The STREAM's Testdefinition facilitates type of test independent database storage.

Paul E.L. SCHAMINÉE[a,1] and Ardt A. KLAPWIJK[b]
[a] *Deltares, Delft, The Netherlands*
[b] *NCIM B.V., Leidschendam, The Netherlands*

Abstract. Results of many different types of physical experiments have to be stored. To facilitate efficient archiving and exchange of test results a new method, STREAM, has been developed. This paper describes a database that utilizes the characteristics of STREAM, in order to create a flexible, robust and incorruptible tool to store data from a broad range of geotechnical experiments. The result is a type of test independent database. All information stored in the database is checked to be correct and complete. The principle structure of the database is a searchable set of both general metadata as type of test specific metadata to identify, to query and to select the data file. Using STREAM's standardized data description, this type of database is 'prepared' to accept any new type of test with corresponding data files, without specific programming. The described storage process is unique in its kind and currently operational at Deltares.

Keywords. Standardize the description of a test, store results of experimental testing in a database, Testdefinition, STREAM.

Introduction

In any field of engineering and research, data are gathered, analysed and reported. These data are transferred from one user to another, several times. In some cases the data remain on one location for a longer period: it is stored. In this paper the focus is on data obtained by experimental testing. At the start of an experiment only data is gathered about the setup of the experiment. When the actual experiment is performed the actual measurement results are obtained. After performing the actual experiment the obtained results are analysed and reported. Finally the results are presented to a client and stored for future reference.

Concerning the information available two observations were made: it differs (i) at different stages and (ii) for the different types of tests. Deltares was confronted with the problem to store and exchange data from numerous types of tests, varying from simple moisture content tests to complex geotechnical centrifuge tests. No appropriate method was found that covered this broad range of tests.

Therefore in the last fifteen years Deltares has autonomously solved this problem by developing and implementing a methodology, called STREAM. In this method each type of test is treated independently and for each single type of test five chronological - so-called SMARF - phases involving any form of experimental activity are

[1] Corresponding Author: Deltares, P.O.Box 177, 2600 MH, Delft, The Netherlands; Email: paul.schaminee@deltares.nl.

distinguished. In this method all information available after a phase of an experiment is stored in a data file, so five data files for one single test are created. This set of five data files contains all meta data and test results of a single experiment. A database was designed, which was able to store and retrieve all data files created during the STREAM process.

The developed database can be easily extended by new types of test without any programming and still tests can be found by using type of test specific metadata. Now users, i.e. operators and researchers, can define their own set of relevant type specific metadata. This feature appears to be very powerful in research environments where many different (or modified) types of tests are performed and need to be stored securely. Of course general metadata such as filename, date of creation, project identification are also available for queries.

This database, designed to facilitate STREAM for a broad range of geotechnical tests, is described in the next sections. In Section 1 the underlying concepts are described and in Section 2 the functionality and design of the database are described.

1. The Underlying Concepts

The database described above requires chronological structuring of the measurement process and standardization of the descriptive information of corresponding data files. The recently developed method fulfils these requirements [1]. This method is centered around two elements (i) a chronology that captures the activity involved in any form of experimental activity: SMARF and (ii) a method that structures information generated throughout the SMARF process: STREAM.

1.1. The Concept of SMARF

The SMARF concept identifies five chronologically-ordered phases from the design of a test to its factual reporting. These phases lead to the SMARF acronym – set-up, measurement, analysis, reporting and filing.

- *Set-up phase* in which the equipment and samples are prepared;
- *Measurement phase* in which the actual experiment takes place, i.e. the gathering of sensor readings;
- *Analysis phase* in which the recordings are analyzed and new derived quantities are calculated or key values at particular times are extracted;
- *Reporting phase* in which both the measured and calculated results are presented attractively by means of figures, tables, etc;
- *Filing phase* in which the results are prepared for long term storage and future accessing.

The SMARF concept provides the moments to transfer a well defined phase data file by storing it in the database, to make it available for the users in the next phase.

1.2. The Concept of STREAM

The method called STREAM is developed to facilitate efficient exchange and archiving of test results. STREAM is an acronym for Standardized Test Results Exchange and Archiving Method. Documentation of exchanged or archived results is an integral part of the process. Tests of the same type share the same procedures and therefore share the same documentation with respect to the description of quantities and equipment. Each quantity or item is referred to as 'element'. The precise descriptions of all these elements (i.e. an explanation in a standard way, comprehensible by any user) for a type of test are stored in a so-called 'Testdefinition', which is a document that is formatted in a standard way. Only one Testdefinition is created for all tests of the same type, illustrated in figure 1. All values, parameters and readings from each single test, are stored in a separate data file.

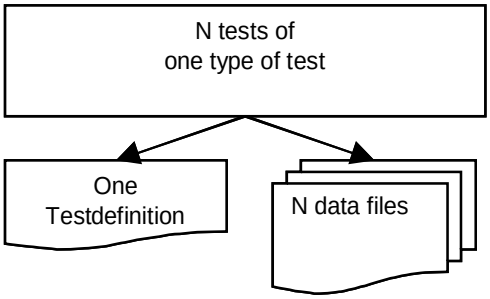

Figure 1. A Testdefinition and the corresponding data files

In STREAM the five SMARF-phases are defined separately. In the Testdefinition it is clear in which of the five SMARF phases an element is created. Using the Testdefinition each data file can be checked for completeness and correctness before entering the next SMARF phase.

1.3. Data File

A data file consists of a header and a data section. The data section contains quantities with more than one observation: these elements are referred to as columns. Typically columns contain the series of registered and analyzed values. The header contains the context information, referred to as metadata:

- *General metadata.* This general information is common to most types of tests and is therefore predefined. These general information elements are well defined. They can be grouped in two categories: (i) file tracing, i.e. information to identify for example the organization where the test was performed, the person responsible for the test (ii) test description, i.e. the reference to the corresponding Testdefinition.
- *Type of test specific metadata, or typespecific metadata.* These elements are used to store information that is only relevant to describe the context of one specific type of test, for all other types of test this information is meaningless.

All elements in the data file are described in the Testdefinition. For STREAM it is essential that the relation between the value in the data file and the Testdefinition is maintained continuously. The interaction between the Testdefinition, the test data and the software is based on an unique key for each element, referred to as 'shortname'. Although a shortname is defined for one type of test, its value assigned will be different for each single test performed in most cases.

Currently STREAM is implemented with a GEF formatted data file [2] and an MS-Excel data file, but could be easily extended for example an XML data file. Conversion between different types can be performed easily.

1.4. Testdefinition

The Testdefinition turns out to be the central part in the STREAM process. Only with a completely defined Testdefinition, test results can be acquired, validated and stored.

A Testdefinition can be created using the Testdefiner®. For each SMARF phase the elements to be acquired can be defined with a distinguishable shortname and adequate properties. This results in the Testdefinition, which is an XML-file. Besides a detailed description of each element, a Testdefinition also offers the opportunity to enter reference data. This additional information is crucial for comprehensive description of a test and its procedures. Examples are:

- *Coordinate Systems*. Coordinate systems are used to be able to work with sensors in mounted in modular test set-ups. A sensor location is referenced by a position relative to one defined coordinate system. In one Testdefinition all coordinate systems are related, and software has been developed to change the coordinate systems in which locations are expressed originally.
- *Input Editors*. Each element can be assigned to a person or role. This user is responsible to enter these elements. A generic application has been developed to allow a specific editor to enter information using specific dialogs based on the Testdefinition. This appeared to be very powerful and by combining the advantages of the STREAM concept and the type of test independent database storages.
- *Applied Procedures*. Each element can refer to an applied procedure, which itself is a reference to a procedure that is applied, stated in another document. This connects the Testdefinition to existing standard procedures.
- *Enumerations.* General en typespecific metadata elements can be assigned a list of enumerated values or strings. These enumerations can be used to create an input program and forces the user to choose an item from a limited list.

2. The Database

A database is a common way to store data permanently. This is done to keep the data available for a longer period of time on a centralized location, but also to be able to query the data in the defined entities (tables) in order to gain statistical information. In most situations when large amounts of information need to be stored a relational database system is used, with a limited number of specified entities. However, storing test results from a data file in a database is not so straightforward. The general metadata

can be stored in prescribed entities, but the typespecific metadata can vary from type to type. It varies in the number of elements and the type of the elements. On the other hand, only storing the data files doesn't offer the opportunity to query the results very easily and negates the possibility to check the validity of the data in the files.

Deltares has developed a mixed solution in which a set of the general and type of test specific metadata is stored in a relational database and the data files itself are stored on a central location. The relation between the relational data and the data files is taken care of by central services. Additionally, important general and type specific metadata is extracted from the data files and stored in an entity.

To apply the STREAM methodology, a test should be related to a Testdefinition and all metadata should be referenced by a shortname. This is done by putting the Testdefinitions in the database too, both as a file and extracted in relational entities. In this way, type specific metadata can be stored.

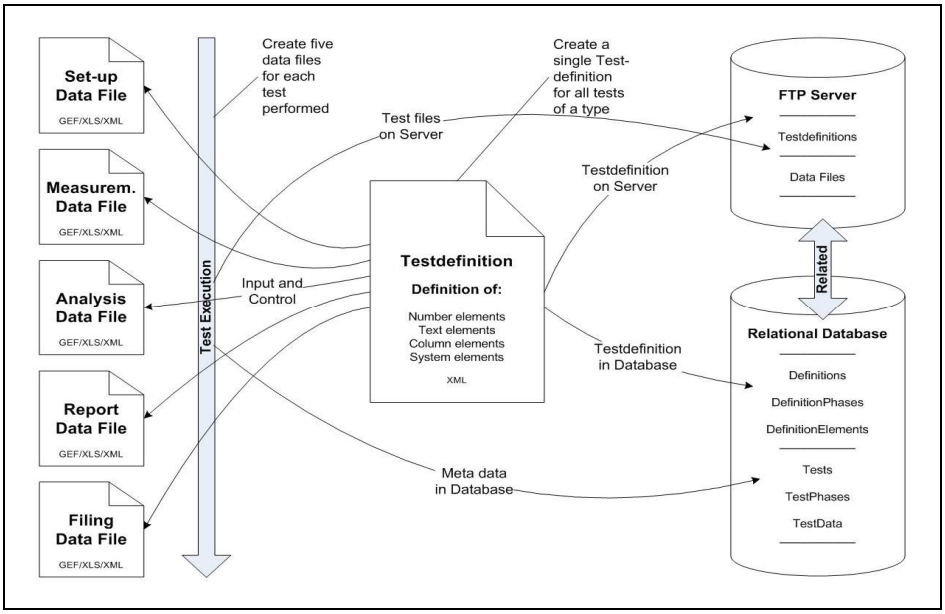

Figure 2. Relation between Testdefinition, SMARF phases, database and FTP server

2.1. Data Types and their Properties

The Testdefinition describes all elements that might occur in the data files. These elements are classified as :

- numerical elements (Number),
- textual elements (Text)
- system elements (System); these are predefined elements included for meta information like project identification, file owner and file date, and test information like start date and location. These elements can have a composite structure.
- columns of numerical values (Column).

In the Testdefinition the elements of different data types are defined by several properties, depending on the data type of the element. In table 1 the most important properties are listed, related to the data type.

Table 1. The properties for data type

	Number	Text	Column	System
Shortname	**x**	**x**	**x**	**x**
Required	**x**	**x**	**x**	**x**
Documentation	**x**	**x**	**x**	**x**
Searchable	**x**	**x**		**x**
Quantity	**x**		**x**	
Unit	**x**		**x**	
Minimum/maximum value	**x**		**x**	

It should be noted that a Column does not have the property “Searchable”, this is because the values of a Column can not be used as a parameter in a query. Furthermore, a Column element can be enhanced as a sensor, which provides additional options to enter information regarding the sensor calibration and location.

2.2. Identification and Versioning

A Testdefinition is identified by a definition code. This is combination of a label, including the owning company and the type of test, and a release, version and update number (version for short). For example: Deltares.nl:ConePenetrationTest, 1, 0, 1. The first part of the identification of a Testdefinition is the internet address (without the preceding ‘www’) of the company responsible for all security aspects of the Testdefinition, i.e. uniqueness, version management.

Also each of the five SMARF-phases is identified by a phase code. These codes are also structured as a label and a version number. This gives the opportunity to copy definitions of phases between various versions of a type of test or even between different types of test. Whenever something changes in a certain phase, the phase version of that phase needs to be increased. Also the definition code needs to be increased, because a Testdefinition is uniquely identified by its definition code. Table 2 presents an example when changing the definition of the analysis phase. It is common practice that also the report and filing code are increased although nothing has been changed to those phases. This guarantees that the software is backwards compatible.

Table 2. Version before and after applying a small change in the analysis phase

	Definition	Setup	Measurem.	Analysis	Report	Filing
Before	1, 0, 0	1, 0, 0	1, 0, 0	1, 0, 0	1, 0, 0	1, 0, 0
After	**1, 0, 1**	1, 0, 0	1, 0, 0	**1, 0, 1**	**1, 0, 1**	**1, 0, 1**

2.3. Checks at SMARF Phases Transition

After finalizing a SMARF-phase the file is put in the database after validation. In the same process the values of the metadata elements indicated by the Testdefinition as ‘searchable’ are copied into the relational database. To keep the database up to date and

incorruptible, some portal software has been developed. Its main function is to check all files that are submitted to the database and store them in the right location. The extracted metadata are put in the relational tables. A file that doesn't pass the validity checks based on the Testdefinition or checks against reference data (e.g. does the project exist?) is rejected and cannot enter the database. The user should correct it and retry.

2.4. Design of the Database

Figure 3 shows the structure of the database, which is explained in the next paragraphs.

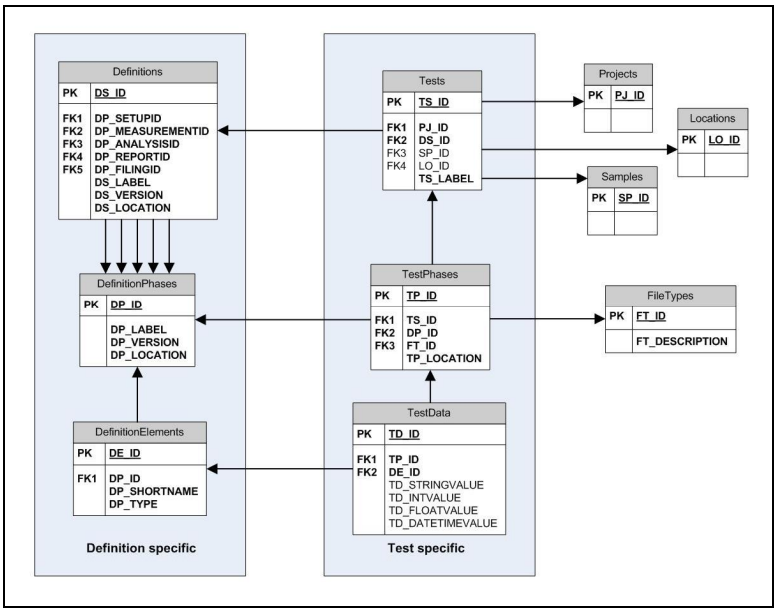

Figure 3. Key entities and their relations in the database

The left block - the Testdefinition related part - applies to a specific type of test. It contains all relevant information from each single Testdefinition. Entity Definitions contains the definition code (DS_LABEL, DS_VERSION) and the link to location of the actual Testdefinition file (DS_LOCATION). Entity DefinitionPhases contains information the separate parts of the Testdefinition for the five SMARF-phases. This is the phase code (DP_LABEL, DP_VERSION). It also contains a link to location of an extract of the Testdefinition file that contains the checks for the specific SMARF-phase (DP_LOCATION). These comprise the checks whether an element is present, if it is of the right type and if its value lies between the specified minimum and maximum value. Entity DefinitionElements contains the general and typespecific metadata elements from the Testdefinition which are marked as 'Searchable' (not for Column).

Note the special relationship between Definitions and DefinitionPhases. As DefinitionPhases can be shared among more than one Definition, entity DefinitionPhases acts as a parent, where logically Definitions is the parent.

The right block - the test related part - consists of the entities that contain data of a each single test. Entity Tests contains the specific tests that are carried out. Each test has an identifying label and can be related to a Definition. It can also be related to certain reference data in the database, like Projects, Locations and Samples. Entity TestPhases contains the separate data files for each of the five SMARF-phases of each test. Finally entity TestData contains the extracted values for the general and typespecific metadata elements in DefinitionElements which were present in the data file. Depending on the type one on of the value fields is filled.

3. Conclusions

The STREAM concept offers the possibility to store and exchange test results of all types of test in a way that is both effective and flexible. The Testdefinition plays a central role in this concept. STREAM can only function properly if the Testdefinition is complete, which requires quite some effort. Therefore it is less suitable for volatile tests.
Test independent database storage creates a centralized location for storing test results of all types. The data files itself are stored, but general and typespecific metadata are stored in relational tables. With these two concepts Deltares is able to put all acquired data in a single database, which is effective and flexible. The result is a database that:

1. Permits deploying new types of tests without any database programming.
2. Is suitable for most geotechnical experiments from a simple laboratory test as the moisture content, field tests as CPT, but also complex geotechnical centrifuge tests.
3. Can find files based both on general metadata (project, sample and test identification, etcetera) as well as on information available for certain types of test only (typespecific metadata).
4. Validate each file stored, by checks on completeness (are all required elements present) and correctness (are all numerical elements between the specified limits).
5. Is accessible by all standard means, i.e. SQL.
6. Supports an external web portal to allow (external) clients direct and secure access to their own test results.

The implementation of the STREAM methodology in combination with the type of test independent database has lead to a flow of well defined intermediate data files during the test procedure itself and to a collection well described of test results in the permanent archive.

References

[1] P.E.L. Schaminée et. al., *STREAM: a method to facilitate efficient data exchange and archiving*, Proceedings ICPMG 2010, Zürich 2010, to be published.

[2] P.E.L. Schaminée et. al., *Geotechnical exchange format language a base for physical modeling data standard?*, Physical Modeling in Geotechnics – 6th ICPMG '06 (2006),Taylor & Francis Group London. 241-246.