Proceedings of DET2009
6th International Conference on Digital Enterprise Technology
Hong Kong
14-16 December 2009

DESIGN OF AN INFORMATION SYSTEM FOR METROLOGY CONTENTS

Carlo Ferri
c.ferri@bath.ac.uk

Jafar Jamshidi
j.jamshidi@bath.ac.uk

Craig Loftus
c.loftus@bath.ac.uk

Paul Maropoulos
p.g.maropoulos@bath.ac.uk

Department of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK

ABSTRACT

Dimensional and form inspections are key to the manufacturing and assembly of products. Product verification can involve a number of different measuring instruments operated using their dedicated software. Typically, each of these instruments with their associated software is in fact more suitable for the verification of a pre-specified quality characteristic of the product than others. The number of different systems and software applications to perform a complete measurement of products and assemblies within a manufacturing organisation is therefore expected to be large. This number becomes even larger as advances in measurement technologies are made. The idea of a universal software application for any instrument still appears to be only a theoretical possibility. A need for information integration is apparent. In this paper, a design of an information system for consistently manage (store, search, retrieve, search, secure) measurement results from various instruments and software applications is introduced. Two are the main ideas underlying the proposed system. First, by abstracting from the data structures and formats of measurement files, complexity and compatibility between different approaches to measurement data modelling is avoided. Second, the information within a file is enriched with meta-information to facilitate its consistent storage and retrieval. To demonstrate the designed information system, a web application is implemented.

KEYWORDS

Dimensional Measurement, Metrology, Relational Database, Entity Relationship Diagrams, Data Flow Diagrams, Web Application

1. INTRODUCTION

A large number of metrology systems and technologies are commercially available for the measurement and digitisation of parts and assemblies with various sizes, materials and geometries. Each of these systems may have specific advantages that can make them complementary to other systems. For instance a laser tracker system can provide high accuracy results while an articulated arm CMM (Coordinate Measurement Machine) can give more flexibility, and a network of Indoor-GPS (Infrared Global Positioning System, iGPS) can offer a seamless coverage of large volumes. For instance, the working volume of some measurement systems can be extended to cover the whole extension of a shop floor. However, these systems may not provide the accuracy needed by some of the inspection tasksto be performed within the shop floor volume. To overcome this limitation, frequently a number of more capable complementary instruments (an articulated arm, for instance) are deployed within the same area jointly with these shop-floor-wide systems (for instance an iGPS or a network of laser trackers).

In any case, even in more traditional production settings, typically a variety of measurement systems are used for the different inspection tasks that need to be performed. In spite of the traditional strive for the reduction of the variety of production tools within manufacturing environments, in practice quite rarely a single instrument or a limited number of instruments may be adequate or capable of taking all the measurements needed to gauge the many and often heterogeneous quality specifications imposed on a product by its designers. For instance, a shaft may have specifications of tolerances regarding the surface finishing of some of its elements (e.g. Ra=0.8 μm), the dimensions of others (e.g. diameter with tolerance h7) and the form of others again (e.g. co-axiality of two cylindrical surfaces of 0.1 mm).

The verification of conformity to tolerance specifications of heterogeneous nature as exemplified above is expected to contribute to broaden the range of measurement systems made available within a manufacturing organisation.

Typically, the data structure of the measurement result is instrument-dependent. For instance, the primary output of a laser scanner are Cartesian co-ordinates of clouds of points, the primary output of a 2D profilometer is a series of 2D profiles differently filtered whereas the primary output of a iGPS system are the spatial coordinates of a point, possibly moving.

The software applications governing different instruments need therefore to account for this diversity of data structures and verifications of tolerance they cater for. For this reason, digital instruments are typically supplied by their vendors with dedicated software applications to operate the instrument hardware and to process the measurement data coherently with both the raw measurement data generated and the information needed by the operator to perform a pre-assigned inspection task.

The multiplicity of software applications based on completely or partially different data structures are deemed to produce structures of information that are different even if they are stored with the same or compatible digital formats. For instance, the coordinates of a point measured with a laser tracker and the verification of perpendicularity of a plane A to a pre-specified datum plane B carried out with an articulated arm may be both saved in ASCII text files but the structure of these two files is likely to be different.

The complexity of maintaining a multiplicity of different software applications constitutes both an economic burden and a barrier to the current trend mentioned above towards joint-deployment (integration) of measuring instruments.

In this scenario, software vendors have emerged that have been trying to overcome this barrier with measurement software solutions equipped with customisable instrument interfaces supporting a number of different measurement systems within a shared software environment. Typical example of this is Spatial Analyzer (SA) from New River Kinematic, specialised in measurements of large-size products. However, two are the limitation envisaged in these kind of approaches. First, it is difficult to conceive that a single software application can provide interfaces covering adequately the constantly expanding range of instruments commercially available. Second, even assuming that an adequate coverage would be possible, it is expected that it may be difficult for such a software application to be economically justified, at least in a large number of manufacturing organisations.

In this investigation, a different approach has been proposed. The ambition to reduce the above mentioned multiplicity of software metrology applications to a single extensible application has been given up to. It has been preferred to abstract from such a multiplicity and from the variety of data structures that contribute to generate it. No attempt has been made to delve into the potentially intricate and variegate data structures stored into metrology files produced in output by the many different existing software applications. Each of these files has been instead treated as the minimum quantum of information to be managed (stored, searched, retrieved, secured, etc). This approach does not allow to reduce the variety of software applications within an organisation, but it enables and promotes a consistent use of metrology contents within the organisation itself.

In manufacturing environments, a second dimension of the complexity entailed with the management of metrology contents is the provision of quantitative evidence of the stability over the time of both the manufacturing and the inspection processes.

Measurement results need to be accessible not only for verifications of conformance of products to specifications, but also to calibrate the instruments, and to estimate the variability of both inspection and manufacturing processes. For all these kind of uses (part measurements, instrument calibrations, part variability, instrument variability), control charts and other statistical process control techniques can be used to give quantitative evidence of the stability of the processes involved .

These characteristics of an information system for data are an integral part of requirements for the competence to perform tests, calibrations and sampling prescribed by the BS EN ISO/IEC

17025:2005 [1]. This international standard sets the general principles for laying out a management system for quality, administrative and technical operations within any organisation performing testing and/or calibrations. The same standard (section 5.4.7) acknowledges the central role of data control when establishing a management system for quality within measurement laboratories. Therefore, with a view to an organisation implementing or working towards the implementation of this standard, it is essential that all data is recorded in a consistent manner in an information system. For example, when the state of calibration of a measurement system must be monitored and kept under statistical control, the Standard BS ISO 11095:1996 [2] provides methods for designing basic control charts. Yet, methods of this kind can be actually implemented in production environments only with an easy access to the historical data of the past measurements.

In this study, this kind of challenges has been tackled by enriching the often heterogeneous metrology contents stored in files with additional information which is purpose designed in the information system. This additional information specifies that part of the file contents that is reputed more important for the manufacturing organisation to implement its quality policies. This additional information being potentially referred also to the information encapsulated in the files, in this study is also referred to as meta-information (I.e. information about information) or metadata (I.e. data about data).

For instance, a file containing the results of a line scanner measurement of part A, can be enriched with the information regarding the CAD model file of part A, the operator who did the measurement, the environmental conditions at the moment of the measurement, the person who authorised the measurements, the specific instrument used and its calibration conditions, the annotations of the operator regarding details or anomalies when performing the measurement. All these metadata are meant to facilitate a consistent storage, search and retrieval of specific metrology content.

A third dimension of the complexity of the design of a system for metrology contents is constituted by the implications and relationship between metrology contents and organisation structure within the quality control or quality assurance department.

Within a quality control department it is expected that some sort of structure to be present with different tasks and responsibilities assigned to different people. For a small organisation, there may instead be a single person that accomplishes many tasks, with different levels of responsibility overlapped. In any case, people may interact with the metrology content in many different ways. For instance, an operator may be only interested in storing the results of measurements he performed with the instrument he/she is specialised in. But he/she may not be interested in knowing when another instrument that he/she is not authorised to use has been calibrated last time. There are instead information that need constantly to e easy accessible to anyone involved with quality control. For instance, the standard operating procedures (SOP) connected with establishing reference systems when carrying out measurements or the digital geometrical models of the parts to be inspected (i.e. CAD files).

In the proposed design, to address these aspects, roles have been defined and authentication has been requested as a main feature of the information system. In this way different people with different roles, when accessing the information system should be presented with some information common to all the roles and some other information typical of their role. Authentication also offers the potential of securing metrology contents.

A further characteristics for the information system to have a key role in the organisational structure is to enable concurrent access to the store information, possibly from access points that can be also remote one from the others. This especially holds for large-size product where inspection of products may require mobility of the operators. It would be also beneficial for organisations with multiple production sites.

To summarise, the intent of this study is not to propose a further format of standard for driving instruments or for analysing already available measuring data. Many different file formats (often proprietary) already exist and their complexity is driven by the richness of the geometrical information stored and of the features of the software application. The intent is rather that of building upon this existing variety of formats an information system based on standard wide-spread information technologies that facilitate the retrieval of information and the selective access to them.

In particular, when designing and implementing the system it has been deliberately made extensive and exclusive use of Free Software in accordance with the principle of the Free Software Foundation (cf. www.fsf.org) and the GNU project (cf. www.gnu.org).

In the following section, the specification of the functional requirements of the information system (i.e. a functional model of the data) is formally specified using the graphical language for building data flow diagrams (DFD). Then, in section three, the specifications of the data structures (i.e. a conceptual model of the data) is formally described by making use of the graphical language for building entity relationships diagrams (ERD). In section 4, a description of the implemented system is presented. The main conclusions are then drawn.

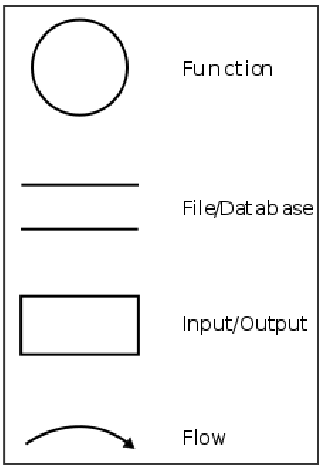
2. FUNCTIONAL MODEL

Data flow diagrams (DFD's) are a formal graphical representation of data aimed at representing the flow of data throughout their life within an information system. To obtain a DFD a graphical language is used. This graphical language uses four graphical objects (Function, file/database, input/output, flow), whose semantic values and syntactic use is mentioned in this section.

The function is the process that is performed by an agent such as a person or a software within or outside of the system. Each process can be detailed into its sub-system functions and each relevant number of systems can be in integrated to create a super system, again with its own inputs and outputs.

A function during its evolution may require or produce information which is represented by a flow element. The generated data in the form of a file or database is then stored for future use that can be done by the same agent or a different agent. Therefore a procedure should be in place to verify which agent has produced which piece of information and to ascertain to which other agents such information should be made accessible.

Information systems can be defined based on different abstraction level and requirements. For instance an information system can be used only as an access point to the data without any data manipulation possibilities. In this paper we consider this level as level zero of data flow diagram ($\boldsymbol{DFD_0}$). In the next level the functions of the information system at $\boldsymbol{DFD_0}$ are defined in more details, forming $\boldsymbol{DFD_1}$. The subsequent levels can be denoted as $\boldsymbol{DFD_2}$, …, $\boldsymbol{DFD_n}$, where $\boldsymbol{DFD_n}$ defines the final required level of information detail. In this approach the information flow in $\boldsymbol{DFD_i}$ can be taken into consideration in $\boldsymbol{DFD_{i+1}}$. Therefore an increasing level of detail can be achieved recursively.



Notations

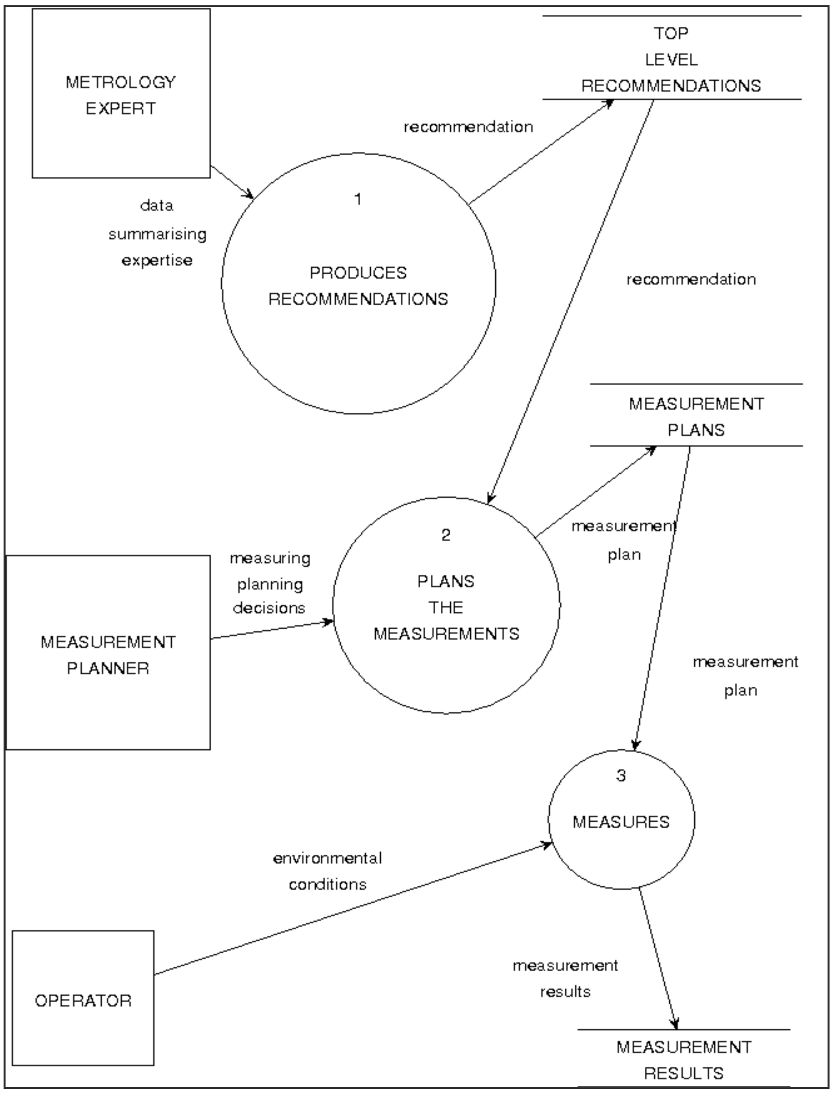


Figure 1 – DFD$_0$ context diagram

For the purposes of this investigation, a context diagram has been considered as sufficient to offer all the details needed to encompass the specifications of this information system. Three different external entities have been identified representing three distinct roles within the organisation. They are the metrology expert, the measurement planner and the instrument operator. The metrology expert is the ultimate responsible for quality assurance. This role entails making decisions regarding investments in measurement equipments to satisfy the demands of the measuring tasks generated by the manufacturing processes. They also have to provide to the measurement planner role high level recommendations regarding specific jobs that the quality assurance business unit within the company is requested to perform. These recommendations may range from the selection of specific measuring systems among the pool of those available to set the priority between contrasting required specifications. As an example, in a specific measurement task for an assembly a metrology expert may recommend the use of one single laser tracker to determine both the relative position of the parts and to detect the shape of the mating surfaces. He/She may concurrently issue some warnings of this choice such as, for instance, the sensitivity of the uncertainty of this type of measurements to angular displacements of the instrument head. If needed, he/she may therefore also recommend the usage of more than a single instrument in a network of instruments to counteract this expected problem. he/she needs to store this information for her/his own records and for the measurement planner to access.

The measurement planner receives the recommendations of the metrologist expert and on that basis produces a plan for accomplishing a specific measurement task. In this plan, he/she needs to establish the sequence in which the features to be inspected are to be measured. He/She also specifies the details on how each of the features needs to be measured. For instance, in the above example, he/she needs to use the laser tracker, as recommended by the metrology expert, but he/she needs to decide the actual position of the instruments, where the reference systems on the part or assembly are established. Also he/she needs to decide how many points are needed to be measured to identify the actual form of the matching surfaces. The produced measurement plan needed to be stored so that the relevant operator can access it.

The measurement system operator is the person within the organisation that actually knows how to interface with different kind of instruments. he/she is also aware of environmental factor (temperature gradients, dust free environment, etc.) that may affect one measurement system more sensibly than another. So he/she needs to access the measuring plan prepared by the measuring planner and needs to store the results of her/his measurements together with environmental conditions in which the measurements were performed.

The model described above is displayed in Figure 1. This figure was generated using “The Tool for Data and Event Flow Diagrams” (TEFD) which is part of the “Toolkit for Conceptual Modelling” (TCM) developed by Dehne and Van de Zandschulp [3] and released under General Public Licence (GPL). The three main roles identified in this analysis are separated. Yet, within an organisation two or even the whole three roles may be covered by the same person.

3. DATA MODEL

The data that the users of the information system are meant to store, update, delete and retrieve need to be defined. In this study, the formal approach that has been followed hinges on the concept of entity. An entity can be defined as an aspect of the real world that exists independently from the others and that can be uniquely identified. This definition is deliberately abstract so that it can be used to model any different situation where there is a demand for the introduction of a computerised information system. Entities can be thought of as nouns. For example, a customer, a product, a vendor are all entities. Typically, entities fall within four categories: people (e.g. vendor, customer, operator), things/concept (e.g. table, chair, wardrobe, song, theorem), events (e.g. sale), locations (e.g. office, shop).

Once the entities of interest have been identified, how each of them relate to all the others should be defined. Relationship is the concept that is introduced to describe the connections, if any, between two or more entities. Relationships can be thought as verbs linking two or more nouns. Examples of relationships are the purchases relationship between customer and product and the sings relationship between singer and song. Each relationship must be fully characterised. This means that information must be provided regarding how much of one entity relates to the other(s) entity involved in the relationship. The term cardinality of a relationship is used to express this quantitative information. For example, in a relationship supplies between vendor and product, one product can be supplied by many vendors but also one vendor can supply many products. This is an example of what is called a many-to-many relationship. Another example is the relationship belongs between finger and hand: one _nger belongs to one hand only, but one hand has many _ngers (unless someone has been particularly unlucky). This is an example of what is called a one-to-many relationship. In a similar way, examples of one-to-one relationships can also be fabricated.

After that the entities and the relationships between them have been completely identified, the characteristics or properties of entities and relationships need to be defined. These properties are called attributes of the entities and/or relationships. The attributes ultimately represent the information contents that the data modeller wants the information system to store, update, retrieve and delete for each of the previously defined entities and relationship. For example, attributes such as first name, last name, VAT number, address may be considered for a vendor entity, whereas VAT number and product ID (or product code) may be used for a relationship supplies between the entities vendor and raw material. Alternatively, an entity can also be seen as a collection of attributes.

Not all the attributes of an entity are the same. In fact, each entity can have a minimal set of attributes that uniquely identifies an occurrence of the entity. Such a minimal set of attributes is called a primary key. For example, for the entity vendor the primary key is identified in the VAT number. In fact, in an extreme case, two different vendors may have the same name, surname and business address, but they should be assigned two different VAT numbers. In some cases, it is possible to model the data so that for some entities it is not possible to identify any primary key. Such entities are called weak entities. They are not used in this report so their existence is mentioned, but no further details are provided. The end result of applying this formal method for the design of the structured data for the information system is a schema called Entity-Relationship Diagram.

The application of the method summarised above led to the diagrams shown in Figure 2 and Figure 3. These figures were generated using `Ferret the GNU Data Modeller' [4], a software tool for data modelling available at no cost with GPL licence. Entities are displayed with an E in the upper left end corner of blue (darker) large rectangles, whereas relationships have an R in the upper left corner of yellow (lighter) small rectangles. The entities of a relationship are connected to the relationship rectangle by solid black lines. The cardinalities are displayed in the middle of such lines. The attributes of each entity are listed inside the entity rectangle itself. The type of each attribute (double, integer, character, text, etc.) is also displayed. The primary key of an entity is then identified as that (or those) attribute (or attributes) which has (have) a continuous straight line just below its (their) name. The information that is most relevant to the metrologist has been grouped in the entity metrologist recommendations, which will also be referred to as `measurement task'. The names of the attributes are self-explanatory. The information describing the instruments, not surprisingly, has been modelled as an entity called instrument. When the metrologist is defining a specific measurement task, they have also to specify which instrument to use for that task. This is accounted for by the relationship involve instrument. The information that is most relevant to the person who is developing a measurement plan has been modelled as an entity called measurement plan. For each measurement task there must be one and only one measurement plan. This constraint has been modelled by defining a one-to-one relationship called are used in between metrologist recommendations and measurement plan. For accessing the information system each user need to be authenticate. For this reason, an entity Users has been created. Both the metrologist recommendations and the measurement plan need to have a clearly identified author. This is done by introducing the prescribe relationship between the entity Users and metrologist recommendations and the relationship produce between Users and measurement plan. It is believed then that there is reference information, such as the nominal coordinates of the main jig reference system that would be beneficially accessed by any people involved with measurement procedures, mostly for consultation. For this reason an entity called references material has been introduced. All these entities and relationships are illustrated in Figure 2.

The measurement plan may also refer to information regarding the features to be measured. On their turn, for each of these features the nominal co-ordinates of the points where they are going to be probed can be specified. For these reasons the entities feature and measurement plan have been introduced together with the relationships consists of and probed at which relate the first entity to measurement plan and the second to feature. Finally, the operator that is actually going to perform the measurement generates results which need to be dealt with by the information system. These pieces of information are modelled with the entity measurement result. The information regarding the operator that is accessing the system to store the measurement results needs also to be stored jointly with the measurement data. The ability of knowing who did a particular measurement is in fact a desired characteristic of the information system. For this reason, a relationship generate is modelled between the entities measurement result and Users. The entities and relationships described in this paragraph are displayed in Figure 3, where, for simplicity of the figure itself, relationships already illustrated in Figure 2 are no longer shown.

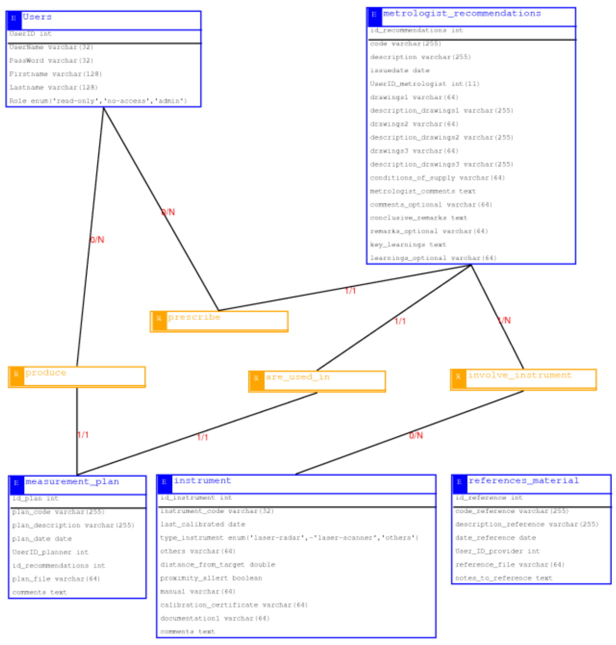


Figure 2 – ER diagram part one

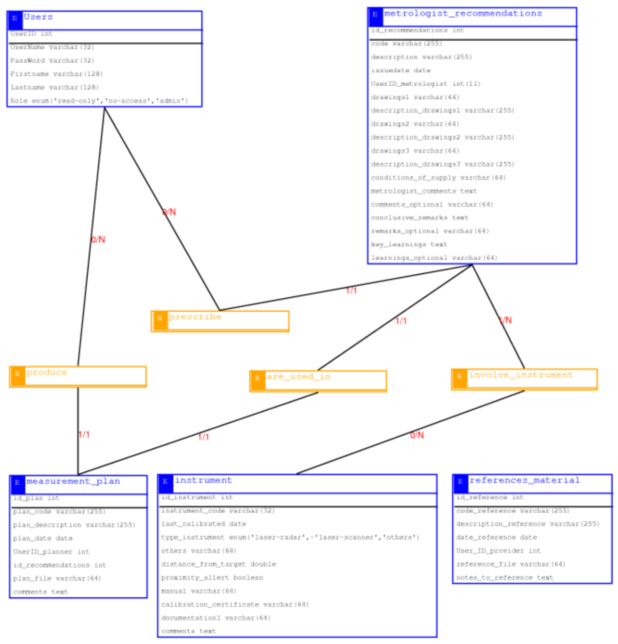


Figure 3 – ER diagram part two

4. IMPLEMENTATION

The model for the context and the data which were designed in the previous two sections have been used to implement a database. Among the many possible alternative data definition languages (DDL) and data manipulation languages (DML), Structure Query Language (SQL) was selected due to its widespread use. Among the many existent database management systems (DBMS), MySql was chose due to its speed, reliability, GPL licence and, last but not least, its availability at no cost [5].

All the development was carried out on a Debian Operating System, due to its GPL licence, its stability and the fair number of tools and application packages available within it (for more information see the Debian homepage [6]).

The data model presented in the previous section has been translated in a set of tables. Each entity is directly associated with a table where the attributes become the fields (or columns) of the table and each row is an instance of the entity (or record). Each individual row is then uniquely identified by the primary key field(s).

The relationships are dealt with in different ways according to their cardinality. On the one hand, in a one-to-many and one-to-one relationship between the entity A (source entity) and B (destination entity), the primary key of A is included among the attributes/fields of B. This enables the users of the database to formulate queries which involve the selection of those records of B that are related to a record of A with a specific primary key. For instance, a user of the database may wish to know all the information stored for all the measurement results that have been generate following a specific measurement plan.

On the other hand, in order to achieve the same wanted effect, in a many-to-many relationship between the entity A (source entity) and B (destination entity), a new table, say AB, whose fields are the primary keys of A and B have been introduced. In the data model designed two are the many-to-many relationships: involve instrument and probed at. So, eight tables are needed to describe the entities and two to describe the many-to-many relationships.

The creation of the ten tables has been described in a text file with the necessary SQL statements. In such a file, each of the ten tables has been completely defined in terms of fields, type of each field, primary key(s) of each table and constraints on the fields.

The actual creation of the database in the MySQL DBMS has been accomplished by using the command line interface mysql client, which is provided with the standard installation of MySQL. Also, during the several iterations needed for tuning the database, the graphical user interface PhpMyAdmin released under GPL licence has been used. For more information regarding PhpMyAdmin, please refers to the website of the community that has developed this application (cf. [7]).

Once the designed database has been created within the DBMS MySQL, it can be accessed via mysql client. Yet, this may require a very minimum of knowledge of some commands of such an interface. To enhance the usability of the created database, a graphical user interface may be designed and implemented. In doing so, it is highly desirable that two main characteristics of the command line interface are preserved. First, the ability of the users to access the database from remote PC’s connected via a LAN or the internet to the machine where the DBMS is running. Second, the ability of the users to access the application concurrently.

Moreover, it would be highly desirable that this interface application does not require any customised software to be installed on all the PC's of the users. It would then be also highly desirable for the interface to the database to be completely independent from the operating system running on the users' machines (any Linux, Unix, Apple, Sun-Solaris, any Microsoft Windows, etc.). For these reasons, it was decided to implement a web application which enables any web browser, commonly found on any PC, to become the interface which the DBMS can be accessed by. To build this web application, it was used Xataface, a framework for building data-driven applications [8]. Xataface is based on the scripting language PHP [9], the template engine Smarty [10], the cascade style sheets (CSS) and javascript programs from the Plone content management systems [11]. All of these pieces of software are available under GPL licence at no cost. Moreover, for the users to be able to requests the execution of the Xataface scripts using their browser, a web server program has also been used. Due to its large popularity, its free software licence policy and its coming at no cost, the web server Apache2 was chosen [12].

The combined usage of these pieces of software is so popular that the acronym LAMP is often used in the public domain to describe it. This stands for Linux, Apache, MySQL and Php or Python or Perl. Overall, the resulting approach can also be described as a three-tier architecture model. The first is the database tier, which consists of the DBMS that manages the database storing the data users create, update, delete and query. Built upon it, there is the middle tier constituted by all the scripting programs (PHP and Javascripts) that implement the logic of the application. This middle tier needs also to convey data between the other two tiers. So the web server, Apache2 in this case, is another constitutive element of this tier. Due to the multiplicity of functions that the tier performs, it is perhaps the more complex. The third tier is also called client tier and it is built upon the middle tier. Usually the third tier is a web browser that interacts with the application implemented in the middle tier. Access to the database tier is made by the application scripts of the second tier when it is executed by the commands transmitted to it by the users that send requests to the web browser via the web server. The three-tier architecture model is illustrated in Figure 5, where a typical flow of information between the various elements of each tier is also shown.

This figure has been enriched with the considerations from the authors of this paper starting from a similar figure in Williams and Lane [13]. Each software tier may reside on the very same machine or on separate machines. For instance the DBMS MySQL may be installed on machine A, Xataface files and the Apache2 web-server may be installed on Machine B and the users may be using machine C, D, E etc. Each of these machines must then be somehow connected to the others either in a local area network (LAN) or via the internet. Typically, the database tier and the middle tier reside on the same machine, whereas the user-base accesses that machine either via a LAN or the internet. As an example, a screenshot of the application is displayed in Figure 5.

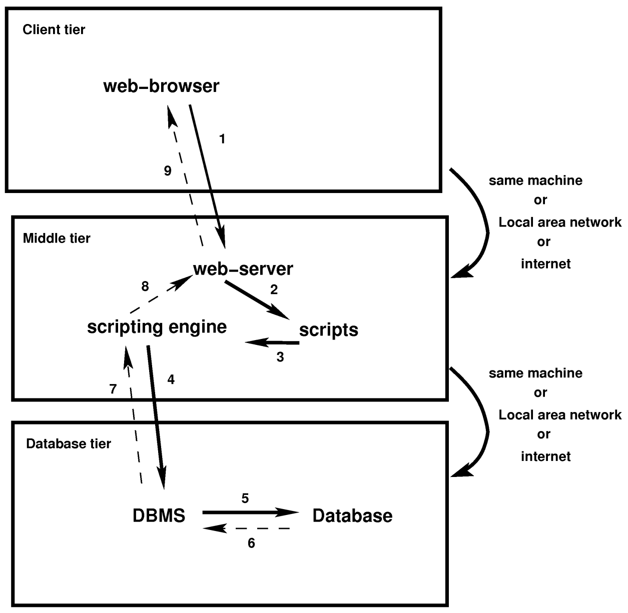
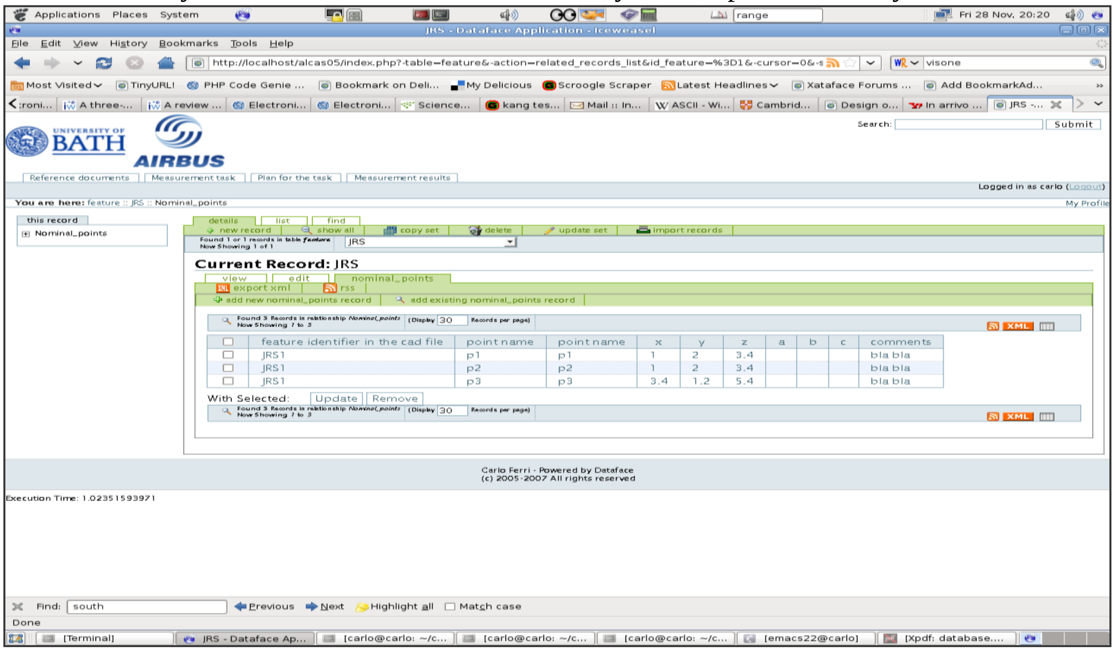


Figure 4 – Three-tier architecture model

5. CONCLUSIONS

An information system for measurement related activities within a manufacturing organisation has been considered. The main objective of this system is to enable a pervasive use of consistent information throughout the organisation so that measurement activity information can be easily stored, accessed and shared within the organisation. Application of data intensive procedures to assess the state of control of measuring systems can be almost effortlessly put in place. The main requirement that the authors have endeavoured to satisfy is independence from the large variety of different and most often proprietary software applications commercially available for measurement analyses and measuring systems control. This was achieved by abstracting from specific measuring tasks to a level where the output of a proprietary software in text or binary formats are considered as atomic basic blocks for the this information system.

A conceptual data model and an analysis of the interfaces of this information system have been proposed using an Entity-relationship (ER) model and a data flow diagram (DFD), respectively. The design of the DFD and ER have then been implemented in a software application. Such an application has been developed in a three-layer web-based architecture. The first layer is a database management system (DBMS), the second is a web server supporting scripting language capabilities, and the third is a web browser. All the information technology used in the implementation are released under a no-cost General Public Licence (GPL) from the Free Software Foundation, Inc. The selected architecture of the application enables the provision of access to the data in multiple sites of the organisation and also at external locations, as long as they are served by an internet connection. Multiple users can access concurrently the developed system and the data from the same or different computer systems regardless of operating systems, provided they have a web browser.


Applications Places System
range
Fri 28 Nov, 20:20
JRS - Dataface Application - Iceweasel
File Edit View History Bookmarks Tools Help
http://localhost/alcas05/index.php?-table=feature&-action=related_records_list&id_feature=%3D1&-cursor=0&-s
visone
Most Visited TinyURL! PHP Code Genie ... Bookmark on Deli... My Delicious Scroogle Scraper Latest Headlines Xataface Forums ... Add BookmarkAd...
:roni... A three-... A review ... Electroni... Electroni... Science... kang tes... Mail :: In... ASCII - Wi... Cambrid... Design o... In arrivo ... JRS -...
Search: Submit
Reference documents
Measurement task
Plan for the task
Measurement results
Logged in as carlo (Logout)
You are here: feature :: JRS :: Nominal_points
My Profile
this record
Nominal_points
details list find
new record show all copy set delete update set import records
Found 1 or 1 records in table feature Now Showing 1 of 1
JRS
view edit nominal_points
export xml rss
add new nominal_points record
add existing nominal_points record
Found 3 Records in relationship Nominal_points Now Showing 1 to 3 (Display 30 Records per page)
XML
With Selected: Update Remove
Found 3 Records in relationship Nominal_points Now Showing 1 to 3 (Display 30 Records per page)
Find: south Previous Next Highlight all Match case
Done
[Terminal] JRS - Dataface Ap... [carlo@carlo: ~/c... [carlo@carlo: ~/c... [carlo@carlo: ~/c... [emacs22@carlo] [Xpdf: database....

Current Record: JRS

	feature identifier in the cad file	point name	point name	x	y	z	a	b	c	comments
☐	JRS1	p1	p1	1	2	3.4				bla bla
☐	JRS1	p2	p2	1	2	3.4				bla bla
☐	JRS1	p3	p3	3.4	1.2	5.4				bla bla

Carlo Ferri - Powered by Dataface
(c) 2005-2007 All rights reserved

Execution Time: 1.02351593971

Figure 5 – A screen-shot of the implemented software application

ACKNOWLEDGMENTS

The work has been carried out as part of the Composite Wing Research within the Airbus Composite Structures Development Centre and the LVMA project within IdMRC of the University of Bath under EPSRC grant number EP/E00184X/1. Amir Kayani and Kevin Foster have been the main contacts within Airbus.

REFERENCES

[1] BSI - British Standards Institution. BS EN ISO 17025:2005.General requirements for the competence of testing and calibration laboratories, 2005.

[2] BSI - British Standards Institution. BS ISO 11095:1996. Implementation of ISO 11095:1996. Linear calibration using reference materials, 1996.

[3] Frank Dehne and Henk Van de Zandschulp. The Toolkit for Conceptual Modelling. University of Amsterdam and University of Twente, 2003. URL http://www.cs.utwente.nl/ tcm. Last accessed on 17 October 2008.

[4] Anonymous. Ferret the GNU data modeller. Ferret project development group. URL http://www.gnuferret.org last accessed on 14 November 2008.

[5] Anonymous. MySQL website. MySQL AB. URL http://www.mysql.com last accessed on 14 November 2008.

[6] Anonymous. Debian Project homepage. URL http://www.debian.org last accessed on 14 November 2008.

[7] Anonymous. The PhpMyMyAdmin project. URL http://www.phpmyadmin.net last accessed on 14 November 2008.

[8] Anonymous. Xataface website. URL http://www.php.net/ last accessed on 14 November 2008.

[9] Anonymous. php website. URL http://www.php.net/ last accessed on 14 November 2008.

[10] Anonymous. Smarty template engine. URL http://www.smarty.net/ last accessed on 14 November 2008.

[11] Anonymous. Plone content management system. URL http://plone.org/ last accessed on 14 November 2008.

[12] Anonymous. Apache http server project. URL http://httpd.apache.org/ last accessed on 14 November 2008.

[13] Hugh E Williams and David Lane. Web database applications with PHP and MySQL. O'Reilly, Sebastobol, CA, 2002.