WP. 6
ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE**

Joint UNECE/Eurostat/OECD work session on statistical metadata (METIS)
(Geneva, 9-11 February 2004)

Topic (ii): Metadata interchange

MetaAPI – improving the usability of existing metadata systems by application programming interface

Invited Paper

Submitted by Statistics Finland[1]

I. INTRODUCTION

1. This paper describes the application programming interface built on top of the existing metadata systems at Statistics Finland. The interface is here referred to as MetaAPI. Metadata are stored in RDB-based systems (Classification and Concepts DBs, Unified File System, etc.), which were developed in-house during the 1980s and 1990s. The main concern is in the implementation of the classification database services. The first experiences of using MetaAPI in different stages of the statistics production process are also described.

2. The reasons that lead us to the development of this interface are described as well as its structure, technological choices and basic services. One goal was to reach platform and software independence, which was achieved by using XML Web Services. The need to have a theoretically solid optimal metadata model can be reduced by creating interfaces, because this makes it possible for the applications to use metadata in a flexible way regardless of the underlying data models and systems. The life cycle of old systems can be extended in this way.

3. In the MetaAPI development some attention is given to the need to utilise common standard solutions and interfaces but the main focus is on the improvement of the existing in-house systems. This will not, however, rule out adoption of standard solutions at a later stage if desired.

II. BACKGROUND

A. Statistics Finland's metadata systems

4. The **Classification Database** was set up in the mainframe environment in 1985 and it was transferred to the client/server environment in 1993. The statistical classifications used at Statistics Finland are stored in the Classification Database. It contains the codes, labels, explanatory notes and keywords of the items and the correspondence tables, tabulation rules and metadata related to classifications (validity period, responsibility

[1] Prepared by Mikko Mäkinen (mikko.makinen@stat.fi), Toni Räikkönen (toni.raikkonen@stat.fi), Risto Syvänperä (risto.syvanpera@stat.fi), Saija Ylönen (saija.ylonen@stat.fi)

person, etc.). At the moment, the database contains around 1,020 classification versions and 1,700 text versions. The database can have several time, statistical and text versions of the same classification. The time versions are different versions of the same classification from different years (e.g. NACE rev. 1.0 and NACE rev. 1.1.). Statistics Finland's statistical units can also produce their own versions of the classification standards on the basis of their summing rules, for example. There can be several text versions, such as one made for a publication where the labels are shortened, or different language versions. Finnish, Swedish and English-language versions of classifications can be stored in the Classification Database.

5. The **Concepts Database** was built in 1999 as a storage location for statistical concepts and their definitions and other metadata related to concepts. The purpose of the Concepts Database is to collect all concepts used at Statistics Finland into one location so that they are readily available to all and statistical units need not maintain any concept lists of their own. At the moment, the database contains around 1,000 concepts. There may be several versions of the same concept. In addition to time versions, statistical units may define the same concept operationally in different ways. It is possible to define each version of a concept at two different levels of precision. The **general** definition is fairly brief and mainly intended for incorporating into publications, databases and other products and services. The definition for **experts**, as the name implies, is precise enough to serve the purposes of experts such as producers of statistics. The database also comprises Swedish and English versions of concepts.

6. The **Unified File System** is used to describe and process statistical data and to describe Statistics Finland's electronic data to be archived. The application connected to the system can be used to view, search, maintain and make new unified file descriptions and to process unified file data. The system also performs conversions to SQL, Gesmes, PC-Axis, SAS, PC and Statfin[2] file formats, and it can be used to transfer unified file descriptions and data between the mainframe and the application. The idea for a unified file system first emerged in the early 1990s from the needs to transfer data for national accounts. The system was created to the mainframe environment and it was intended for general application. From 1996 the Unified File System has also operated in the Sybase database in the client/server environment.

7. Statistics Finland's other metadata systems include the Archive System AMS and the System Register SYSREK and certain Statistics Finland's internal administrative systems.

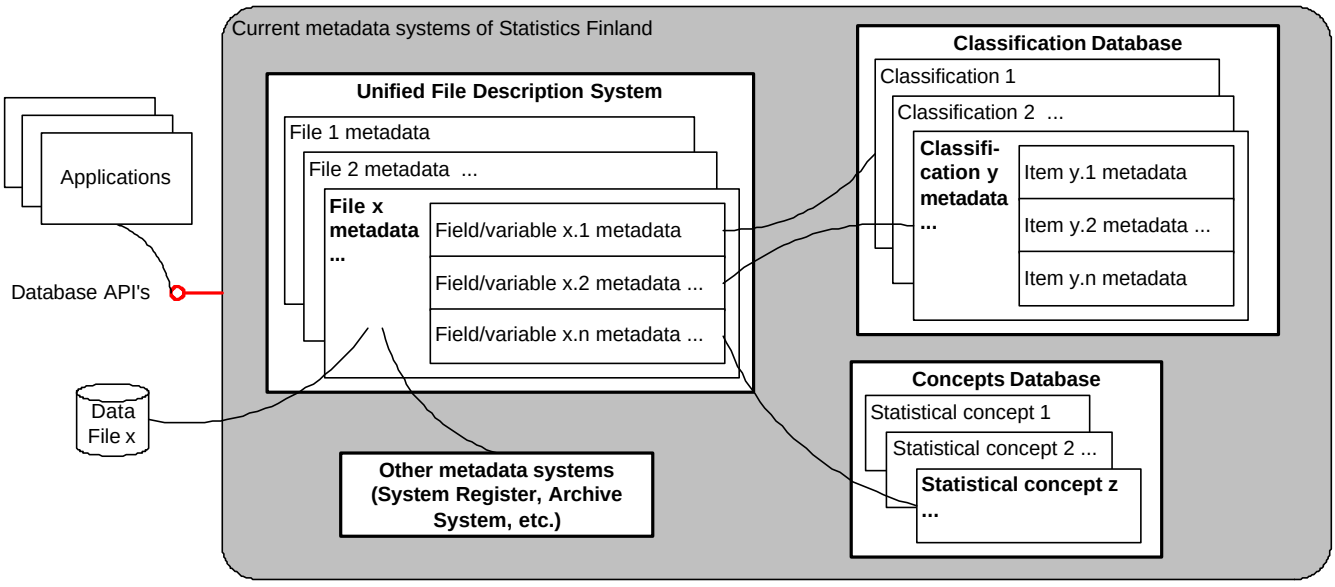

Figure 1. General image of Statistics Finland's metadata systems and connections between them.

B. Metadata system interfaces and use cases before MetaAPI

[2] StatFin is a service database on Statistics Finland's Internet pages containing statistical data on Finland.

8. The Unified File System is used to link other metadata systems with actual data materials. In that way the system builders have access to all the metadata needed through one centralised system. Linking to the Data File, the Archive System and the System Register takes place on the data material level. On the variable level linking is made to the Classification and Concepts Databases. On the variable level linking is not yet committed to any text version, whereby language selections for publications cannot be made until in the publication stage, as required, without needing any separate metadata definitions for different uses. The Classification and Concepts Databases and the Unified File System can be utilised through open database interfaces (e.g. ODBC).

9. Examples of metadatabase interfaces and use cases:
- The Classification Database is used to produce classification handbooks and classifications and related metadata on the webpages of Statistics Finland's Classification Services unit (http://tilastokeskus.fi/tk/tt/luokitukset/index.html).
- The Classification Database can be used to fetch label and summing rule formats to SAS environment and code and grouping lists to the online StatFin service.
- The Concepts Database metadata and SQL inquiries can be used to search concepts and their definitions for statistical applications.
- The Unified File System contains several maintenance programs for tabulation, conversions and data transmission, such as for text, SAS, PC-Axis, SuperCross and Gesmes formats. Unified files can be converted into other formats or files made in other systems can be converted into unified files.

C. Why MetaAPI?

Problems with the present metadata systems

10. The following factors have hindered any wider implementation of metadata systems at Statistics Finland:
- User interfaces are somewhat outdated, rigid and unsuitable.
- Text descriptions are not sufficiently structured and the description power of plain text is not adequate for mathematical formulas, figures, highlighting, etc.
- Technical and statistical metadata are intermingled.
- There is no direct data connection to several file formats.
- Training in application development does not yet have any practical connection to metadata systems.
- Metadatabases are often conceived as mere archiving tools. However, the same metadata should benefit all the stages of production.

11. In order to extend the use of databases, easy-to-use general modules and services are needed.

Production model project

12. In 2002 Statistics Finland initiated a production model project that aims to harmonise and improve statistics production processes. The basic principles of the project have been its data warehouse foundation, modularity, and the key role of metadata, interfaces and standard solutions in the production process architecture. The development of the metadata interface has been part of this work.

XML Web Services

13. XML Web Services are the extreme end of distributed application development, which enable implementation of B2B type solutions between totally unknown parties. Although the highest benefit of Web Services must come from such implementations, they can also be utilised to develop substantially the organisation's internal application architecture.

14. By means of XML Web Services, it is possible to move from component-based architecture models closer to service-oriented architecture. Service-oriented architecture also allows production of software and platform-independent solutions in the organisation's internal application production.

III. MetaAPI (META APPLICATION PROGRAMMING INTERFACE)

15. Over the years, it has been discovered at Statistics Finland that metadata could actually be used as a key tool for the development of production processes. At the same time, it has appeared that the existing metadata systems do not always serve production as efficiently as they could. The major problems were found to be a certain degree of inflexibility of the systems and the need of users to be familiar with the internal metadata models of several separate systems in cases where the functions of ready-made general service applications are not sufficient. Correcting these defects has been one of the most challenging tasks of the production model project.

16. The primary objective was to introduce some ready-made – preferably standard – solution. However, it has not, at least yet, been possible to find any satisfactory ready-made solution that would maintain the service standard of the present systems. Studies for finding or developing a suitable, comprehensive metadata solution still continue, but a faster route was also needed for realising the requirements the production model sets for metadata.

17. The work was based on the existing metadata systems as such and on the new possibilities offered by technological development (particularly XML and Web Services), and stress was also laid on the importance of interfaces. We believe that in this way most of the benefits sought could be attained and thus prepare the ground for an advanced, general metadata solution possibly introduced in the future.

18. MetaAPI must be able to serve optimally many different solutions produced with different tools and functioning on different platforms. It would also be advantageous if metadata could be provided by using the same interface not only for internal use but also restrictedly for external use through the Internet. Therefore XML-based Web Services were chosen as the primary implementation technique using SOAP [3] as the communication protocol. The primary advantage of the XML-based interface is above all openness, because then we are not committed to any specific binary format.

19. When defining the interface services the aim is to make them simple and easy to adopt and use. For this reason the number of services should be as low as possible but nevertheless, comprehensive. The location of MetaAPI between the metadata systems and applications is shown in Figure 2.

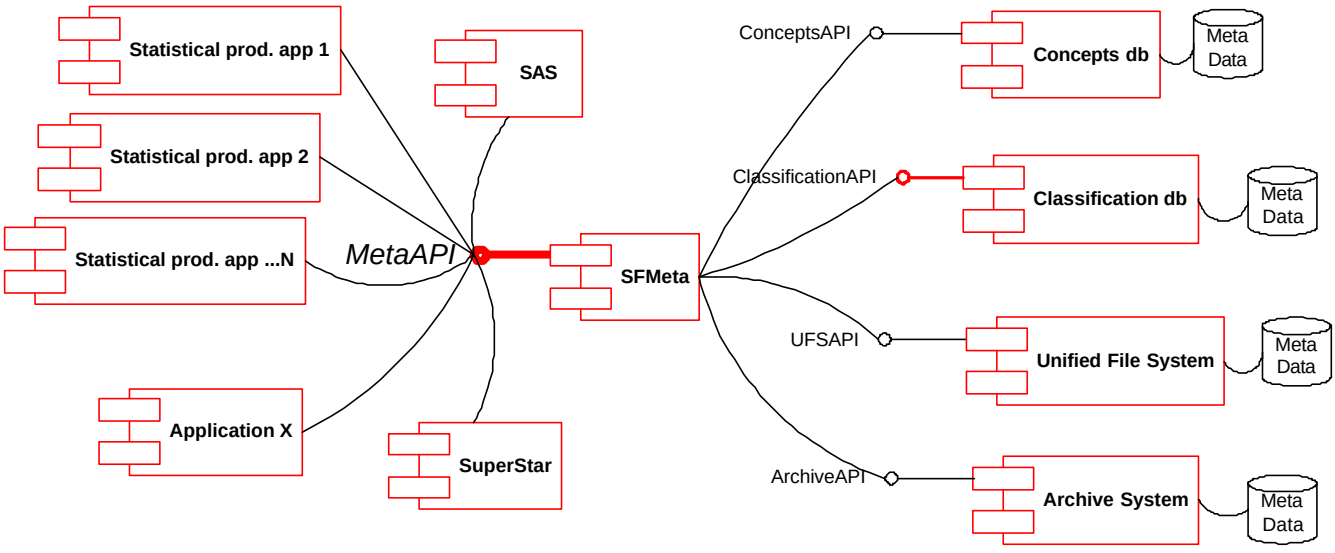

Figure 2. MetaAPI metadata interface, target situation

[3] Simple Object Access Protocol (SOAP)

D Implementation guidelines

20. The first stage is restricted to the existing systems and no changes are actually planned into their structure or no attention is paid to any other possible replacing systems possibly introduced. The existence of general models and interfaces, such as CWM/MOF/XMI and ComeIN, is noted but they are not meant to be implemented primarily at this stage. The main focus is on improving the usability of the present systems and on separating the applications from metadatabases. The aim is to produce an interface that defines Statistics Finland's principal metadata services. In case other metadata interfaces and systems are introduced, it should be possible to implement MetaAPI by means of them as well.

21. The idea was to create service interfaces separately for each metadata system as well as a general MetaAPI connecting all metadata systems. Services in system-specific interfaces could be familiar with the structure of the background system underlying them and be as general as possible, accurately parameterised, and not necessarily optimally user-friendly. Technical services intended for maintainers would also be placed into these interfaces. The general service interface, MetaAPI, would contain clear and easy-to-use services intended for end users and applications. Meta systems would be used in MetaAPI through each system's own service interface. MetaAPI could also comprise services that search for and combine metadata from several metadata systems. One important goal is to reduce the users' need to know the technical structure of separate systems.

22. The work was started from defining and piloting the basic services of the Classification Database. Other metadata systems will be connected to the service later on.

E. About change management

23. The presented interface division serves the interchangeability of systems. If the entire metadata solution or only some separate system is changed into another, it is necessary to correct only the interface of the component to be changed. The actual MetaAPI can stay unchanged regardless of the background systems. Then no changes need to be made to the application programs that use MetaAPI when the metadata systems are being modified. MetaAPI would also dictate the needs that new metadata systems should fulfil. At the very least, the existing service standard must be retained when introducing new systems.

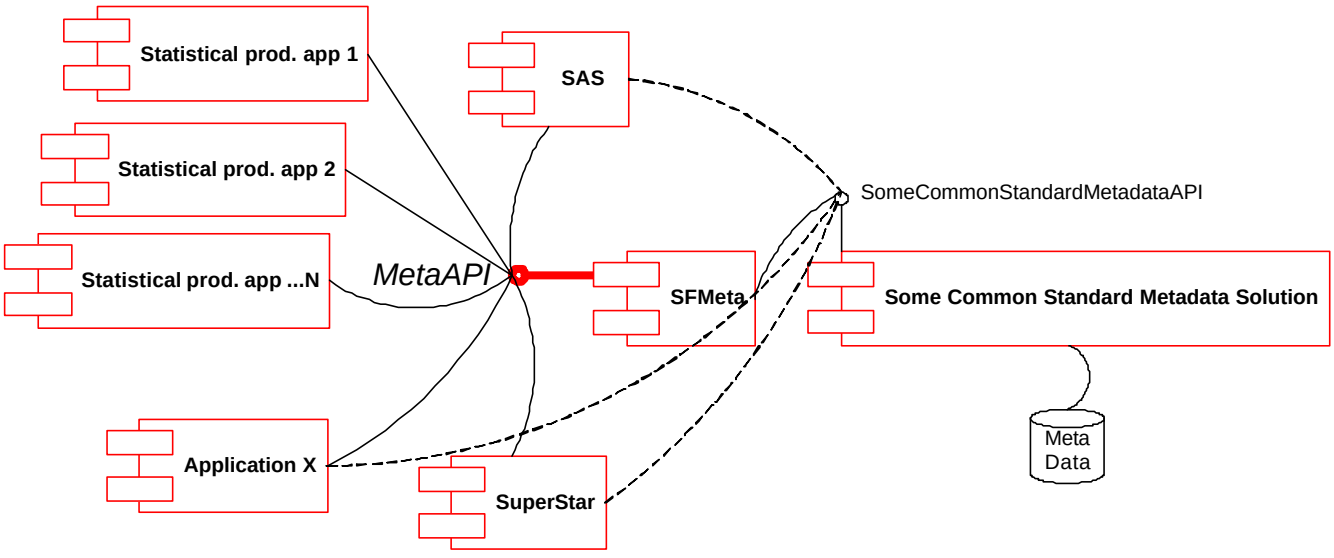

Figure 3. Change management by means of MetaAPI: changes are not needed in the application regardless of the changes in the background systems. The dashed lines describe the possible direct connections in ready-made programs to the actual metadata interface past MetaAPI.

24. A prerequisite for such flexible change management is that MetaAPI may not be connected in any way to the implementation of the actual metadata systems. MetaAPI must be completely unaware of the names or structure of the background system tables, fields and other such objects. In a change situation the necessary

changes are made to the SFMeta[4] without touching MetaAPI. In order to minimise the transition time caused by the simultaneous use of several metadata system generations, all metadata services are to be made through MetaAPI and direct connections are denied to separate metadata systems or even to their interfaces. This restriction can naturally be put to use only when the service capacity of MetaAPI is sufficiently flexible and comprehensive.

25. Integrated solutions accompanying ready-made programs constitute a probable threat to that all metadata processing would be made in a controlled manner through the specific metadata interface. It may be difficult to explain why metadata processing should be made in a different way than planned by the software provider.

F. Services

26. Services are here discussed on a very general and principled level. In the first stage, metadata read services, or the means for utilising the existing metadata, are described and implemented. Metadata write and maintenance services are made later on.

27. MetaAPI comprises general and user-friendly services for applications. It can also contain more tailored, often needed special services, such as those related to regional classifications. These services are connected to some frequently needed use case, by which the benefit gained from user-friendliness is greater than that produced by a generalisable solution. It is also necessary to define in MetaAPI basic services that easily provide a search facility for large metadata entities. Then each bit of metadata need not be searched separately by calling services. The application program then attends to the management and use of the metadata entity received. The comprehensiveness of the interface is also ensured by means of basic services. If some service is not (yet) available, it can be replaced by using a basic service and by working on the management of the result set at the client side.

Basic services

28. As basic services MetaAPI offers the facility to search in a parameterised way the metadata of an individual metadata entity. The result is returned as an XML document containing all the essential information related to the required metadata entity. This requires making of XML schemas for each entity. The purpose of using basic services is that the application can have access to a large metadata entity by one service request. Further use of the received XML document is made application-specifically utilising general XML programming techniques.

Individualised services

29. Services returning a smaller metadata entity are often provided in addition to basic services. Where possible, these services are offered so that the returned document complies with the structure of the return document of the corresponding basic service. The structure of the return set of the individualised service is then a sub-set of the schema describing the structure of the result set returned by the basic service, i.e. a ”sub-schema”.

G. XML schemas

30. XML schemas are essential for the functioning of MetaAPI. Without them it is difficult or nearly impossible to use XML documents returned by Web Services. The importance of XML schemas in MetaAPI is similar to that of the interface in component-based software production. It is an agreement on that the data according to the given model will be obtained as a result of services. By means of the XML schema it can be easily verified whether the received XML document follows the schema, i.e. whether it is valid.

31. Some of the schemas needed by MetaAPI are:

---

[4] **S**tatistics **F**inland Meta: A binary interface that collects the services of individual metadata interfaces into one component.

- Classification schema: codes, explanatory notes, text and language versions, correspondence data, etc. related to an individual classification.
- Concept schema: definitions, text and language versions, metadata, etc. related to an individual concept.
- Unified file schema: technical data, definitions, location of data, used classifications, concepts, keywords, etc. related to an individual file description.
- AMS schema: data related to an individual archive formation plan.

32. Figure 4 illustrates the rough structure of a classification schema. As metadata it is possible to transmit general information about the classification version (e.g. validity period, number of levels, length of code) and text description about the structure of the classification and classification principles. Text versions are label versions of different languages and lengths. In addition to codes and labels, explanatory notes and keywords can be given about items.

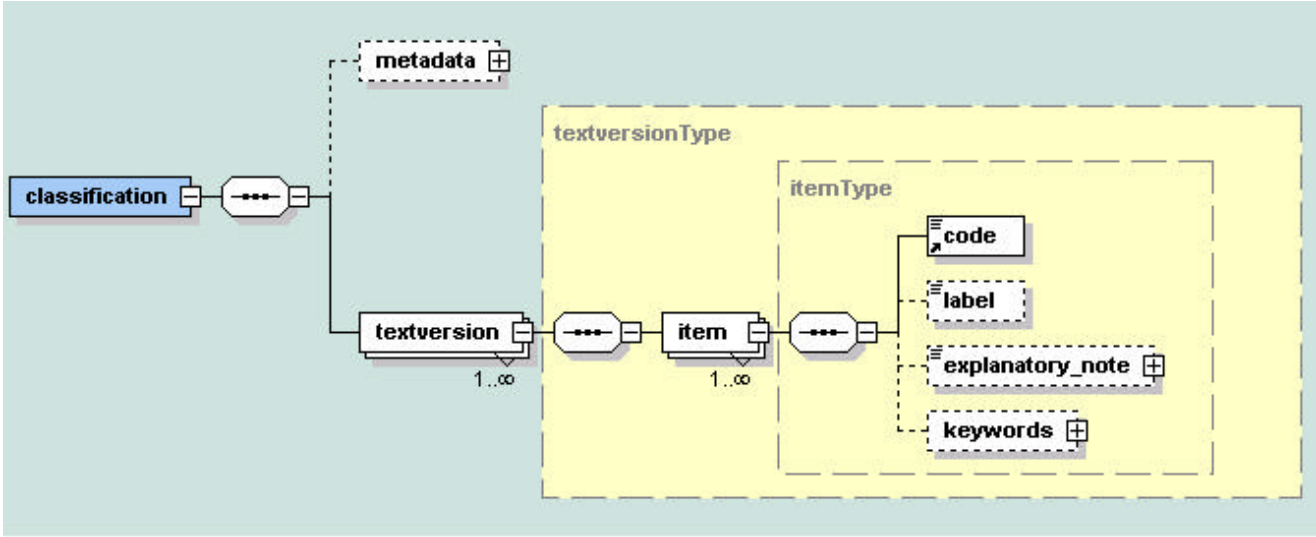

Figure 4. Classification schema

H. Technical solutions

33. MetaAPI is implemented in accordance with the layer architecture so that it could be changed as easily as possible, when needed, and the changes would concern only a limited part of the system.

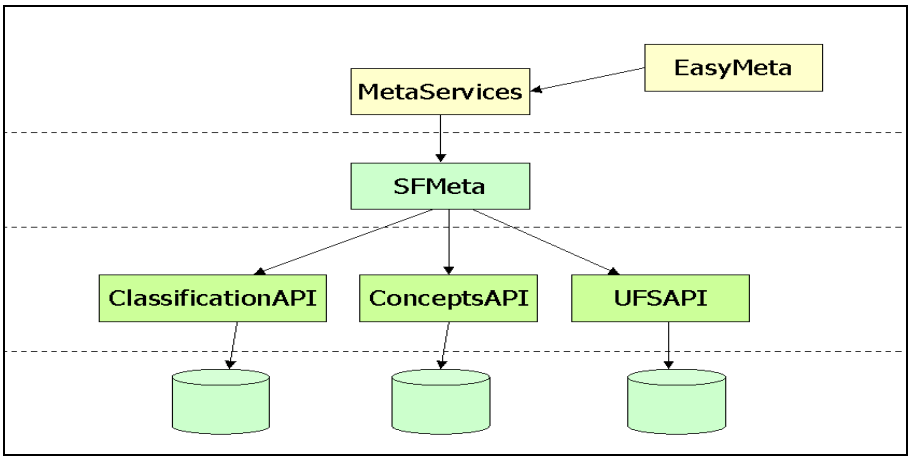

Figure 5. MetaAPI layer structure

34. The highest layer was implemented by means of the XML Web Services technique. It contains two Web Services interfaces: MetaServices and EasyMeta. The MetaServices interface includes the basic services with which to obtain the metadata (e.g. municipality classification) related to an individual entity. The services in the MetaServices interface typically comprise a large set of parameters with which to select the desired metadata entity and, where necessary, limit the size of the result set to be returned.

35. The EasyMeta interface contains simplified services requiring either only very few or no parameters. A typical service implemented in the EasyMeta interface could be searching for a valid classification.

36. The services of the EasyMeta interface exploit the services implemented in the MetaServices interface by transmitting to them suitable built-in parameters in connection with a call.

37. Underneath the topmost Web Services interface there is an SFMeta binary interface that primarily functions as a facade that collects the services of the lower level individual interfaces into one component. The SFMeta component is in charge of directing a received service call to the correct lower level interface.

38. A layer containing the components of individual metadata systems is below the SFMeta level. These components also implement the binary interface. On this layer a separate programming interface has to be implemented for each individual metadata system. Only this interface is familiar with the database structure of the underlying metadata system and the schema of services. The services implemented in this interface are able to produce an SQL inquiry to the underlying relational database with the help of the parameters derived from the MetaServices interface.

39. The lowest MetaAPI layer contains the relational databases of individual metadata systems.

I. Use cases

Browser-based testing site

40. An Intranet site was developed for easy testing of the XML Web Service interface. The site provides MetaAPI services in a simple form format. The user fills in the fields of the desired service in the form and initiates a call to the MetaAPI Web Service interface.

Statistics production applications

41. Statistics Finland's statistics production applications are mainly produced with Microsoft Visual Basic.NET and Sybase PowerBuilder. Both of these tools provide a user-friendly facility for using Web Services. In practice, it is almost invisible to the programming language whether the metadata are fetched direct from the relational database or whether they are derived through Web Services.

Commercial Software

42. All commercial metadata utilising software used by Statistics Finland cannot yet exploit Web Services directly. This produces extra work if the same metadata services are to be used in all the production stages. It is highly probable that some kind of Web Services support will appear to most of such software in the near future.

43. A few examples of the problems with ready-made software:
- SAS: Use of Web Services from SAS is not possible if the SOAP protocol is used to package calls. Calls can be made using the HTTP-GET protocol but this solution is not in compliance with the Web Services ideology.
- SuperSTAR II: Unable to utilise XML Web Services at least for now.

J. Experiences / problems, etc.

44. The MetaAPI interface has been tested on Classification Database services in a revision project of a statistical system. In the project PowerBuilder is used to implement an application that utilises classifications with the help of MetaAPI. The experiences have been quite positive. The initial problems were caused by that applications analysts were accustomed to using classifications directly from the relational database.

45. The use of Web Services requires XML skills of applications developers particularly in situations where the services return large result sets as XML documents. This is made easier by the existing and evolving XML features in application development tools.

46. It is difficult to parameterise the services of the Web Services type in certain situations (e.g. optional parameters, variable-size table parameters, exceptional data types). The parameter list easily becomes long and confused. Our solution to this problem is the EasyMeta interface. Another solution would be transmitting the parameters as a structured XML character string, but then it would be difficult to create the parameter XML at the client side.

47. From the viewpoint of efficiency, the selected technique seems promising. Formation and moving of large result sets takes time, however, for which reason it is advisable to optimise the uses of MetaAPI services at the client side.