowledge that is required of users or enhance the presentation of retrievals. This strategy is also consistent with the principle that the InfoSys relational database management system (RDBMS) be portable [8].

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.

The Level 1 Prototype

Table 3 gives a bird's eye view of the database. A complete map of the InfoSys Level 1 prototype is provided by Exhibits 2 and 3. Exhibit 2 describes the relations contained in the relational database. Relations are displayed in a top-down order, from general information about the metadata to specific information about the data objects, principally InSole. Exhibit 3 lists and describes attributes for the relations shown in Exhibit 2. These system relations provide a "data dictionary" for all elements in the InSole file. They also link those elements back to labels for data captured from the administrative files (RTF) by SOI processing. References to the chapters of the Editing Manual that deal with each form or schedule are provided. Those chapters can be retrieved by calling the appropriate document in WordPerfect processing. Similarly, each tax form processed can be retrieved by calling its image from a second WordPerfect document. Users can begin their search for information in one of three ways:

- naming an InSole element,
- naming a tax form, or
- initiating a text search of element descriptions.

Table 3.--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 for bibliographic and file control	
DOCDIRWP	WordPerfect documents	Shows DOS filename and creation date to establish version control
PUBLICAT tions,	Publications	Bibliographic description of insole relevant publica- memoranda, and documents
	C. 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

The prototype reveals labels used in processing. It provides immediate access to codes created for analysis. It reduces the search time for information in unpublished memos, documents, and processing manuals. Because the latter are electronic documents, text can be quickly searched. However, the text searches within the database have the advantage that each database table reflects relationships among variable names at different stages of processing, applied to different levels of aggregation, or relating names for the same item of information, as it passes from one tax schedule to another. Exhibit 1 displays a schematic relationship between the processes that generate the individual income tax return sample and the corresponding metadata in the InfoSys DMS (column 7). The prototype includes the tax forms with annotations of edited fields. This facility makes it possible to locate many attributes through a visual scan. (Were generic tax forms available in paper form, they would not show the location of edited variables.)

■ What Have we Learned?

I can comment definitively about development of the InfoSys. My assertions about the value and performance of DMS are speculations and need to be tested. This requires SOI to create a production process that will maintain, correct, and revise the InfoSys.

Development

- *Images.* -- Available scanning technology generates satisfactory substitutes for complex paper documents, albeit with a cost in disk storage requirements. Optical character recognition of documents is not cost-effective for a DMS, because extensive editing of the product by experts is required. The need to proofread conversion of the scanned image, of even printed pages published on high quality paper, makes conversion of scanned documents inappropriate for a DMS at this time.

- *Spreadsheets.*--Borland has a common data server for spreadsheets and its Paradox database. This makes incorporation of appropriately-structured spreadsheets into a database an easy task. A spreadsheet was the source of the *92derlmf* table [9].

- *WP documents.*--WordPerfect (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 in directories, to permit users to read or print chapters. The principal problem with the latter use of the document is that WP5.1 does not incorporate a "read only" mode. Thus, each document must have a time stamp for the definitive version included in the database. That time stamp appears in the database table *Docdirwp.*

- *Portability.*--InfoSys was designed to be exported to other PC environments with Paradox4.0. The InfoSys prototype was installed at the U.S. Treasury and SOI in May 1994. Duplicating and-moving the entire DMS created no problems.

Value Added to Existing Documentation for InSole

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 an inventory of every column appearing in every table. The content of these tables is vital to understand the scope of the database. (Content is displayed in Exhibits 2 and 3.) Furthermore, the creation of these tables controls nomenclature and reduces proliferation of different names for columns that contain identical information.

- The *publicat* table was constructed on sound principles for bibliographic databases. In addition to the citation table, an authority list of authors needs to be added.

- The *docdirwp* table creates a version control for all the WP documents included in the information system. It also ensures that inadvertent changes do not enter the document system. It assures that backup copies exist.

- The *form-edi* table 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.

- The *92derlmf* table was taken from a spreadsheet specialized for directing SOI's PRISM staff to designing the data capture operation. It contained many duplicate rows that were difficult to understand and incompatible with a relational concept. Many of these duplicate rows have been eliminated. The table can be much more efficiently searched than its Lotus predecessor.

- The *bus-farm* table establishes an important distinction between similarly labeled fields that may refer to tax return aggregates or enterprise detail. In a relational conception this difference in unit of analysis should be mirrored in separate tables.

Quality of a DMS

Several key benefits of the DMS bear repeating:

- *Timeliness.*--A DMS can be generated concurrently with all phases of data collection and dissemination.

- *Institutional memory.*--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.

- *Consistency.*--DMS can assure uniform use of names in different phases of data development and can create authority lists for all aliases used. We discovered several weaknesses in existing documentation that can be overcome in a DMS. Repeated specifications in existing SOI documentation were not always consistent. Confusing differences in describing the same material in different documents exist. In addition, descriptions of elements do not adequately classify elements in InSole in terms that relate to analysis of tax returns. The descriptions are not precise enough to enter a text database on tax law and regulations.

- *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.*--The 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.

IRS and users of the DMS share responsibility to adhere to scientific and research use. Restricted documentation will need to be distributed to licensed/bonded research users. Techniques for such access to sensitive information are discussed in National Research Council (1993).

Flexibility

Tables in the relational database can easily be reorganized. Columns are easily renamed. This flexibility can assist in defining classes of similar attributes. Columns can be added or deleted from tables, to minimize the complexity of queries and to conform to the context from which analysts wish to view the data.

The principal constraint on reorganization is the fact that the system tables must also be amended. Failure to do so would leave the map of the information system in an inconsistent and incomplete form. (Mark and Roussopoulos, 1986, offer an approach to automating the updating of system tables that could be accomplished within a Paradox application.)

Weaknesses in current InSole documentation

- *Accuracy* --Version control: SOI does not now have bibliographic control of the archival versions of critical electronic media. InfoSys creates a mechanism to establish that control.

- *Logical complexity of InSole* -- Information on business entities, persons, and tax return aggregates are assembled in the InSole data file. Neither WP documents nor InSole descriptions adequately revealed the process for dealing with multiple forms. The InfoSys distinguishes tax return level data and data on business entities. We believe the distinction will reduce confusion that has existed in the past. Similar procedures can be applied to other instances of multiple forms.

■ What is Gained by Extending DMS to Other Statistical Data?

Application to other data is feasible. The process that created InfoSys can be used to create a DMS for another tax year.

Extensions of InfoSys have great value

1993 Tax Year: We believe the costs of replicating and extending the InfoSys prototype will be smaller than the activity to date. Many parts of the InfoSys can be generated as design of processing is created. SOI staff will be more knowledgeable about the information captured and will be able to execute capabilities that were daunting for outsiders. For example, consistency checking documents can be indexed by elements, and the indices can be incorporated into relationships describing Tax Model variables. David and Robbin (1992) establish that metadata can accelerate the development of DMS for scientific designs that are near replicates, at a fraction of the effort involved in the original.

We also believe that using the prototype as a template in future SOI/I processing will reduce the time spent in preparing specifications and adjusting to changes in specifications for the data collection program from year-to-year. At the same time consistency will be assured by the relational architecture of the metadata base in InfoSys.

■ Acknowledgments

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 1991, 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 U.S. Treasury OTA.

■ Footnotes

[1] The Individual tax return sample is the first instance of probability sampling of administrative records in the Federal government.

[2] Confidentiality restrictions on dissemination of tax returns for statistical purposes are enacted in the Tax Reform Act of 1976. They result in four classes of users: Employees of the Treasury Department and selected Special Committes of Congress, who have security clearance to make use of the data for drafting legislation and revenue estimating purposes; a limited number of other Federal agencies, whose staffs have ac-

cess to selected tax data, as specified by law; employees of states, who exchange data with IRS and are privileged to examine Federal tax returns; and the general public, who can only access unidentifiable tax information. The latter group includes most academic researchers and consulting firms that assist governments in estimating the impacts of legislative proposals on their revenue and its distribution. The public version of the Tax Model contains fewer variables and is a proper subset of the version used by Federal analysts to evaluate revenue and distributional impacts of Federal legislative proposals.

[3] These requirements can be met by query languages associated with relational databases. The naming of variables is controlled by the database, and the logic of data manipulation is easily parsed. Lastly, Codd's insistence on data integrity assures a standard of performance that is not present in lower level programming. See Date (1988).

[4] Microdata on errors are putatively the data most sensitive to disclosure. Thus, any public data would need to be aggregated (National Research Council, 1993).

[5] This is being done by the *Panel Survey of Income Dynamics*, the *National Longitudinal Survey*, the *German Socio-economic Panel*, among others. It was done for the *Survey of Income and Program Participation* by the University of Wisconsin; it is not done by SOI.

[6] For example, information included in the attribute "DIST" does not lend itself to a uni-dimensional code. For that reason a special relationship, STATE, was generated to display the meanings of codes given in that attribute.

[7] PARADOX also has capability for storing large text blocks and images (Version 4.5). In addition, Paradox has interfaces to other databases (Dbase) and proven capacity to operate in a LAN.

[8] PARADOX application programming is extremely powerful and includes "programming-by-example."

[9] Spreadsheets can, however, easily pose problems for the database. The "Analytical Tables" produced to use in connection with Insole displayed data in matrix form, but did not contain a unique key for each entry. Attention to the design of the spreadsheet -- or capturing the aggregates into the database in the first instance -- would have avoided this problem.

■ References

Brooks, Camilla and 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.

Cilke, James and Roy A. Wyscarver (March 1990), The Treasury Individual Income Tax Model, U.S. Treasury, Office of Tax Analysis.

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

Congressional Budget Office (1987), *The Changing Distribution of Federal Taxes: 1975-1990*, Congressional Budget Office, U.S. Congress.

Date, C. J. (1988), *An Introduction to Database Systems*, Reading MA: Addison-Wesley, 444.

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.

David, Martin H. (1991), "The Science of Data Sharing: Documentation," In Joan Sieber (ed.) *Sharing Social Science Data: Advantages and*

Challenges, Newbury Park, CA: Sage Publications, 91-115.

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.)

Hostetter, Susan; Czajka, John; and Schirm, Allen (1990), "Choosing the Appropriate Income Classifier for Economic Tax Modeling," *Proceedings of the Section on Survey Methods, American Statistical Association.*

Jabine, Thomas B.; King, Karen E.; and Petroni, Rita J. (1990), *Survey of Income and Program Participation: Quality Profile*, Bureau of the Census, U. S. Dept. of Commerce.

Mark, Leo and Nick Roussopoulos, (1986), "Metadata Management," *Computer*, 26-36.

National Research Council (1993), *Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics.*

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

Witte, John F. (1991), The Tax Reform Act of 1986 A new era in tax politics, *American Politics Quarterly* 19:438-57.

Internal Revenue Service Documents

Statistics of Income -- Individual Income Tax Returns [Year] (Publication 1304), Washington, DC: GPO (annual).

IRS, Internal Revenue Administrative Processing Manual (unpublished/restricted).

Statistics of Income, Memorandum: Weighting and Variability Specifications Package for the SOI 1992 (unpublished).

Statistics of Income, Specifications for Consistency Testing and Related Processing of Tax Year 1992, *Statistics of Income Individual Income Tax Return Records*, Forms 1040, 1040A, and 1040EZ (unpublished). ■

EXHIBIT 1 -- Process and Information: SOI Individual Tax Returns, 1992

Agent	Data generating system: Operation	Process definition	Object	Info system - metadata: Label	Level 1 Mode	Level 1 Object	Level 2 Object	Control & Monitoring process
1	2	3	4	5	6	7	8	9
Tax-payer	Return preparation							
	precursors	TaxpayerX	Generic Return	Form/ Schedule	I1	IMAGE		
					R1	FORM-EDI		
		Regulations, Code					REGS IRC	
	outcome		Return document					
IRS	Administrative processing							
	precursor	IRS Manual						
	outcomes		RETURNS TRANSACTION FILE (RTF)	Form/ Schedule				Audit TCMP
	precursors	Audit manual TCMP manual						
	outcomes		Audited return					
			TCMP DATA				TCMP TABLES	
SOI	Statistical processing							
	Sampling	Publication 1304						
	Data capture		PRISM database system	Tables Attributes	R1	FORM-EDI		
					R2	92DERLMF	ERROR	Error rate
		Editing Manual			D1	WP documents		
	Consistency checks	Consistency .. , Editing Manual			D2	WP documents		
					R3	INSOLEDF		
	Transformations	Consistency ..			D3	AppendixG		
					R4	APPG-CODE		
					R5	APPG2		
					R6	SECTION3		
					R7	BUS-FARM		
					R8	SOURCE		
					R9	STATE		
					R10	DOCDIRWP		
	Validation	Validation specifications	INSOLE					
	Tabulation, Analysis		ANALYTICAL TABLES, etc.		R11	PUBLICATIONS	TABLES	

Key
I images
R Database relations
D WP documents

Exhibit 2.--Relations in the InfoSys Database

Relation: Relation
Order: 1
Attributes: 9
Rows:

Key(s): Relation Reference: InfoSys
Entities: Meta-relation
Linkage: 1:M
Semantic principle: Keywd: InfoSys relations: structure and links
Semantic principle:
One row per relation in the information system. The attribute "semantic principle" describes the logic that generates the relation and exceptional features of that logic.

Relation: Attribu3
Order: 1
Attributes: 4
Rows:

Key(s): Relation,Attrib Reference: InfoSys
Entities: Meta-attribute
Linkage:
Semantic principle: Keywd: InfoSys attributes, column labels
Semantic principle:
One row for each attribute in each relation of the information system. Identical names are used when the meaning of the attribute is identical across relations. For example, "element" always connotes a variable of the Insole data file. Attributes with identical names can be used to link relations, with some exceptions. "Description" is a common language meaning for the combination of attributes that defines a unique row in the relation. Because that combination varies across relations, it will not provide a linkage between relations. Similarly, "countnum" is a serial that is unique to each relation.

Relation: Source
Order: 1
Attributes: 4
Rows: 20

Key(s): Source Reference: Section 3, CONSISTENCY
Entities: Meta-value
Linkage: 1:1
Semantic principle: Keywd: Meanings,attributes in Section3,Appgcode
Semantic principle:
One entry for each abbreviation used in the "Source,Note,Line Reference" attributes of Section 3 and Appgcode.

Other attributes of the Section3 relation are defined with no value shown.

Relation: Form-edi
Order: 2
Attributes: 9
Rows: 59

Key(s): None Reference: InfoSys
Entities: Forms or Prism Table
Linkage: M:M
Semantic principle: Keywd: Prism Tables or Individual return forms
Semantic principle:
One or more rows for each IRS form used in Individual tax returns. In addition one row is included for Prism tables not labelled with a form/schedule. (The attribute "Form/Schedule" is blank in that instance.) Sequence is "attachment sequence" as specified in EDIT MANUAL. "Sequence, Form/Schedule, Table" gives a natural order to the relation. Rather than repeating rows for 1040A, 1040-EZ which relate to the same Prism table, the relation displays Form 1040 (and sequencing for Form 1040A). Associations among related forms is indicated in the attribute "Table family." The combination "Sequence, Edit order" has a unique value for each row. Negative values are assigned to "Edit order" for forms that are not edited.

Exhibit 2.--Relations in the InfoSys Database--continued

Relation: Appgcode
Order: 3
Attributes: 10
Rows: 327
Key(s): Element
Reference: Appendix G, Sec. 3 CONSISTENCY
Entities: Attribute, Insole
Linkage:
Semantic principle: Keywd: Coded Insole attribute
Semantic principle:
General: One entry for each item whose label is AA[..]. A is any letter of the alphabet and "[..]" indicate any combination of letters and numbers.
Exception: Includes flags for spcific tax forms of the form Ai[i]. Constructed as the intersection of information in the "Codes" table of Section3, CONSISTENCY ..., and "Appendix G" in the same document.

"Form/Schedule" is only shown for an arbitrary subset of attributes, primarily "Form 1040." Corresponding line references relate to Forms 1040/1040A/1040-EZ in succession. "Schedule C" is entered in instances where the meaning is both Schedule C and Schedule F The interpretation is made clear by the attribute "Header ."

Relation: Appg2
Order: 3
Attributes: 7
Rows: 1233
Key(s): Element
Reference: Appendix G, CONSISTENCY
Entities: Value, Insole attrib
Linkage: 1:M
Semantic principle: Keywd: Code values for Appgcode, Insole
Semantic principle:
Several forms of information are included in the rows. Each element of Appgcode matches many rows in this relation. Two major forms of information are shown:
1. Values of a numeric code assigned linked to a definition of the value.
2. Minimum and maximum values for the attribute with reference to sources of further information about the meanings of those values.

Relation: STATE
Order: 3
Attributes: 8
Rows: 71
Key(s): State/PScode, Dist
Reference: Appendix G, CONSISTENCY
Entities: Value, Insole attrib
Linkage:
Semantic principle: Keywd: Districts mapped to States
Semantic principle:
One row for each geographic area assigned to an IRS district. The districts have numerical codes. The district may include several political jurisdictions. A single state may include several districts.

The mapping of districts to service centers is also shown together with code designations for the centers.

Exhibit 2.--Relations in the InfoSys Database--continued

Relation: Section3 Order: 3
Attributes: 8
Rows: 969
Key(s): Element Reference: Section 3, CONSISTENCY
Entities: Attribute, Insole
Linkage:
Semantic principle: Keywd: Money amount, control element: Insole
Semantic principle:
General: One row for each Insole element whose label is Ai[i..]. A is an initial letter followed by one or more integers, i or ii or ii.. .
Exception: Flags showing the presence of dif ferent forms with labels of the form "Ai[i] are included in the relation Appgcode.

The source of the information is Section 3, CONSISTENCY ... .

Attributes for farms and businesses are shown for the aggregate of such entities on the tax return. For information about particular entities see the relation Bus-Farm.

The attribute is related to particular documents through the attributes "Form/Schedule and Line reference." The attribute "Sub-schedule" contains information that complements "Description."

Relation: Bus-farm Order: 3
Attributes: 9
Rows: 217
Key(s): Element,entity Reference: Section 3, CONSISTENCY
Entities: Enterprise, Insole
Linkage: M:1
Semantic principle: Keywd: Farm, Business enterprise amounts: Insole
Semantic principle:
One row for each enterprise entity. Up to three business entities and two farm entities. The attributes provide entity-level information corresponding to the combined information in Section3. However , some entities are not coded so that entity-level information does not necessarily aggregate to the total shown on the return in Section3.

Relation: 92derlmf Order: 3
Attributes: 13
Rows: 1362
Key(s): P.table,new,P.n Reference: SOI spreadsheet
Entities: Attribute, Prism
Linkage:
Semantic principle: Keywd: Prism attributes used in edit
Semantic principle:
One entry for each attribute used in the editing of tax returns by the Prism processing. Attributes may be transferred from the R TF or entered directly from returns.

Exhibit 2. --Relations in the InfoSys Database--continued

Relation: Insoledf
Order: 4
# Attributes: 4
# Rows: 1155
Key(s): New Reference: CONSISTENCY (Detroit)
Entities: Attribute, DCC
Linkage: 1:1
Semantic principle: Keywd: Element-P.new relation
Semantic principle:
One row for each "New" attribute used in DCC CONSISTENCY .... "Insole" contains label used in Insole (Y ear ?). If "Insole" is blank, "New" is the Insole element label. The relation between "New" and "Prism name" is in 92derlmf.

Relation: Docdirwp
Order: 4
Attributes: 7
Rows: 199
Key(s): Filename Reference: InfoSys
Entities: WP document
Linkage: 1:1
Semantic principle: Keywd: Filename for WP documents
Semantic principle:
One row for each document and one row for each sub-directory that or ganizes documents by phase of processing.

Relation: Publicat
Order: 4
Attributes: 15
Rows: 0
Key(s): Author,Title Reference: InfoSys
Entities: Publication
Linkage: 1:1
Semantic principle: Keywd: Bibliographic database
Semantic principle:

Exhibit 3.--Attributes in the InfoSys Database

2/20/94 Attributes of InfoSys Page 1

Relation	Field Name	Field Type	Description
92derlmf	Comments	A23	
92derlmf	Line reference	A18	Label on document line, section
92derlmf	New	A15*	Alias used in Detroit processing
92derlmf	Pos/neg	A9	(+/-) indicates amounts that may be signed.
92derlmf	Positions	N	Number of bytes in the data field
92derlmf	Primary key	A4	Attribute used in creating a unique key for table
92derlmf	Prism Table	A24*	Name of relation containing the attribute
92derlmf	Prism name	A38*	Name assigned by RTF and SOI to attribute
92derlmf	Screen line	A24	location of entry on Prism edit screen [ctst-pri].
92derlmf	Seqnum	N	Serial
92derlmf	Source	A16	Section refers to CONSISTENCY
92derlmf	Source-note	M1	
92derlmf	Type	A6	
Appg2	Association	N	Serial linked to element group in Appgcode
Appg2	Definition of Outcomes	A	Meaning attaching to value or interpretation rules
Appg2	Detailnum	N*	Serial ordering values for the code
Appg2	Element	A15*	Name of Insole attribute
Appg2	Maximum	A22	Largest value of code (where applicable)
Appg2	Special Note	A37	
Appg2	Value or Minimum	A24	Cipher for meaning; alternatively smallest value
Appgcode	Association	N	Serial that groups repeated/aggregate elements
Appgcode	Countnum	N	Serial order Appendix G, CONSISTENCY (15 blank=NA)
Appgcode	Description	A111	Meaning of attribute
Appgcode	Element	A13	Name of Insole attribute
Appgcode	Form/Schedule	A8	Name of IRS document associated with returns
Appgcode	Header	A23	Partitions codes into Business and Non-business
Appgcode	Line reference	A19	Label on document line, section
Appgcode	Note	A2	NV=No code Values. NW=NorthWest corner of form.
Appgcode	Seqnum	N	Serial order Section 3, CONSISTENCY (14 blank=NA)
Appgcode	Source	A15	Origin/generating algorithm. See source relation.
Attribu3	Description	A50	Meaning of attribute
Attribu3	Field Name	A25	Label for attribute in Infosys
Attribu3	Field Type	A5	Attribute datatype
Attribu3	Relation	A8	Label for relation in Infosys
Bus-farm	Countnum	N	Serial maintaining order of Section3, CONSISTENCY
Bus-farm	Description	A110	Meaning of attribute
Bus-farm	Element	A7	Name of Insole attribute
Bus-farm	Entity	A9	Business#/Farm# = serial attached to Sch.C/Sch.F
Bus-farm	Form/Schedule	A12	Name of IRS document associated with returns
Bus-farm	Line Reference	A49	Label on document line, section
Bus-farm	Note	A9	
Bus-farm	Source	A9	Origin/generating algorithm. See source relation.
Bus-farm	Sub-Schedule	A44	Class of data elements, or location within Form/Sc

Boldface denotes key attributes in each relation.

Exhibit 3.--Attributes in the InfoSys Database--continued

2/20/94			Attributes of InfoSys	Page 2
Relation	Field Name	Field Type	Description	
Docdirwp	Bytes	N	Size of document in bytes	
Docdirwp	Countnum	N	Serial to order files by sub-directory and date	
Docdirwp	Date	A10	Last update to archived file	
Docdirwp	Directory of c:\irs	A41	Path to WordPerfect document files	
Docdirwp	Extension	A5	DOS extension of WordPerfect document name	
Docdirwp	Filename	A10	File name for WordPerfect document	
Docdirwp	Time	A8	Time of last update to archival document	
Form-edi	Description	A65	Approximate title of document form or schedule.	
Form-edi	Edit order	N	>0= Processing order in Prism edit. <0= unedited.	
Form-edi	Edit screen	A42	Label. See PRISM CONSISTENCY for picture.	
Form-edi	Form/Schedule	A12	Name of IRS document associated with returns	
Form-edi	Manual-WP	A23	Name of WP document in [edit] directory.	
Form-edi	Sequence	A6	Document sequence specified on the form	
Form-edi	Table	A24	Prism Table name corresponding to the document	
Form-edi	Table family	A45	Indicates suffix added to Prism Table for update.	
Form-edi	TaxpayerX	N	Reference to TAXPAYERX; read as <volume>.<page>.	
Insoledf	Cobol	A6	COBOL attribute description	
Insoledf	Insole	A21	Insole element, when different from "new"	
Insoledf	New	A12	Alias to Prism name (for DCC CONSISTENCY tests)	
Insoledf	Width	A10	Number of characters in attribute	
Publicat	Author	A63*	Authors identified by surnames and initials	
Publicat	Caldate	D*	Month and year of publication	
Publicat	Comment	M100	Abstract of interest to Insole users	
Publicat	Journal	A50	Serial publication title	
Publicat	LC_number	A20	Library of Congress number	
Publicat	Page1	N	Start page of article, excerpt	
Publicat	Page2	N	Ending page, article, excerpt	
Publicat	Part	N	Part of multi-volume title	
Publicat	Place	A20	Address of publisher	
Publicat	Publisher	A20	Publisher, or source of document	
Publicat	Seqno	N	Acquisition number, inscribed on archival copy	
Publicat	Specialty	A18	Detail classification of keyword	
Publicat	Subject	A18	Keyword, used with specialty	
Publicat	Title	A125	Title of article or book	
Publicat	Volume	N	Serial volume number	
Relation	# attributes	N	Number of columns in the InfoSys relation	
Relation	# rows	N	Number of rows in the InfoSys relation	
Relation	Entities	A15	Unit that replicates within the relation	
Relation	Key(s)	A20	Attributes forming unique identifier for each row	
Relation	Linkage	A10		
Relation	Order	N	Serial ordering presentation of relations	
Relation	Reference	A40	Source of information	
Relation	Relation	A8	Label for relation in Infosys	
Relation	Semantic principle	A40	Principles generating the relation, & exceptions	

Boldface denotes key attributes in each relation.

Exhibit 3.-- Attributes in the InfoSys Database--continued

2/20/94			Attributes of InfoSys	Page 3
Relation	Field Name	Field Type	Description	
STATE	DGROUP	A6	"Sampling group code." See relation Appgcode.	
STATE	DIST	A4	Numeric code for district	
STATE	District	A15	Name of District	
STATE	Note	M7	Special rules for APO/FPO addresses	
STATE	STATE	A20	Name of state/country jurisdiction	
STATE	SVCCTR	A6	Abbreviation for Service Center name	
STATE	Service center code	A4	Numeric code for Service Center	
STATE	State/Alpha Postal code	A17	Numeric state/country code; Postal alpha code	
Section3	Countnum	N	Serial maintaining order of Section 3, CONSISTENCY	
Section3	Description	A110	Meaning of attribute	
Section3	Element	A6	Name of Insole attribute	
Section3	Form/Schedule	A12	Name of IRS document associated with returns	
Section3	Line reference	A23	Label on document line, section	
Section3	Note	A9		
Section3	Source	A9	Origin/generating algorithm. See source relation.	
Section3	Sub-Schedule	A44	Class of data elements, or location within Form/Sc	
Source	Attribute name	A20	Attribute of Section3, Appgcode	
Source	Description	A205	Meaning of attribute, attribute value	
Source	Section 3 Heading	A100	Title for Section 3, reference key	
Source	Value of attribute	A6	Value of attribute used in Section3, Appgcode	

Boldface denotes key attributes in each relation.