THE BANCA D'ITALIA'S ACTIVE STATISTICAL META-INFORMATION SYSTEM.

Vincenzo Del Vecchio
Banca d'Italia, SISC

March 2001

INTRODUCTION

Statistical information provides wide-ranging support for the Bank of Italy's institutional activities. User needs are satisfied by mean of statistical surveys on supervised intermediaries, data extracted by internal EDP operative procedures, data coming from Italian and international counterparts. Moreover, processed statistical information is disseminated to many institutions, to survey participants and to the public.

The growing dimensions and variety of this information system,[1] the continuous evolution of contents, and the number of natural and legal persons involved in its operation have required considerable organizational, technological and methodological measures.

The organizational measures serve to coordinate the needs of the institutional functions, define the information area of common property and the private property areas of the specific institutional functions and provide joint administration of common concepts and data.

The technological and methodological measures serve to ensure the effectiveness and efficiency of the system. They are the primary focus of this paper.

METADATA AND THE SOFTWARE ARCHITECTURE

The Bank of Italy strategy in developing statistical applications has been based for a long time on software packages and their reuse.

Software packages, prevalently made in-house, are usually function specific (e.g. extraction, collection, control & cleaning, storing, processing, searching and inquiry, dissemination, analysis, publication, …) but generalized with regard to the data. A unique software set is able to process independently many different sets of statistical data, in the same way as different production lines in a factory.

An application requested by a user is a software product obtained reusing one or more packages to process the specific set of data.

The ability to apply the same function to different data (generalization) depends on parameters that drive software functions, known as metadata.[2] They describe statistical concepts, statistical data and processing rules.

[1] To give an idea of the order of magnitude, the information system contains more than 100 thousand array data definitions and 100 thousand time series definitions, corresponding to about 3 billion data records, which are growing at a rate of about 500 million records per year.

Modeled like data, metadata are stored in the same environment as data and form a large metadata system that is the fulcrum of the whole statistical information system.

In fact, metadata are active, because they drive the software functions, are the means of specifying and document data and operations, are available to all kind of human users, and ensure a fast response to the evolution of needs.

As a matter of fact, running an information system based on active meta-information has been found to bring many benefits:

- self-documentation (active metadata are intrinsically correct and updated, and document information system operations for users, administrators and EDP people).

- user autonomy (administrators are able to prepare and modify survey definitions and processing without any - or with only a very little - EDP department work, final users are able to search and read data without administrator or EDP help)

- time to market and cost reduction (implementing and modifying applications mainly involves acting on definitions, and in only a few cases on software, the overall dimension of the software system is reduced because software uniqueness and metadata make it easier to deal with complexity)

The proven value of these benefits has further enforced the trend at the Bank of Italy to store all the information needed by users, including software behavior definitions, outside software in a data base, and at the same time to deal with the resulting growing complexity of the metadata system.

In a specific program without parameters, the behavior of the software cannot be changed without changing the software itself. Software testing ensures correctness of the response once and for ever, but every change in behavior implies software updating and testing.

In software driven by business metadata, the behavior can be changed by the user; consequently, the coherence of the metadata given to programs is of fundamental importance for obtaining correct processing.

It is thus necessary to have a specific package to deal with metadata, with the function of helping the administrator to define, modify, store and retrieve them, control their overall coherence, and manage them in a sort of configuration system, because they can be viewed as software components stored in a data base. Moreover, in a very complex environment, metadata management software must allow both the contemporaneous and independent activity of different administrators and the cooperation of many of them in carrying out the same definition work.

To meet these needs, a package that deals with metadata administration has been built. This package is also generalized and active and deals with metadata just as if they were data. Therefore, it has its own model to describe metadata structures (metadata that describe metadata structures are known as meta-metadata, i.e. metadata structure descriptions). Such a model is also able to describe itself, so that the package can also manage its own metadata.

[2] They are mainly user-defined “business” metadata, not only technical metadata needed for a specific package or data base management system.

Consequently, the system architecture, although suited to specific needs, is very similar in principle to the OMG four-layer architecture, in which every layer is the description of the previous one. The first is the data layer, in this case the statistical data and the second is the data description layer (metadata contents). The third layer is the structure of the data description (metadata model or meta-model), in this case the statistical metadata model, and the fourth is the model to describe meta-models (meta-metamodel), in this case the metadata package model.

COMMON METADATA REPRESENTATION MODEL

Statistical active metadata can be divided into parts. The first is used to drive many software packages and can be called the "common metadata area", each of the other parts drives a single package, so they are "package-specific metadata".

This is the origin of the so called "statistical dictionary", a metadata system born to unify at a conceptual level the administration of metadata needed by different packages (and of course different surveys).[3] The main purpose of the statistical dictionary is to have a unique metadata language and a unique metadata container for the common metadata, vis-à-vis users and administrators. A second goal is to allow an independent evolution of the metadata system and the data processing functions and packages.[4] A transformation system is used to convert the statistical dictionary metadata structure to the structures and formats needed by specific packages.[5]

Package-specific metadata can be thought of as islands related to the common metadata area. They are used to specify typical aspects of a generic processing function (such as extraction, collection, control & cleaning, storing, processing, searching and inquiry, dissemination, analysis, publication, and so on).

The common metadata conceptual layer, i.e. the "statistical dictionary", refers to a representation model that is known as "matriciale" (it means matrix). It can be thought as a multidimensional type model.

The aim of the "matriciale" model is to represent three kind of things, which we can think of as three layers of an onion:

- statistical concepts (internal layer), i.e. every abstraction reusable to define data;
- statistical data (middle layer): definitions (intension) of the data at a conceptual level;
- transformations (external layer): e.g. algorithms to obtain a certain datum (both in intension and extension) using as operands other data available in the system.

The model is based on algebraic concepts. This makes it possible to refer to a powerful and well-established theory.

[3] The set of potentially common-use meta-information is gradually developed with experience and theorization.

[4] E.g. to allow addition, evolution or substitution of packages without changing the conceptual level metadata structure and to promote the integration of in-house and bought-in packages.

[5] Having a common metadata standard also makes it possible to reuse software modules in different packages when the processing functions to be performed are the same (e.g. calculus).

Statistical concepts

The aim of the statistical concepts model is to allow representation of abstractions that we need to define the statistical data.

The main algebraic object used for concept definitions are elements, sets and variables:

- elements are used to represent a single, simple abstraction (e.g. Rome, London, New York, …) .

- sets are used to represent a set of elements (e.g. cities).

- variables are used to represent a set of elements with a more specific associated meaning (e.g. city of residence, city of birth, …).

All three are regarded in a historical context, that is they exist in a particular period and their characteristics can change with time.

They are linked by basic integrity relationships (referring to a generic time instant):

- a set may contain many elements, an element may belong to many sets.

- a variable can have values in one set, in a set many variables can have values.

Relationships are also regarded in a historical context (e.g. the elements of a set can change with time).

A set can be defined in extension (giving the list of cities) and intension (giving the property of the set’s elements, such as “numbers from 0 to 100”).

A set can have an internal structure. Many kinds of structure are admitted. The power set structure is the most general. It is used to represent many elements in the same set, where the elements refer to the same aspect of reality (e.g. “territory”) but at various levels of detail (e.g. “cities”, “countries”, “continents”, “world”). The root element of a power set structure represents the whole (“universe element”), any other element represents a generic subset of the whole.

Between elements of a power set there can be relationships coming from set theory algebra (e.g. union, classification, …). Using such relationships, a multi-level classification structure can be defined inside a power set, or more than one such structure with the relevant mutual relationships.

Classifications can be manual or automatic. The former are directly user-made, the latter are automatically updated by proper software on the basis of user-defined classification criteria and relevant data in the data base (e.g. to classify banks in term of their number of employees).

Multidimensional sets can be obtained by defining a Cartesian product of one-dimensional sets or, in a more complex situation (such as excluding combinations of values), by composing Cartesian products (or combinations of values) with set-type operators (such as union, intersection).

Other relationships can be defined between sets and between set elements.

The “subset” relationship, for example, forces a set to be a subset of another. The subset can be defined only in terms of the superset elements or in terms of their properties.

Relationships between elements belonging to different sets are used to correlate different coding systems, and also to correlate different multidimensional spaces, when different spaces are used to represent the same mental concepts (something that can easily happen between different organizations, surveys or times).

To conclude, the representation of active concepts is used to formalize concepts used in the statistical data and processing. Further, it is used to correlate possible ways of describing the same mental meaning and to allow joint processing and the use of data with different representations.

Statistical data

A statistical datum has a definition (intension) and a representation (extension). Only the former is mandatory: very often a datum is defined before it has an extension (the opposite is not possible in such a formalized environment because, without a definition, no package will process that datum). For a datum, intension and extension must be referenced to each other.

Data definition is made using explicitly defined concepts. Every definition, therefore, refers to concepts. An implicit relationship is stated between different data that refer to the same concept (or related concepts), so the reuse of the same concepts is recommended, because it is a way of declaring relationships between data and allowing joint use.

The basic algebraic object used for data definition is the so-called “statistical function”. Just as all other algebraic functions, it has independent variables that assume values in a set known as “domain” and dependent variables that assume values in a set known as “co-domain”. Both domain and co-domain are in general multi-dimensional. The function associates every domain element with one co-domain element.

Independent variables can be classification variables or time variables. The former are used to specify the groups of statistical units, the latter the time periods.[6]

The domain of the generic statistical function includes many groups and many time periods (array data in GESMES, the EDIFACT standard). Important function subtypes are historical series (one group and many time periods) [7] and cross-sections (many groups and one time period).

Dependent variables represent the kind of information we want to know and can be quantitative or qualitative. Other optional dependent variables are the attributes, used, when needed, to describe properties of the observations.

The extensional form of a statistical function associates every pair “group - time period” with the relevant specific dependent information (however it is obtained).

[6] A time instant is a particular case of a time period.

[7] GESMES/CB (CB is for Central Banks) also uses a multidimensional Key-family to define a homogeneous historical series family (i.e. an array).

The statistical function definition therefore includes the structure specification in terms of variables and the domain specification. If the domain is a Cartesian product, the latter implies the specification of a one-dimensional domain set for each independent variable; elsewhere it may be necessary to define more Cartesian products (or combination of values) and their composition with set operators.

Two kind of domain are specified:

- "definition domain" is the domain in which the statisticians want to know the function, e.g. the groups and dates for which they ask to have observations. The specification is made when the function is defined, typically before having the relevant observations.

- "knowledge domain" is the domain in which the extensional form of the function is really known, e.g. groups and dates whose observations are in the system. It must be updated in parallel with data feeding. In many cases, it is enough to define knowledge domains in terms of a few independent variables, those that actually influence the knowledge of the observations (e.g. the date for periodical surveys, the sender for surveys that collect data from many sources, etc.).

This statistical data model appears to be similar to the conceptual model on which GESMES is based. A high level overview shows that they are constructed on the same principles and that the basic representation structures are similar.

As a parallel to the EDIFACT standards, we find that CLASET covers the matters of the statistical concept definition and GESMES those of the data structure definition, and that although the representation structures can be different, the basic principles are the same.

Transformations

The transformation part of the model is the most recent part.

"Transformation" is thought of as a process that calculates a "statistical function" by applying an "algorithm" to one ore more "operands", which are also statistical functions.[8]

Transformation can easily be represented as relationships between the calculated function and the operands. The calculated function is also represented in the metadata system and can be used as an operand in other transformations. So, in the space of the function definitions, transformation relationships between data form a network that traces the system's calculus processes.

Since a transformation involves both the intensional and the extensional form of the functions, the algorithm has to describe both aspects. An example is the ability to calculate the result in connection with the knowledge of operands. In other words, the transformation must incorporate rules to determine the knowledge domain of the result from the knowledge domain of the operands.

Transformation algorithms can be defined using very different kinds of operators (algebraic, logical, statistical, qualitative data, etc.), which are previously inserted in the system grammar and that refer, at a less conceptual level, to software routines able to perform them.

[8] So that the statistical function is the transformation unit.

In addition, transformations are typically defined by the administrators.

Some examples of basic transformation type can be given.

Aggregation consists in calculating the function value of the generic group “g” of statistical units (or time periods) starting from knowledge of the function values of other disjointed groups “g_i”, of which “g” is a set-type union.[9]

Domain transformations operate on the statistical function domain and include, for example, sub-setting, partitioning, joining as in the case of extraction of time series or cross sections from the most general form of a statistical function, and vice versa.

Function composition, whose definition also comes from mathematics, makes it possible, for example, to define algebraic operators on functions, which can be further combined in expressions to generate more complex algorithms.

STATE OF THE ART OF THE SYSTEM AND TRENDS

Work has been under way on the statistical information system and related models for a long time, and it is far from finished. The early project of the actually running software packages started in the mid-eighties, on the basis of earlier experiences of a similar kind. The last packages were built in the second half of the nineties.

A first step has been substantially achieved. This is to refer to the same conceptual model to define concepts and data structures of different packages. Packages are thus able to interact easily to perform the overall processing.

Further improvements have to be made, mostly as regards transformation model compliance, not supported in the same way from different packages. Only the most recent implementations, in fact, were able to take advantage of improvements coming from new practical and theoretical experiences.

As regards building a unique metadata system at a conceptual level able to drive different packages (the so called “statistical dictionary”), at the moment there is a first running release, which supports part of the common metadata model (concepts and data) and drives the packages involved in the surveys of institutions subject to supervision (they process most of the data).

A second phase unification is now starting, to support the whole model and all the packages and to achieve further software unification. In this phase, attention will also be paid, as far as possible, to compliance with international standards issued in the meanwhile.

On the technical side, statistical software packages are mainly made in-house. In many cases there was no choice. Few software packages on the market could be easily integrated into such an architecture. Essentially, only software packages such as DBMS, some user-oriented packages to access and process data (such as SAS), and some specialized packages (e.g. for time series processing and storing, such as Speakeasy and Fame) were bought.

[9] Such a relationship type must be defined within the statistical concepts model.

Even now that data warehouse technology has spread in the market, there are difficulties in buying, because software packages often do not deal well with some important requirements of the system (such as the historic qualifications of concepts, data and transformation rules) or have other serious limitations (e.g. it is difficult to drive them with external metadata).

There is therefore great interest in the international standards development process and in the possible future availability on the market of software products able to support such standards.