Web and Knowledge-based Decision Support System for Measurement Uncertainty Evaluation

Peng Wei

A thesis submitted in partial fulfillment of the requirements of

Brunel University

for the degree of Doctor of Philosophy

April 2009

Abstract

In metrology, measurement uncertainty is understood as a range in which the true value of the measurement is likely to fall in. The recent years have seen a rapid development in evaluation of measurement uncertainty. *ISO Guide to the Expression of Uncertainty in Measurement* (GUM 1995) is the primary guiding document for measurement uncertainty. More recently, the *Supplement 1 to the "Guide to the expression of uncertainty in measurement" – Propagation of distributions using a Monte Carlo method* (GUM SP1) was published in November 2008. A number of software tools for measurement uncertainty have been developed and made available based on these two documents. The current software tools are mainly desktop applications utilising numeric computation with limited mathematical model handling capacity.

A novel and generic web-based application, web-based Knowledge-Based Decision Support System (KB-DSS), has been proposed and developed in this research for measurement uncertainty evaluation. A Model-View-Controller architecture pattern is used for the proposed system. Under this general architecture, a web-based KB-DSS is developed based on an integration of the Expert System and Decision Support System approach.

In the proposed uncertainty evaluation system, three knowledge bases as sub-systems are developed to implement the evaluation for measurement uncertainty. The first sub-system, the Measurement Modelling Knowledge Base (MMKB), assists the user in establishing the appropriate mathematical model for the measurand, a critical process for uncertainty evaluation. The second sub-system, GUM Framework Knowledge Base, carries out the uncertainty evaluation process based on the GUM Uncertainty Framework using symbolic computation, whilst the third sub-system, GUM SP1 MCM Framework Knowledge Base, conducts the uncertainty calculation according to the

GUM SP1 Framework numerically based on Monte Carlo Method.

The design and implementation of the proposed system and sub-systems are discussed in the thesis, supported by elaboration of the implementation steps and examples. Discussions and justifications on the technologies and approaches used for the sub-systems and their components are also presented. These include Drools, Oracle database, Java, JSP, Java Transfer Object, AJAX and Matlab.

The proposed web-based KB-DSS has been evaluated through case studies and the performance of the system has been validated by the example results. As an established methodology and practical tool, the research will make valuable contributions to the field of measurement uncertainty evaluation.

Acknowledgements

Firstly, I would like to sincerely thank my PhD supervisor, Doctor Qing Ping Yang, for his invaluable support and guidance throughout the course of this research, without which the completion of this PhD project would not have been possible.

My deepest gratitude also goes to Prof. Barry E. Jones for his supervision and motivation throughout this project.

I would also like to extend my deep appreciation to Brunel University for the financial support for this study. I am grateful for the support and encouragement from my friends and colleagues, Yang Hu and Yi Li, throughout this research. I am deeply appreciative of the valuable comments and assistance from Quan Zhang.

Deep and special thanks also go to my parents and my wife Tiantian Li, for their love, support and understanding.

Table of Contents

List of Abbreviation

Abbreviation	*Full Format of Abbreviation*
3DMM	Three Dimensional Modelling Module
AJAX	Asynchronous JavaScript and XML
ANN	Artificial Neural Network
ASP	Active Server Pages
BIPM	International Bureau of Weights and Measures
BNN	Biological neural network
CIPM	International Committee for Weights and Measures
CSS	Cascading Style Sheets
DBMS	Database Management System
DGMS	Dialog Generation and Management system
DSS	Decision Support System
DOM	Document Object Model
EA	European Co-operation of Accreditation
EJB	Enterprise JavaBeans
ES	Expert System
EUROMET	European Collaboration in Measurement Standards
GUM	Guide to the Expression of Uncertainty in Measurement
GUM SP1	Supplement 1 to the "Guide to the expression of uncertainty in measurement" — Propagation of Distributions Using a Monte Carlo Method
GUM KB	GUM Framework Knowledge Base
HIS	Hybrid Intelligent Systems
IEC	International Electro Technical Commission
IFCC	International Federation of Clinical Chemistry
IFPS	Interactive Financial Planning System
ILAC	International Laboratory Accreditation Cooperation

ISO	International Organization for Standardization
IUPAC	International Union of Pure and Applied Chemistry
IUPAP	International Union of Pure and Applied Physics
JCGM	Joint Committee for Guides in Metrology
JSCL-Meditor	Java Symbolic Computing Library and Mathematical Editor
JSP	Java Server Page
KB-DSS	Knowledge-Based Decision Support System
MBMS	Model-Base Management System
MCM	Monte Carlo Method
MCM KB	MCM Framework Knowledge Base
MM KB	Modelling Knowledge Base
MSMM	Measurement Standards Modelling Module
MVC	Model-View-Controller
NIST	National Institute of Standards and Technology
NPL	National Physical Laboratory
OIML	International Organization of Legal Metrology
PDFs	Probability Density Functions
PHP	Hypertext Preprocessor
QFD	Quality Function Deployment
RDBMS	Relational Database Management System
SI	International System of Units
SUMM	SI Unit Modelling Module
TO	Transfer Object
UIMM	User Input Modelling Module
VIM	International Vocabulary of Basic and General Terms in Metrology
XHR	XMLHttpRequest

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Overview

Measurement is the assignment of numbers or other symbols, by an objective and empirical process, to attributes of objects or events of the real world, according to a rule or set of rules (JCGM, 2008). Metrology is "the science and art of measurement, embracing both experimental and theoretical determinations at any level of uncertainty in any field of science and technology", defined by International Bureau of Weights and Measures (BIPM). As an old science, it has evolved over many centuries and is of fundamental importance in experimental sciences, manufacturing, engineering, and industry.

Nowadays, metrology is distinguished by the development of new measurement techniques, instrumentation and procedures, to satisfy the ever-increasing demand for higher level of accuracy, increased reliability and rapidity of measurements (Howarth, Redgrave, 2008). It covers three main activities:

- To define the internationally accepted units of measurement.
- To realise the units of measurement by scientific methods.
- To establish the traceability chains by determining and documenting the value and accuracy of a measurement and disseminating that knowledge.

Metrology also can be divided into three categories according to different levels of complexity and accuracy:

- *Scientific metrology* deals with the organisation and development of measurement standards, establishment of the quantity systems, unit systems, units of measurement etc., and transferring of traceability from these standards to users in the society. As the branch of the *scientific metrology*, *fundamental metrology* generally signifies the highest level of accuracy within a given field.

- *Industrial metrology* is the application of measurement science to manufacturing and other processes and their use. Industrial metrology must ensure the suitability and satisfactory functioning of measurement instruments used in production and testing processes in the industry. It also deals with the instruments' calibration and the quality control of measurements. Through these means, it ensures the quality of life of the products.
- *Legal metrology* is associated with regulatory requirements of measurements and measuring instruments for the purpose of protection of public health and safety, protection of the environment, maintenance of adequate taxation, protection of consumers's interests and fair trade. Legal metrology is usually applied in areas where activities of measurement have influence on the transparency of the commercial activities or where there is a requirement for legal certification for the measuring instrument.

In metrology, measurement uncertainty describes a region about an observed value of a physical quantity which is likely to enclose the true value of that quantity. The International Organization for Standardization (ISO) defines the term "uncertainty of measurement" as follows (JCGM, 2008):
"non-negative parameter characterizing the dispersion of the quantity values being attributed to a measurand, based on the information used."

As one of the most critical parts of the modern development of metrology, the concept of measurement uncertainty has been widely discussed over the years since the late 1970s, when the classical Gaussian error calculus has been considered incomplete.

At that time, many different approaches of evaluating and expressing the uncertainty of measurement results had been used.

In 1977, BIPM was asked by the highest authority in the field of metrology, the International Committee for Weights and Measures (CIPM), to coordinate among the various national metrology institutes and put in place a commonly agreed documentation with regards to the expression of measurement uncertainty (NIST Technical Note 1297).

In the 1980s, firstly BIPM assigned a group to propose a recommendation for the expression of measurement uncertainty that would be accepted at an international level. The recommendation, officially named Recommendation INC-1 (1980), was put in place in 1980. It was then approved by CIPM in 1981 and reaffirmed in 1986. The recommendation was a critical first step in developing internationally recognized procedure for expressing measurement uncertainty, although it was a brief documentation only and demanded a large amount of further detailed explanation.

Therefore, CIPM assigned ISO to carry out the next step, which is to establish a detailed guide on the basis of the recommendation by BIPM. The purpose of the detailed guide is to provide complete information on how measurement uncertainty is expressed and a platform for comparison of measurement results.

This detailed document, *Guide to the Expression of Uncertainty in Measurement* (the GUM), was firstly published in 1993 and corrected and reprinted in 1995 by ISO. It was published in the name of seven international organisations that were involved in the establishment of this document: BIPM, ISO, International Electro technical Commission (IEC), International Federation of Clinical Chemistry (IFCC), International Union of Pure and Applied Chemistry (IUPAC), International Union of Pure and Applied Physics (IUPAP) and International Organization of Legal Metrology (OIML).

GUM sets “general rules for evaluating and expressing uncertainty in measurement that can be followed at various levels of accuracy and in many fields, from the shop floor to fundamental research” (BIPM, 1993). The publishing of GUM proved to be an international success and was accepted widely around the world. It has been adopted by a large number of entities and institutes in the metrology industry as well as those of other related industries. UK National Physical Laboratory (NPL) and European Collaboration in Measurement Standards (EUROMET) are among a large number of organisations who have adopted GUM. Due to its comprehensiveness and detailedness, GUM became the basis of a variety of standards and regulations at both regional and international levels.

Now, GUM is regarded as the primary documentation in evaluation of measurement uncertainty. It has been serving an essential role in metrology science and theories, standardisation, calibration, as well as testing lab accreditation.

In 1997, Joint Committee for Guides in Metrology (JCGM) was established for the purpose of maintaining and updating of the GUM. This newly established organisation also bears the responsibility of defining the other important terms in the area of measurement, the *International Vocabulary of Basic and General Terms in Metrology* (VIM). The seven international organizations listed above: BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML, and the International Laboratory Accreditation Cooperation (ILAC) are all members of this new organisation.

A few working groups have been convened and work to supplement and extend GUM and its applications is ongoing. Supplementary and supporting documents are scheduled to be published by the organisation. The first document of such kind, *Evaluation of measurement data — Supplement 1 to the "Guide to the expression of uncertainty in measurement" — Propagation of distributions using a Monte Carlo method* (GUM SP1) was published in 2008. The document deals with the propagation of distributions and emphasizes the use of Monte Carlo Method (MCM) for measurement uncertainty evaluation. Further discussion with regards to concepts covered in and the principles of GUM and GUM SP1 is carried out in details in Chapter 2.

In this research, a novel web-based application, which is used for evaluating and expressing measurement uncertainty in accordance with both GUM uncertainty framework and MCM simulation, utilising Knowledge-Based Decision Support System (KB-DSS), is investigated and demonstrated. The proposed KB-DSS integrates Decision Support System (DSS) and Expert System (ES) technology. Further discussion of the concepts of DSS and ES will be explained in details in Chapter 3. The focus of this research will be mainly on the development of a web-based KB-DSS on the basis of proposed Measurement Modeling Knowledge Base (MMKB), GUM Framework Knowledge Base (GUM KB) and MCM Framework Knowledge Base (MCM KB). The architecture of the web-based KB-DSS approach and procedures are presented and the great potential and powerful features of this approach for measurement uncertainty evaluation is demonstrated through measurement examples.

The next section highlights the problems that initiated this study and discusses the necessity for this research.

1.2 Problem Statement

Along with the development of computer science and technology, several software tools have become academically or commercially available to facilitate the evaluation and automate the computation. The table below lists some popular and widely used software tools, some of which perform the classical GUM approach and some are able to apply the latest GUM SP1 MCM simulations:

Software & Add-in Package Tools	**Developer**
NPLUnc	NPL, UK
GUM Workbench	Metrodata GmbH, Germany
DFM-GUM	Danish Institute of Fundamental Metrology
Crystal Ball for Uncertainty	Oracle, USA
Timeko Uncertainty	Timeko, UK
Uncertainty Pro	ChemSW, Canada
Uncertainty Manager	VWR International Ltd, UK
Uncertainty Analyzer	Integrated Sciences Group, USA
@Risk 5.5	Palisade Corporation, USA

Table. 1.1 Current Software & Add-in Package Tools

The author has discussed a few limitations of the current software tools in the previous work (Wei, 2007):

- Limited ability to handle measurement mathematical models

  For those users who are familiar with the measurement mathematical model, it is of critical significance if they can establish the mathematical expression in the software tools without too many restrictions, regardless of the complexity of the model or how the model is entered; for those users who are not familiar with the measurement model, a type of approach which can help them establish or select the correct measurement model is critical.

- Limitations in cross-platform compatibility (i.e., Windows, Mac, Linux etc.), accessability, cost-effectiveness and ease of system maintenance due to traditional desktop approach

- Mainly rely upon pure numerical computation (does not support any symbolic computation). The author has discussed the limitations of numeric computation in evaluation of measurement uncertainty in the previous work.

A comparison was made on the basis of user-friendly features and facilities for laboratory personnel (Jurado, Alcazar, 2007). Currently, the critical work is to develop a kind of system which is not only more user-friendly for laboratory technicians but also able to promote and extend the potential use of the application among unskilled end-users via the internet technology.

Based on the problems highlighted above, the aims and objectives of this study are established and the further explanations will be given in the next section.

1.3 Aim and objectives

The aim of this research is to introduce a novel approach for evaluation of measurement uncertainty via a web-based KB-DSS according to the GUM and GUM SP1. The major objectives are:

1. To develop an advanced measurement modeling knowledge base in the proposed KB-DSS. This knowledge base is established on the support of *scientific metrology measurement standards* (EURAMET) and the *International System of Units* (SI) in modern metric system of measurement (BIPM).
2. To develop another two calculation knowledge bases in the proposed system, according to the approach of GUM framework and GUM SP1 MCM framework, to evaluate the measurement uncertainty.
3. To investigate and establish a real-time and user-friendly web architecture for the KB-DSS based on the Internet technology. To develop the system architecture based on the concept of Model-View-Control web-application architecture.
4. To develop a suitable data transfer architecture which is able to transfer multiple data elements across different system components and the subsystems.
5. To develop a user-friendly User Interface (UI) to assist user in sending the request and getting the real time response via the web page.

In order to achieve the above aim and objectives, the following tasks are required to be performed:

1. Literature review: Review relevant theories and other literature in order to：
   - Understand the concept of measurement, measurement error and uncertainty in measurement.
   - Apply the general approach for evaluating the measurement uncertainty based on GUM framework and GUM SP1.
   - Understand the routine procedures of GUM framework and GUM SP1.

- Evaluate the approaches of GUM framework and GUM SP1 MCM framework.
- Understand the methodology of both Decision Support System and Expert System.

2. Identify and examine both the advantages and the limitations of existing approaches of evaluating and expressing measurement uncertainty based on GUM framework and SP1 MCM framework. Studies are to be performed to understand and evaluate the currently available computing software tools as mentioned in previous sections and identify the problems.

3. Determine and establish the appropriate methodology and tool to facilitate the evaluation of measurement uncertainty by developing a novel web-based KB-DSS.

4. Implement the approach and test the procedures with example data.

5. Perform detailed measurement cases in the proposed system and evaluate the implementation of the proposed system.

The next section generally describes the structure of this thesis.

1.4 Thesis Structure

This thesis consists of seven chapters in total. Chapter 2 discusses all the background of measurement uncertainty and the current approaches to evaluate the uncertainty in measurement. Detailed discussions of the approaches will be provided, together with their limitations. Chapter 3 specifically deals with the DSS and ES as part of the literature review. Chapter 4 particularly explains the proposed methodology including the previous research work. Chapter 5 explains the design and the implementation of the web-based KB-DSS including the design and implementation of three knowledge bases, the system framework, the UI and the data transfer object. Chapter 6 gives a detailed evaluation procedure of each component of the system. Measurement cases study is demonstrated through the system. The performing results of this system are presented. Chapter 7 elaborates the conclusion of this research and further discusses the contribution of this research to knowledge. Future research and the extended study of this issue will also be highlighted in this chapter.

Chapter 2 Evaluation of Measurement Uncertainty

2.1 Introduction

A measurement is a quantitative statement of the property of a specific object, examples including weight or length of an object. In most cases, a kind of instrument is needed to give the measurement. However, it is worth noting that not everything related to length, weight etc of something is considered measurement. For example, comparing the weight of two objects is not measurement.

A number of systems of unit have been used and applied in human history, including the Imperial system and the Metric system, the latter has an advantage that for every quantity there is a single base unit and all other units are decimally related to that base unit. The modern International System of Units is a revision of the metric system and is now the most widely used measurement system in the world.

This chapter reviews the development of measurement and measurement uncertainty, focusing on the approaches utilised to evaluate and express measurement uncertainty. Two widely used approaches, GUM uncertainty framework and GUM SP1 MCM Framework are discussed with detailed explanations of their implementation. A brief comparison of them is also highlighted.

2.2 The International System of Units

History of SI at a glance

As mentioned in earlier sections, the International System of Units is the modern form of metric system. It is now widely used by most of the countries of the world, with exception of Burma, Liberia and United States. Its application is widely found in scientific, social and commercial areas. Globally, the System is also known as SI, as abbreviation from French.

The history of SI is thought to have begun in late 1790s, when the decimal Metric System was coined. Gauss was the most important advocate of the Metric System and did a substantial contribution to the promotion of the System in 1830s. Another contribution by Gauss in this area is that he made the first absolute measurement of the earth's magnetic force based on the quantities of length, mass and time. Later, together with Weber, he also extended this to electrical area.

In 1860s, the British Association of Advancement of Science (BAAS) took the lead in developing the application of Metric System in the electricity and magnetism areas. They also started the research on a system which is based on base units and derived units. The CGS system was introduced by BAAS in 1874. The CGS system is based on centimetre, gram and second, and it was an important milestone of the later development of physics. In 1880s, the units including ohm ad volt were introduced approved.

In 1889, the 1st General Conference on Weights and Measures (CGPM) approved the use of the meter and the kilogram, which laid the ground for a system based on meter, kilogram and second. It is important to note that until then, the systems of units, including this CGS system, are all three-dimensional mechanical systems, "mechanical" meaning the systems are based on mechanical units of length, mass and time.

Beginning from the 1900s, a further base unit (electric unit) was introduced to be combined with the three-dimensional system, to form a four-dimensional system. The four-dimensional system based on meter, kilogram, second and ampere was approved by International Committee for Weights and Measures (CIPM) International Committee for Weights and Measures (CIPM) in 1946. By 1971, three further base units, kelvin for thermodynamic temperature, candela for luminous intensity and mole for amount of substance, were introduced into the system. In 1960, the name of International System of Units (SI) was officially introduced at the 11th CGPM. The seven base units mentioned became the basis of the System.

SI architecture

The SI is made up by two parts, the base units and the derived units.

a) Base Units

There are seven base units in the SI, each of them represents a unique physical quantity, independent from each other. The Table 2.1 outlines the seven base units.

Quantity	Symbol	Name
length	**m**	meter
mass	**kg**	kilogram
time	**s**	second
electric current	**A**	ampere
thermodynamic temeperature	**K**	kelvin
luminous intensity	**cd**	candela
amount of substance	**mol**	mole

Table 2.1 Base Units in SI

b) Derived Units

The SI includes derived units which are derived from the seven base units. As shown in Table 2.2, several of the derived units (the second column of the table) are expressed in SI base units, meaning without special name. Apart from them, all other derived units are with special names and symbols.

Quantity	Symbol	Name
area	m^2	square meter
volume	m^3	cubic meter
speed, velocity	m/s	meter per second
acceleration	m/s^2	meter per second squared
wave number	m^{-1}	reciprocal meter
mass density	kg/m^3	kilogram per cubic meter
specific volume	m^3/kg	cubic meter per kilogram
current density	A/m^2	ampere per square meter
magnetic field strength	A/m	ampere per meter
amount-of-substance concentration	mol/m^3	mole per cubic meter
luminance	cd/m^2	candela per square meter
mass fraction	kg/kg = 1	kilogram per kilogram

Table 2.2 Examples of SI Derived Units

2.3 Measurement Error

Measurement error is the difference between an observed value and its true value. Repeated measurement of the same quantity can often get the observed value closer to its true value but the error is unlikely to be completely eliminated.

There are a large number of causes for measurement errors, which include errors due to measuring instruments (e.g. zero error and adjustment errors), interactions between the measurand and measuring instruments (e.g. loading error), operator (e.g. reading error) and the environment (e.g. humidity, temperature).

Measurement errors can often be categorised into three kinds, random error, systematic error and parasitic error (mistake).

2.3.1 Random Error

Random errors relate to the randomly different results when the same measurement is repeated. The random errors cause the measurement results to randomly away from the true value. Essentially random errors are unpredictable, they are often due to factors that are difficult to control, for example fluctuations in the result readings of the instrument, the operator's interpretation of the reading as well as the instability of the environment.

Although every measurement has random error, it can be estimated and reduced through statistical analysis and experiments. A method commonly used to reduce random error is to repeat the same measurement for a number of times and take the arithmetic mean value as the final measurement result. The more times the measurement, the closer the mean value is to the true value.

2.3.2 Systematic Error

Systematic error, also known as bias, always happens when the measurement is repeated and often occurs in a predictable pattern and direction. Commonly known systematic errors include constant systematic errors, linear systematic errors, periodic systematic errors and complex systematic errors (Zemel'man, 1985).

Systematic errors can be caused by imperfection of calibration of measurement instruments or measuring methods. Some systematic errors, constant systematic errors in particular, can be detected by comparing the measurement results with different instruments or calibration. If the exact cause is known, the error can be eliminated. However, it is more often that the cause, size and direction of the systematic errors are unknown, which can make the systematic errors difficult to remove. Commonly used methods to minimise systematic errors include theoretical correction, calibration of the measuring instrument and optimising the procedure of the measurement/experiment (e.g. compensation method and reversal method).

2.3.3 Parasitic error

Parasitic errors usually have large magnitudes and are likely to cause significant distortion of the mean and standard deviation of the results. Once the parasitic errors are detected, they should be rejected. Since it is common in manufacturing to take a single measurement of a measurand, there is no way of detecting such errors during measurement. We must therefore rely on preventive actions and minimise the situations where such errors are likely to occur. Procedures are all important and will include such mundane matters as cleanliness, training and routine maintenance.

2.4 Measurement Uncertainty

The definition of the term "uncertainty of measurement" by ISO is quoted in Chapter 1. Quality standard PD6461-3 (1995) defines measurement uncertainty as "Result of the evaluation aimed at characterizing the range within which the true value of a measurand is estimated to lie, generally with a given confidence."

Every measurement has a margin of doubt, uncertainty. Uncertainty comes from a large number of sources and it is not avoidable in reality. Only when the uncertainty is quantitatively evaluated and stated a measurement result is considered complete (Bell, 1999). The evaluation of uncertainty is essential in determining whether the measurement result addresses its intended purposes appropriately.

Measurement uncertainty can be understood as a range or margin in which the true value of the measurement is likely to fall in. It has a number of sources, including form error, wear/aging of the measuring instrument, calibration uncertainty, imperfection of the measuring method, personal bias when reading, the environment (temperature, air pressure, humidity etc). Generally, each source for error is a source of measurement uncertainty (Bell, 1999).

2.4.1 Difference between Measurement Error and Measurement Uncertainty

It is very important that the measurement uncertainty and measurement error are distinguished clearly from each other. Measurement uncertainty is a quantified range of doubt with regard to the measurement result, whereas the measurement error is the difference between the observed or calculated result and the true value of the measurand.

2.4.2 Importance of Measurement Uncertainty

The uncertainty of measurement is important in senses of calibration, test and indication of the quality of the measurement result.

- As mentioned above, every measurement is prone to uncertainty therefore a measurement result is only complete when it is accompanied by a statement of the uncertainty.
- It is important as uncertainty must be reported on a calibration certificate.
- It is essential to determine whether a preset tolerance is met adequately.
- An adequate evaluation of measurement uncertainty acts as an indication of the quality of a measurement result.

2.5 General Approaches to Evaluation of Measurement Uncertainty

To evaluate the measurement uncertainty, various sources of uncertainty in the measurement under discussion need to be identified, and the significance of the uncertainty from each source needs to be evaluated. Finally a combined uncertainty is generated corresponding to the required level of confidence.

The most widely recognised approach for measurement uncertainty evaluation is the GUM framework. In recent years, the limitations of GUM framework have been widely discussed and resulted in the publication of the GUM SP1 using MCM method for measurement uncertainty.

2.5.1 GUM Uncertainty Framework

As mentioned in Chapter 1, in the area of measurement uncertainty, GUM has been regarded as the most fundamental document and guidance. The research and work surrounding this document has been substantial and led to the development of the recent research and work in this area as well as further supplementary and supporting documents to the original GUM, including the recent GUM SP1.

Since GUM's first publication in 1993, a large number of documents have been published discussing GUM's theoretical framework, its proposed methodologies and its applications in various areas and industries. Before the specific procedures and methodologies of GUM are discussed in this section, a brief literature review on GUM and its application is first given here.

2.5.1.1 Background of GUM Framework Development

GUM's first publication in 1993 formed a sound ground for an extensive research in the area of measurement uncertainty and led to the publication of a number of important documents, a selection of which are briefly reviewed as follows.

National Physical Laboratory (NPL) of the UK has been one of most important advocates and pioneers in the areas of metrology, measurement, associated standards and, of course, the uncertainty in measurement, along with other industries that it specialises in. NPL's history dates back to 1900, when it was firstly founded. During its history of over 100 years, NPL has developed into the largest applied physics organisation of the UK.

Serving as the UK's National Measurement Institute, a variety of documents in the measurement and measurement uncertainty areas have been published by NPL. In September 1994, shortly after the first publication of the original version of GUM, NPL published NIS 80 (NPL, 1994), the first edition of *Guide to the expression of uncertainties in testing*, the latest edition of which was published in March 2001. This guide is closely based upon the first edition of GUM and aims at simplifying the principles so they can be easily followed in specific fields. It emphasises the principles and provides guidance in estimating and reporting measurement uncertainty.

Apart from NPL, a number of other national and regional organisations published important documents in measurement uncertainty around the year 1994. For instance, the Nordic Innovation Centre published *Traceable Calibration and Uncertainty of Measurements and Tests* in 1994. Another important document is the Technical Note 1297 1994 Edition *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*, published by National Institute of Standards and Technology (NIST) of the United States in September 1994. The NIST Technical Note 1297 is actually based on the earlier version of the note, which was published in January 1993, before the publication of the original version of GUM. The 1994 edition of the note was published to address some questions which were raised for the 1993 edition and to recognise the publication of GUM, which served as the basis of the Notes. The aim of the TN 1297 is to provide the staff at NIST with a guidance regarding evaluation and expression of measurement uncertainty in order to improve and standardise the measurement results of the institute, providing a more succinct policy intended to be used by NIST staff.

In 1995, Eurachem, a pan-European organisation specialising in analytical chemistry and quality related issues in Europe, published its first edition of *Quantifying Uncertainty in Analytical Measurement* in 1995. A later edition of the document was published in 2000. On the basis of the GUM, this document extends its application to analytical chemistry to enhance the quality of chemical measurements. In the same year, the National Conference of Standards Laboratories (NCSL) of the US, published the 1995 edition of *Recommended Practice RP-12, Determining and Reporting Measurement Uncertainties*. The document discusses the GUM in further details and provided practical examples for a better understanding of its applications in practical cases.

In 1997, United Kingdom Accreditation Service (UKAS) published the second edition of *The Expression of Uncertainty and Confidence in Measurement,* the goal of which is to assist testing and calibration laboratories in evaluating and expressing measurement uncertainty in a more harmonised manner. This important document is based on both the GUM and other internationally recognised documents and standards, for example ISO/IEC 17025:2005. The GUM served as the basis of methodology for the document, which targeted for both beginners and more experienced users in the area. It is particularly worth noting that this document features an extensive number of worked examples for a better understanding of practical use of methodology. Moreover, Monte Carlo simulation was briefly explained in the document as a potential alternative for evaluation of measurement uncertainty by propagating the distributions.

NORDTEST/Nordic Innovation Centre published a number of documents and technical reports on measurement uncertainty over the years. These include *Traceable Calibration and Uncertainty of Measurements and Tests* published as early as 1994, *Calibration, Traceability and Uncertainty* in 1995 and *Tools for the Test Laboratory to*

Implement Measurement Uncertainty Budgets in 1999 etc. The series of documents provide valuable guidelines for researchers in the area, particularly for testing laboratories.

After the NPL published another important document, *A Beginner's Guide to Uncertainty of Measurement* (Bell,S. 1999) , the second edition of the document was published in 2001. This Guide aims at introducing the principles of measurement uncertainty to beginners in the area, including technicians, manufacturing managers and engineering students, with the important concepts and theories made easier to understand. The methodology of GUM is the basis for the procedures of measurement uncertainty calculation introduced in this Guide, accompanied by easy to follow examples. This Guide also prepares the readers for reading of more in-depth and advanced documents on measurement uncertainty.

Expression of the Uncertainty of Measurement in Calibration was published by European co-operation of Accreditation (EA) in 1999. Similar to *The Expression of Uncertainty and Confidence in Measurement* published by UKAS, this document aims at providing laboratories seeking accreditation with more detailed guidelines. The directions given in the document is extensive and comprehensive, accompanied with worked examples for different types of measurements as well as special cases.

Later in December 2003, EA published *EA guidelines on the expression of uncertainty in quantitative testing*, which features a brief summary of GUM and guidelines on the implementation of uncertainty evaluation, especially associated with quantitative testing fields. Also excellently summarised in this document are the benefits the uncertainty evaluation brings to testing laboratories. Similar to this document, Eurolab, European Federation of National Associations of Measurement, Testing and Analytical Laboratories, issued their technical report *Guide to the evaluation of measurement uncertainty for quantitative test results* in 2006. Apart from providing guidance on uncertainty evaluation for quantitative test results, this document also outlines alternative approaches, e.g. inter-laboratory comparisons.

As an extension to the report issued in 2006, Eurolab issued *Measurement uncertainty revisited: Alternative approaches to uncertainty evaluation* in 2007. This document summarises and compares various current approaches to uncertainty evaluation, which are supported by a large number of examples, from various testing fields and evaluation approaches, and their results.

Further to the documents intended to be used as guidelines by testing laboratories, International Laboratory Accreditation Cooperation (ILAC) published *Introducing the Concept of Uncertainty of Measurement in Testing in Association with the Application of the Standard ISO/IEC 17025* in 2002. This document is a concise guideline based on *ISO/IEC 17025:1999 General requirements for the competence of testing and calibration laboratories* and GUM, with a brief introduction to the concept of measurement uncertainty and its implementation.

From the above reviewed documents pertaining to the master document and guidance in the evaluation of measurement uncertainty, the GUM, it is clearly seen that between the initial publication of GUM in 1993 and early 2000s, a large number of documents in this area discuss the concept and general implementation of uncertainty evaluation. Part of them aim at providing various groups of readers including beginners with an introduction of basic concept and implementations in a simplified manner, whereas the other part of them focus on a more detailed and in-depth exploration into specific fields and practical situations.

Starting from early 2000s, extended discussions based on the GUM started to appear on published papers. These discussions include the limitations of GUM, which are summarised later in this section, after its early practical applications. Following these discussions, researchers began to look into potential alternatives which are able to overcome some of GUM's limitations and extend its functionality and flexibility. This has resulted in the publication of GUM SP1 in 2008, proposing the use of Monte Carlo Simulation for propagating the distributions in uncertainty measurement.

2.5.1.2 Understanding of GUM Uncertainty Framework Procedure

Based on the GUM framework, the uncertainty components are generally grouped into two categories, according to their evaluation method, Type A and Type B. It is important that Type A and Type B are not mistaken for the concept of "random" and "systematic".

Type A:

"A component of measurement uncertainty by a statistical analysis of measured quantity values obtained under defined measurement conditions"

Type B:

"A component of measurement uncertainty determined by means other than a Type A evaluation of measurement uncertainty"

(JCGM, 2008)

A general GUM framework flowchart for evaluating measurement uncertainty is listed below:

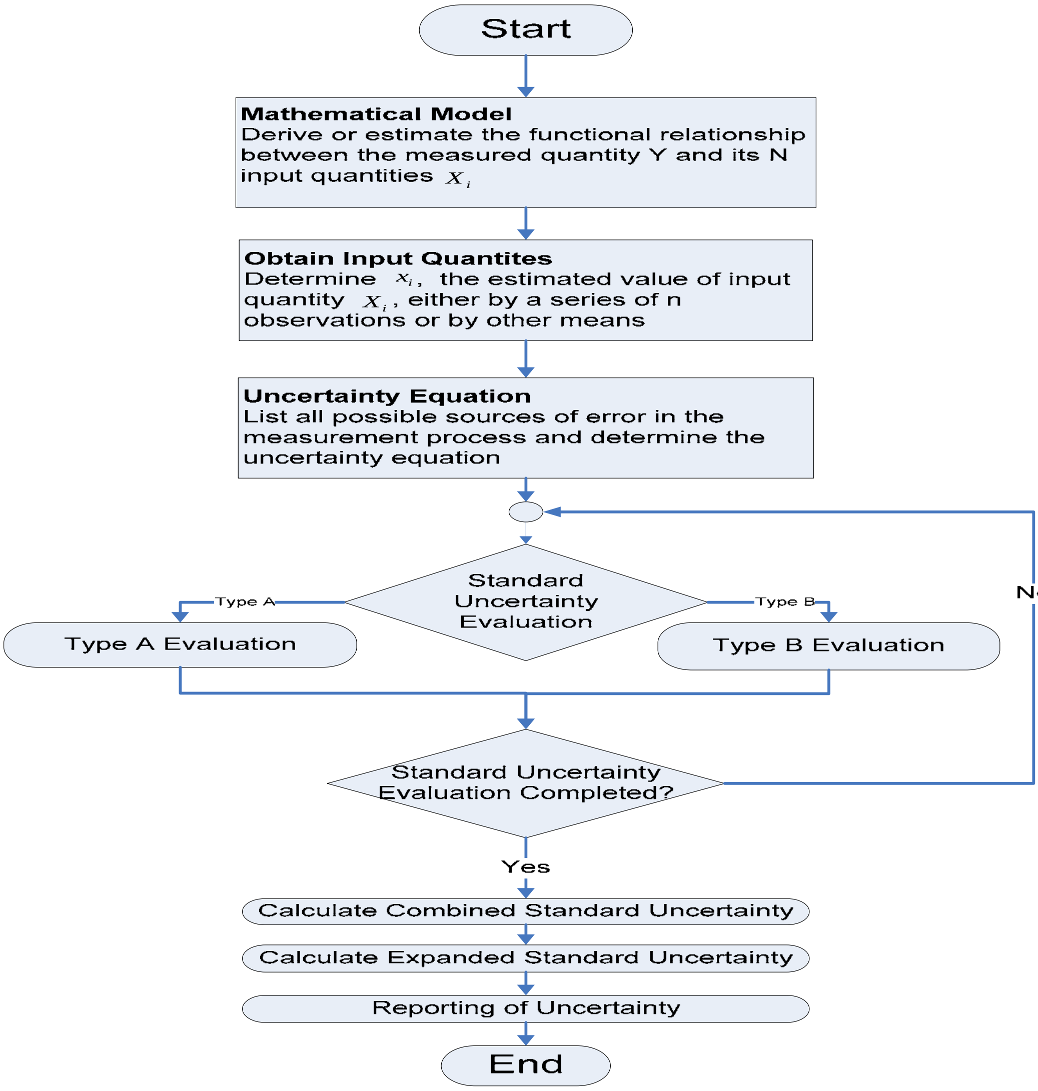


Fig.2.1 GUM Framework

1). Modelling the Measurement

At the outset of a measurement, a model with a functional relationship needs to be identified. The model needs to specify the input quantities and how they are combined to obtain the value of the measurand. Establishing the correct model is critical to the evaluation of measurement uncertainty. However, to a large number of users, it is in some cases very difficult to identify the model.

In a measurement, the measurand Y is usually determined from N input quantities via a function (mathematical model of measurement):

$$Y = f\left(X_1, X_2, \cdots X_n\right)$$

Here the input quantities can themselves be measurands and determined by the functions of other quantities, which will make the function more complicated. An estimate of the output, y, is obtained through the estimates for the input quantities:

$$y = f\left(x_1, x_2, \cdots, x_n\right)$$

The uncertainty related to the output is achieved by combining the uncertainty of each input quantity. Each standard uncertainty is evaluated through Type A or Type B evaluation.

2). Type A Method of Evaluation

As mentioned above, Type A evaluation is performed on the basis of statistical method. Typically, the best estimate of the output quantity is the arithmetical mean from a series of observations:

$$\overline{q} = \frac{1}{n} \sum_{k=1}^{n} q_k$$

To estimate the distribution of q, experimental standard deviation is used:

$$s\left(q_k\right) = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} \left(q_k - \overline{q}\right)^2}$$

In probability theory, standard deviation pertains to a measure of the dispersion of a population or a probability distribution. The standard uncertainty for Type A evaluation has the same value as the standard deviation.

Since the mean of the measurement results is normally used as the best estimate of the measurand, the standard deviation of the arithmetical mean should be used, which is determined by dividing the standard deviation found in a prior experiment by the square root of the number of measurements.

$$u(x_i)=\frac{s_p}{\sqrt{n}}$$

Type A method evaluation flowchart is listed below:

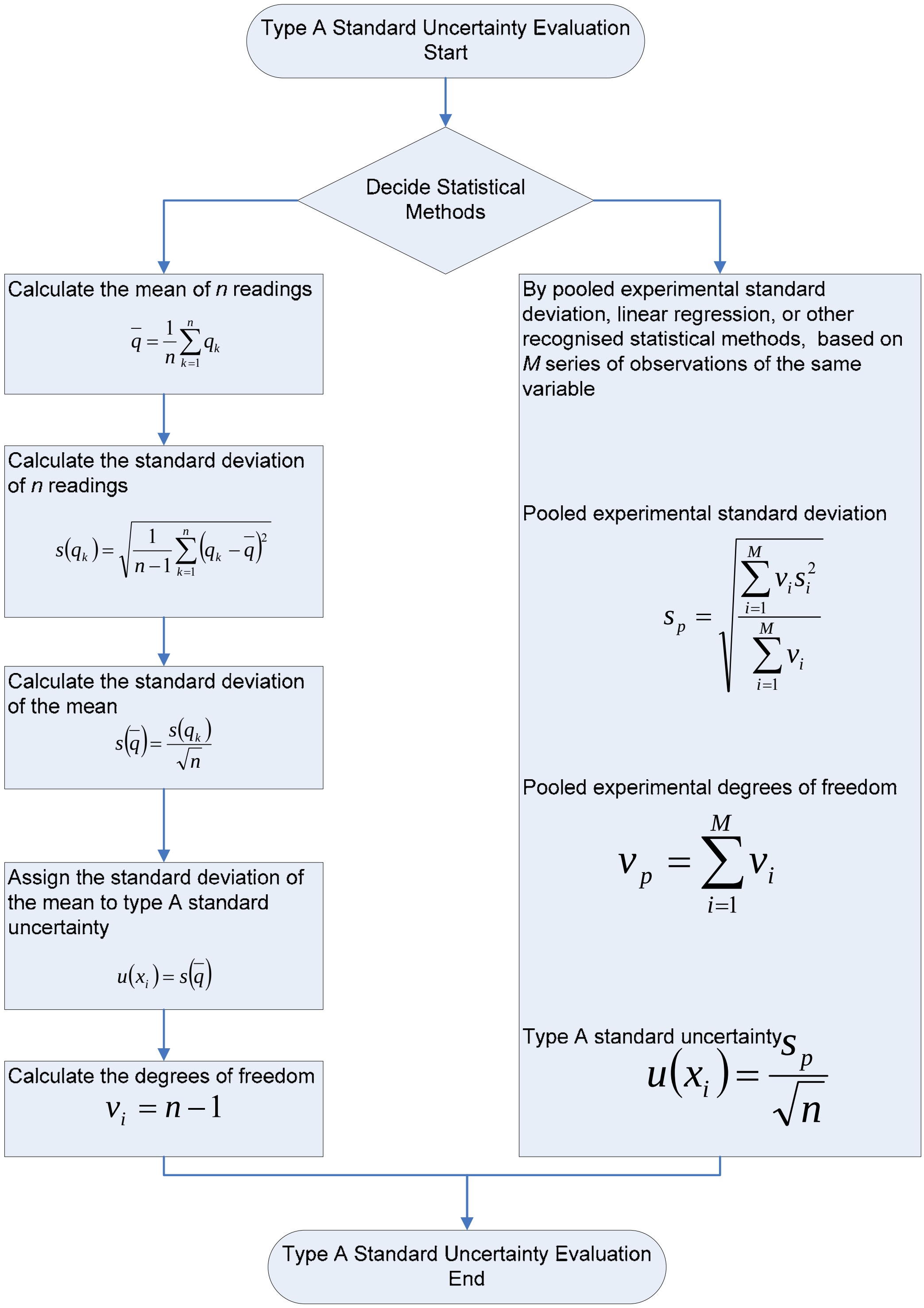


Fig. 2.2 Type A Evaluation

3). Type B Method of Evaluation

Any uncertainty evaluation by means other than the statistical analysis of a series of observations is classified as Type B evaluation. The sources of information usually include the following:

- data from calibration certificates
- manufacturer's specifications
- uncertainties assigned to reference data taken from handbooks
- data from previous measurements
- experience with, or general knowledge of, the behaviour and properties of relevant materials and instruments [GUM]

In Type B evaluation, the quoted uncertainty needs to be converted to a standard uncertainty.

In many cases, the quoted uncertainty, for example the quoted uncertainty stated on a calibration report, is a stated multiple of the estimated standard deviation. In these cases, the standard uncertainty is obtained by dividing the quoted uncertainty by the multiplier (GUM).

By considering the probability distribution, the quoted uncertainty can be converted into standard uncertainty by dividing the quoted uncertainty by a factor which is determined by the probability distribution.

(1) Rectangular Probability Distribution

This refers to cases where the measurement results are equally likely to fall anywhere between the highest and lowest values within the interval. The standard uncertainty, in this case, is determined by dividing the half-interval by $\sqrt{3}$.

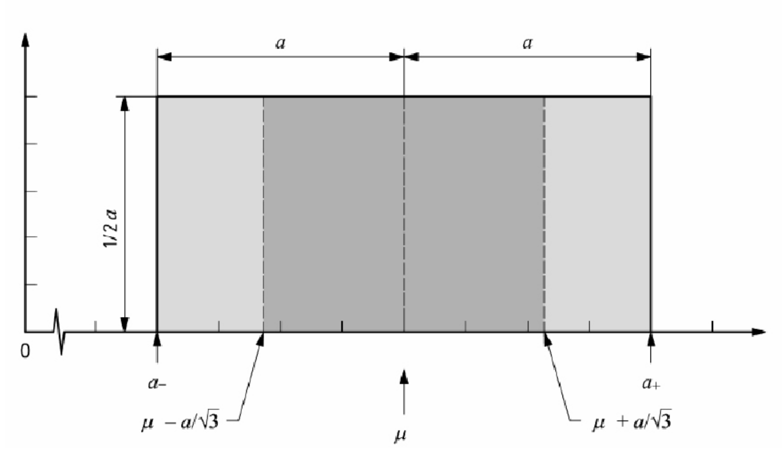


Fig. 2.3 Rectangular Probability Distribution

(2) Triangular Probability Distribution

It is more suitable to use the triangular distribution in cases where most of the measurement values are likely to fall near the centre of the distribution. The standard uncertainty, in this case, is determined by dividing the half-interval by $\sqrt{6}$.

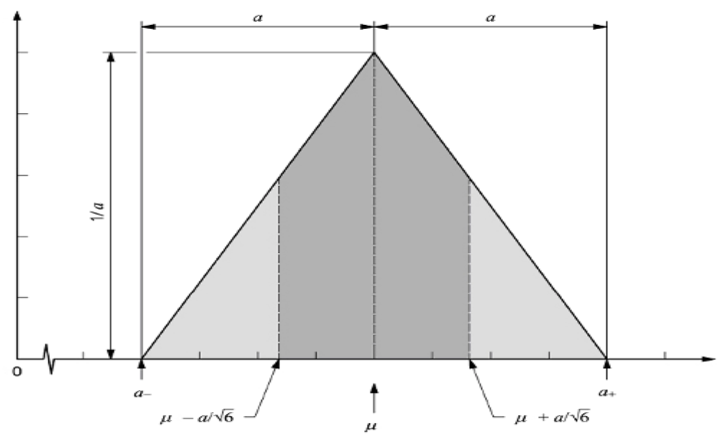


Fig. 2.4 Triangular Probability Distribution

(3) Normal Probability Distribution

Normal, also known as Gaussian probability distribution, usually refers to cases where the series of measurement results are likely to fall near the mean value, rather than away from it. It often involves a level of confidence of 95% or 99% (GUM). The standard uncertainty, in this case, is determined by dividing the quoted uncertainty by the appropriate factor, e.g. 1.96 and 3 for a level of confidence of 95% and 99%, respectively.

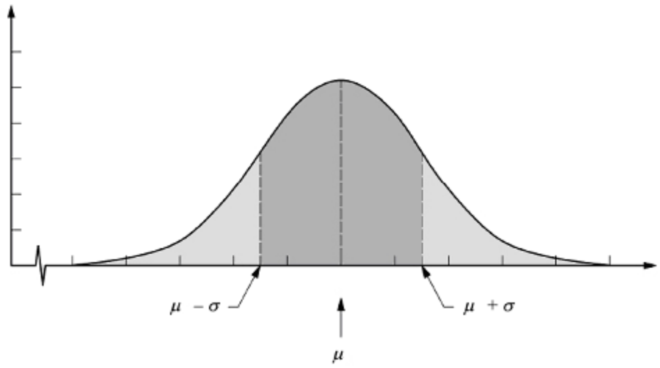


Fig. 2.5 Normal Probability Distribution

(4) U-Shape Probability Distribution

Typically, this is usually used in cases of mismatch uncertainty, often in radio and microwave power frequency measurement (GUM). In these cases, the standard uncertainty is divided by $\sqrt{2}$:

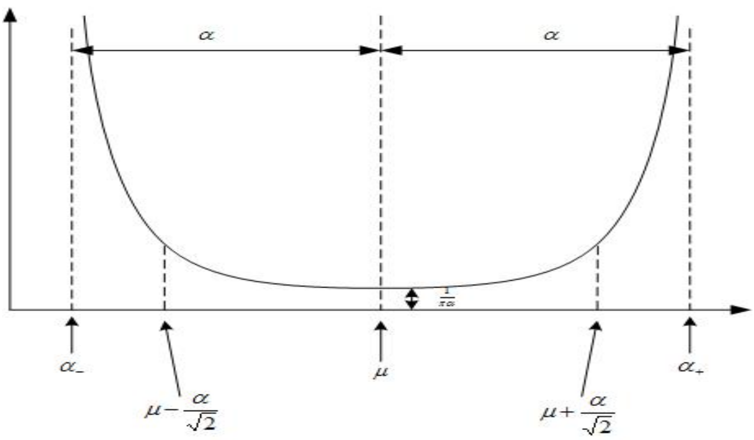


Fig. 2.6 U-Shape Probability Distribution

Type B evaluation is usually based on an assumption that the distribution is known and the degree of freedom is infinite.

In general, degree of freedom is the number of values in the final calculation of a statistic that are free to vary. (Walker, 1940)

Type B method evaluation flowchart is listed below:

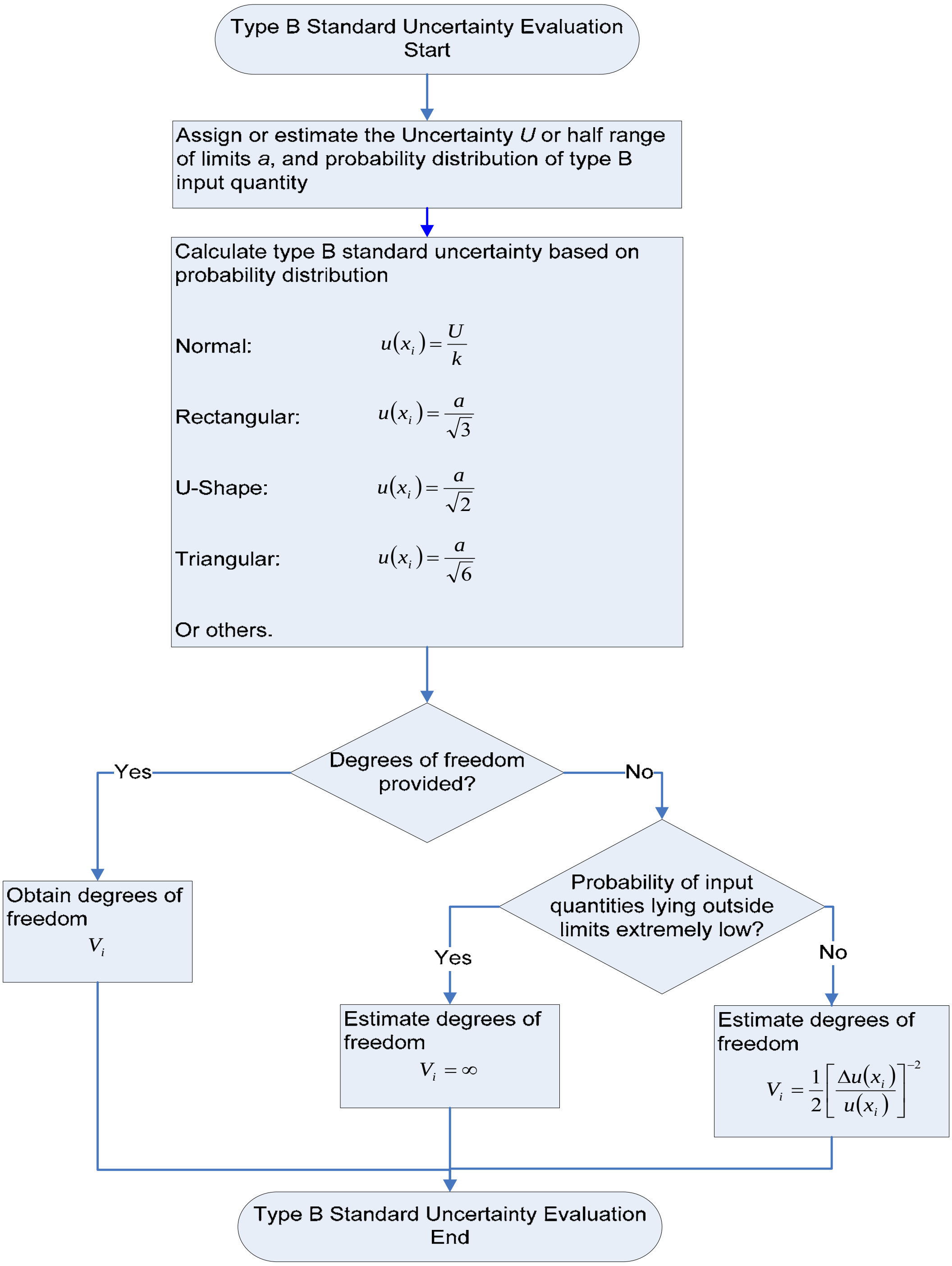


Fig. 2.7 Type B Evaluation

4). Combined Standard Uncertainty

In order to obtain an overall uncertainty for the output estimate, the uncertainty of the input components need to be combined. There are two ways of how the combined standard uncertainty can be calculated.

- Uncorrelated Input Quantities

This refers to the cases where all input quantities are statistically independent from each other. The combined standard uncertainty is calculated as the positive square root of the combined variance:

$$u_c(y) = \sqrt{\sum_{i=1}^{N}[c_i u(x_i)]^2} = \sqrt{\sum_{i=1}^{N}\left[\frac{\partial f}{\partial x_i}u(x_i)\right]^2} = \sqrt{\sum_{i=1}^{N}[u_i(y)]^2}$$

In this equation, $C_i \equiv \frac{\partial f}{\partial x_i}$, C_i are sensitivity coefficients, which reflect how components are related to the result, i.e. how the value of y varies with changes in the parameters x_1, x_2 etc.

- Correlated Input Quantities

In cases where the input quantities are interdependent to each other, the combined variance is calculated as:

$$u_c(y) = \sqrt{\sum_{i=1}^{N}[c_i u(x_i)]^2 + 2\sum_{i=1}^{N-1}\sum_{j=j+1}^{N} c_i c_j u(x_i) u(x_j) r(x_i, x_j)}$$

This equation involves a correlation coefficient which indicates the strength and direction of a linear relationship between two random variables:

$$r(x_i, x_j) = \frac{s(\overline{X_i}, \overline{X_j})}{s(\overline{X_i})s(\overline{X_j})}$$

Combined uncertainty calculation flowchart is given below:

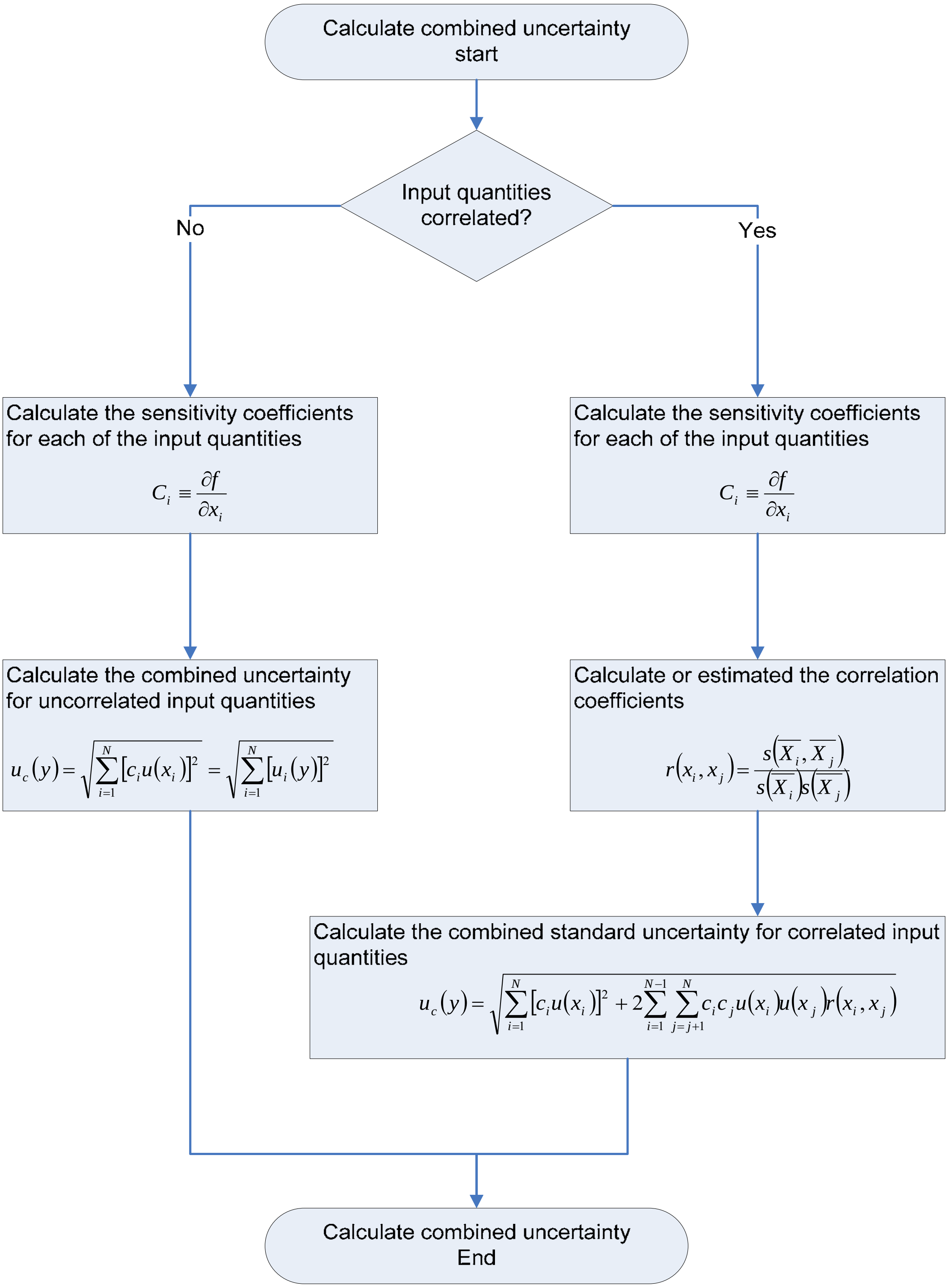


Fig. 2.8 Combined Uncertainty Calculation

5). Expanded Standard Uncertainty

In most cases, an expanded uncertainty is required to be quoted when reporting the result of the measurement/test with its uncertainty. The general idea of measurement uncertainty is about defining an interval within which the true value of the measurement is likely to fall. Expanded uncertainty is related quantitatively to the interval within which the true value is confidently affirmed to fall, based on a calculation involving a coverage factor and related confidence level.

The expanded uncertainty is calculated as multiplying $u_c(y)$ by a coverage factor:

$$U = ku_c(y)$$

The coverage factor is determined on the basis of the level of confidence. Level of confidence pertains to how confident the user is after a series of measurements that the true value of the measurement result will fall within the confidence interval. The method of how k is calculated is shown in Appendix I.

Expanded uncertainty calculation flowchart is shown below:

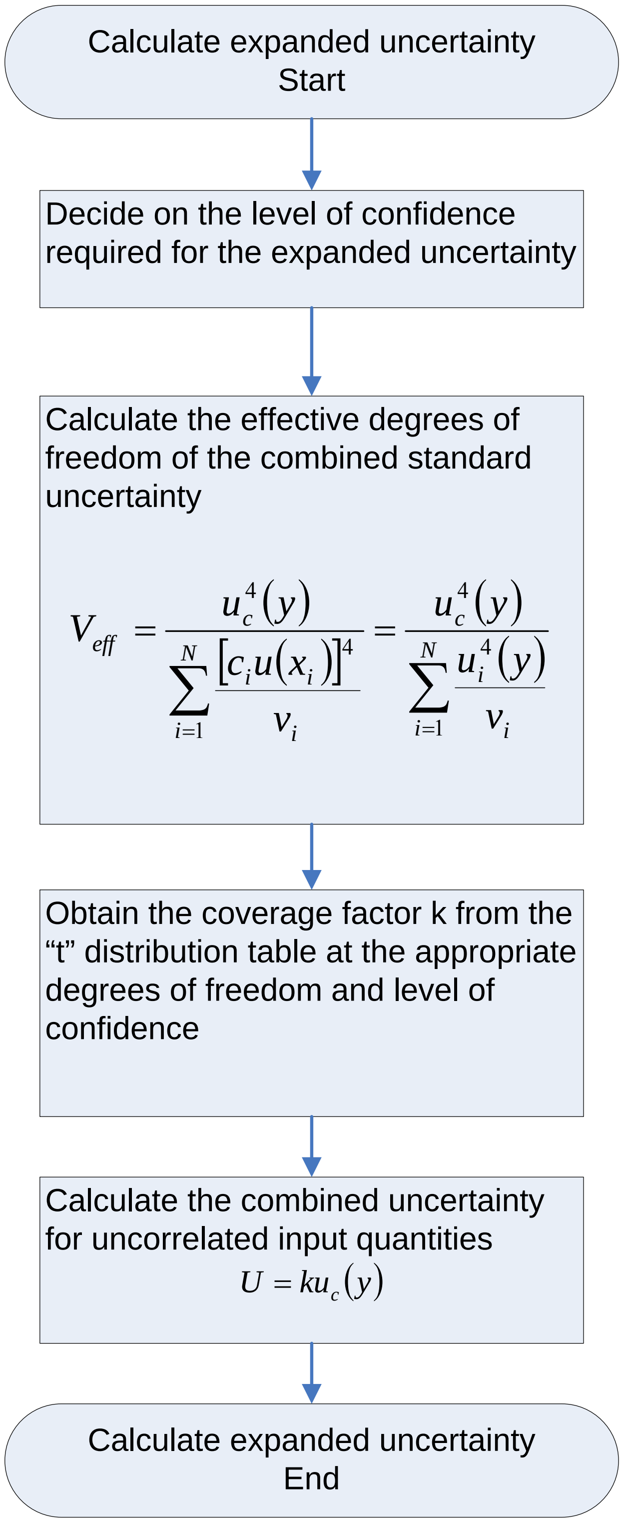


Fig. 2.9 Expanded Uncertainty Calculation

6). Result and reporting

When reporting the result, it is essential that complete information is included:

- The measurement result

- The expanded uncertainty
- The coverage factor and level of confidence
- How the uncertainty is estimated

2.5.1.3 Summary

Since its initial publication in 1993, GUM has been recognised as the master guideline on the evaluation of measurement uncertainty. GUM is very successful in the sense that it has established general and standardised rules and procedures for evaluating and expressing uncertainty and it is easy to implement. GUM aims at being applicable to a wide range of measurements including calibration and testing. The methodology proposed by GUM, the GUM uncertainty framework, has been widely adopted and used globally for a variety of guides, standards as well as in the development of software tools.

Although the GUM has enjoyed wide applications, there are a few conditions that need to be met for the GUM framework to be validly applicable. For example: the conditions for the central limit theorem must be met; the Welch-Satterthwaite approximation (W-S formula) for the effective degrees of freedom must be adequate; the measurement model must have sufficient linearization (Bich, 2006). In many cases, these conditions are all met and the GUM is validly applicable. However, it has been documented that in some cases the GUM framework is used in violation of these conditions, making the results merely an approximation (Bich, 2006).

In the summarising documents compiled by Bich between 2006 and 2008, he pointed out that the specific conditions that need to be met for the GUM framework to be validly applicable have exposed the GUM framework to a number of limitations. He summarises that the limitations of the GUM framework mainly pertain to the determination of a coverage interval. Apart from the fact that the central limit theorem needs to be met, the GUM framework has a problem that the "assignment of degrees of freedom to Type B evaluations looks 'artificial'." Its limitations are also reflected in the evaluation of the standard uncertainty, e.g. in real world, the input quantities may be asymmetric and the measurement model may be nonlinear as mentioned above.

Generally, the limitations associated with the GUM framework make it less appropriate in certain cases and less adaptive to more complicated measurement models. These limitations have been widely discussed in recent years and potential approaches to overcome these limitations are proposed and discussed. The Monte Carlo Method has emerged as the most popular solution in these discussions, which resulted in the publication of GUM SP1 in 2008. The MCM and GUM SP1 are discussed in more details in the following section, accompanied with literature review.

2.5.2 GUM SP1 MCM Uncertainty Framework

Following the discussions on the potential approaches to overcome the limitations associated with the GUM uncertainty framework, Monte Carlo Method surfaced and was eventually adopted in GUM SP1. Basically, the MCM is a purely numerical approach involving numerical simulation. This section discusses the basic concept of MCM and its integration in GUM SP1.

2.5.2.1 Introduction to Monte Carlo Method

The history of the term "Monte Carlo Method" goes back to the 1940s when it was coined by physicists working on nuclear weapon projects. However, its early development is severely limited by the development of computers at that time. It is only after the computer started to be used more often that MCM begins to obtain popularity in the fields of nuclear research, physics, and operations research. Having developed for more than 50 years, the MCM is nowadays often used in physics, mathematics, designing, finance, and business (e.g. analysis of business risks) etc.

Monte Carlo methods are actually a group of algorithms involving repeating of random sampling to calculate results, opposite to deterministic modelling. As the amount of calculation is substantial, it is always required to be carried out by computers, which also explains the importance of computer to MCM. MCM's strength is particularly obvious when dealing with cases where there is high uncertainty in inputs.

It is particularly worth noting that MCM does not always require truly random numbers. Instead, in many cases pseudo-random numbers are used for ease of test and re-run of the simulations. Typically in an application of MCM, the pseudo-random numbers need to pass a series of statistical tests to ensure that they are random in relation to one another.

The basic principles of MCM are widely recognised by a large number of published books and articles. Although MCM is a group of approaches, its basics can be briefly summarised as (Robert and Casella 2004; Rubinstein and Kroese 2008; Frenkel 2004): First of all, a domain of possible inputs is defined; Secondly, inputs are generated randomly; Then the inputs are computed and the last stage usually involves the aggregation of the results from the separate computations to generate the final result.

2.5.2.2 Background of GUM SP1 MCM Framework

The advantages of MCM posed its potential applications in the evaluation of measurement uncertainty. As early as in late 1990s, the user of MCM in this area was briefly mentioned by some organisations in their documents, e.g. UKAS's *The Expression of Uncertainty and Confidence in Measurement*, which is reviewed earlier in the chapter. Weise and Zhang (1997) also outlined in their paper that the approach of MC simulation has significant potential in solving problems involving a huge number of input quantities.

The year 2001 saw the publication of a large number of important documents on measurement uncertainty. Particularly worth noting is that the Monte Carlo method was becoming an increasingly popular topic in the area and started to be regarded as a potential supplementary element to the mainstream GUM.

NPL published *Software specifications for uncertainty calculation and associated statistical analysis* (Cox et al, 2001) in 2001. This document aims at providing specifications of software units for the evaluation of measurement uncertainty. It is compiled on the basis of existing publications including GUM and particularly helps readers who need to use software to evaluate measurement uncertainty. This document serves as the initial version of the later publications, which include the 2006 publication of *Software specifications for uncertainty evaluation* (NPL Report DEM-ES-010), which was later further revised in 2008. In regard to this series of documents, it is particularly worth mentioning that they take into account the discussions related to the use of a Monte Carlo method in measurement uncertainty. In short, these documents outline specifications of software unit in terms of GUM uncertainty framework, a Monte Carlo method and the validation of GUM framework using a Monte Carlo method.

Uncertainty and statistical modeling was published in the same year, which on the one hand provides a guideline on evaluation of measurement uncertainty, and on the other hand focuses on the discussion of statistical modeling. It discusses certain limitations associated with the approach proposed by GUM, for which an alternative numerical approach, MC simulation, may be used in cases where GUM may not apply. This document is especially useful in the sense that it provides a number of examples for a better understanding of practical situations.

The important role of MC simulation in uncertainty evaluation was further emphasised by Cox et al in *Use of Monte Carlo Simulation for Uncertainty Evaluation in Metrology* (2001) published in *Advanced mathematical & computational tools in metrology V.* In this article, the limitations of using GUM for uncertainty evaluation is discussed and MCS is proposed as an alternative. It is claimed in this article that the basic concept of MCS is simple and as a numerical method it offers flexibility especially for cases where GUM may not suit. It is also pointed out in this article, as well as in some other papers in the same period, that MC simulation can also be utilised to validate the uncertainty obtained using GUM.

In the book "Evaluating the measurement uncertainty: fundamentals and practical guidance" published in 2002, Lira has done an excellent job in providing extensive knowledge ranging from SI units and basics of metrology to measurement errors and uncertainty. The book then looks into the steps of uncertainty evaluation which are based on the GUM framework. The point that particularly worth mentioning about this book is its exploration into more advanced topics including MCM and Bayesian inference. The wide range of information and various examples included in this book makes it an excellent guideline as well as a good reference book.

MCM's integration into SP1

In 1997, the Joint Committee for Guides in Metrology (JCGM) was established by the seven organisations participating in the compilation of GUM. The Committee was joined by ILAC in 1998 (Bich, 2007). The working group 1 of JCGM have since worked on supplementary documents to GUM. In 2008, the GUM SP1 incorporating the MCM was published under the name of the eight organisations.

GUM SP1 mainly deals with the construction of a coverage interval with a stated probability for the measurand. As mentioned earlier in this chapter, the aim of SP1 is to overcome some of the limitations of the GUM framework, i.e. to provide guidance on issues that are not explicitly treated in GUM. The principle and fundamental approach of SP1 are outlined below with a brief introduction to the procedures of uncertainty evaluation using MCM.

Basically, the MCM framework involves propagation of probability distributions using a mathematical measurement model. This approach overcomes some of the limitations of GUM uncertainty framework. Specifically, its value lies in cases where the linearlisation of the measurement model is not adequate and where the probability density function for the output quantity is not a Gaussian distribution or a scaled and shifted *t*-distribution. Although the SP1 has a wider application, the supplement points out clearly that the GUM uncertainty framework remains the primary approach for uncertainty evaluation where it is applicable and the SP1 should be used jointly with GUM.

2.5.2.3 Understanding of GUM SP1 MCM Framework Procedure

SP1 outlines a detailed procedure of the MCM framework for uncertainty evaluation using propagation of distributions. The documents by a number of other organisations such as NPL Report DEM-ES-010 also reiterate and summarise the procedure outlined by SP1. Although the wording of the various documents differs from each other slightly, the basic underlying principle of the procedure remains consistent among the documents.

The heart of the MCM approach is the repeated sampling from the PDFs for the input quantities. It is also worth noting that the SP1 is intrinsically Bayesian. The discussion about the Bayesian analysis in uncertainty evaluation started as early as in the 1990s (Weise, 1992). In SP1, the Bayes' theorem can be used to assign and calculate the PDFs to the input quantities (Elster et al, 2007). The consistent Bayesian approach in SP1 is tightly based on the MCM (Cox and Siebert, 2006). According to Lira and Wöger, the use of Bayesian approach in uncertainty evaluation brings improvement to the evaluation especially in terms of the coverage probability. The benefit of the Bayesian approach is further discussed by Kacker (2006), who claims it simplifies the process and is a useful alternative to the W-S formula used in GUM framework.

This section quotes a summary of the procedure of the MCM approach directly from SP1, which is succinct and easy to understand:

a) select the number M of Monte Carlo trials to be made.

b) generate M vectors, by sampling from the assigned PDFs, as realizations of the (set of N) input quantities *Xi*.

c) for each such vector, form the corresponding model value of *Y* , yielding *M* model values.

d) sort these *M* model values into strictly increasing order, using the sorted model values to provide *G*.

e) use *G* to form an estimate *y* of *Y* and the standard uncertainty *u*(*y*) associated with *y*.

f) use *G* to form an appropriate coverage interval for *Y* , for a stipulated coverage probability *p*.

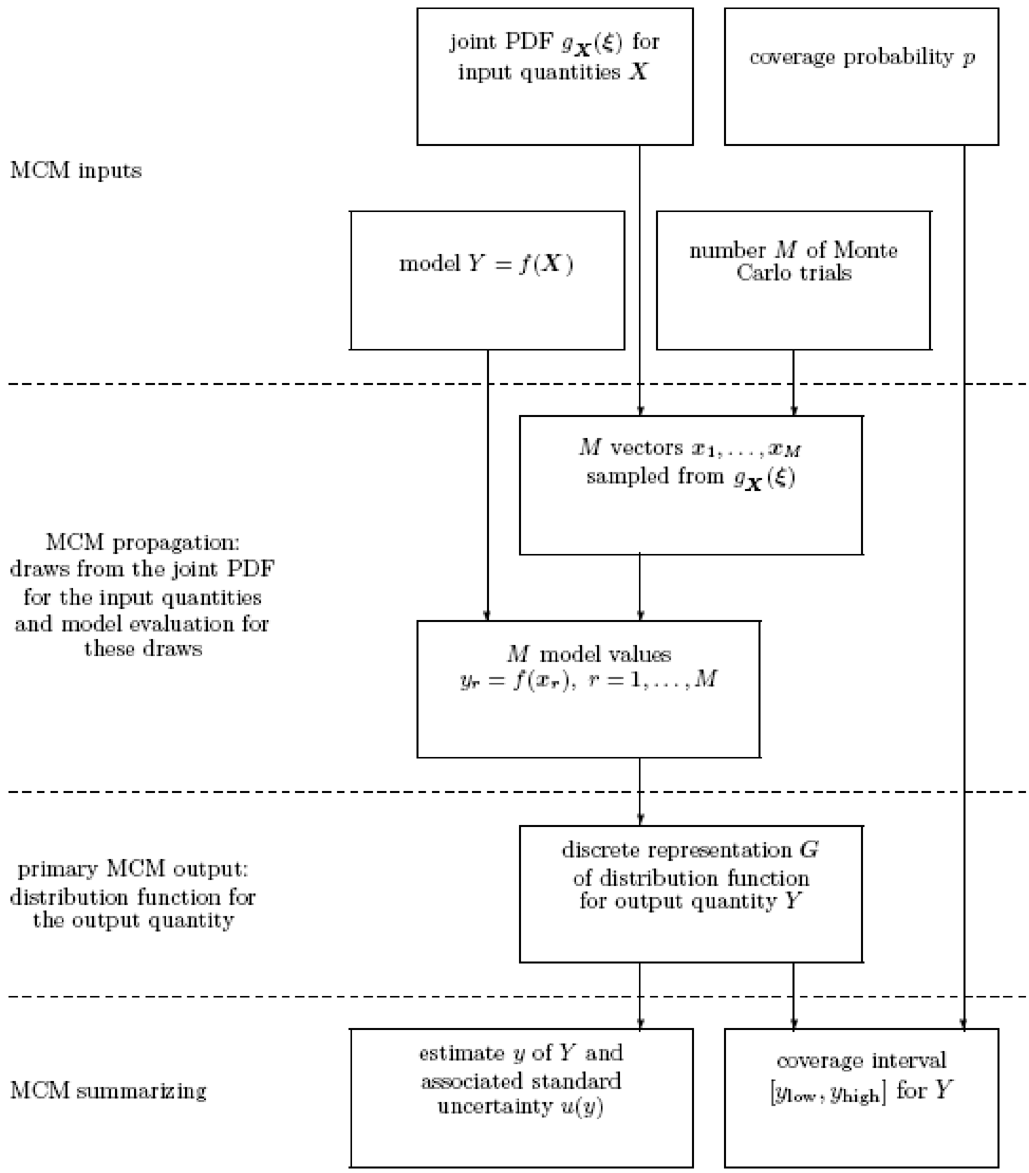


Fig. 2.10 The Propagation and Summarizing Stages of Uncertainty Evaluation Using MCM to Implement the Propagation of Distributions (GUM SP1)

To sum up, the MCM framework is applicable to a wider range of problems, compared with GUM framework, making it more general in this sense. However, it is important to take note that SP1 is in accordance with the basics of GUM and should be used jointly with it. SP1 puts more emphasis on the expanded uncertainty (a coverage interval) than on the standard uncertainty, i.e. the main output of the MCM approach is a coverage interval with a stipulated coverage probability. For SP1, Type A and Type B do not apply to PDFs and there is no need for determining the degrees of freedom. The MCM approach can also be used as a means to validate the results provided by the GUM framework.

2.5.3 Currently Available Software Tools for Uncertainty Evaluation

As mentioned in Chapter 1, a number of software tools for uncertainty evaluation are currently commercially or academically available. Chapter 1 also briefly outlines some of the limitations associated with these current tools, which was covered in more details in the author's previous work (Wei et al, 2007, 2009).

Jurado and Alcazar provided in their paper in 2005 a review and comparison of some the current software tools. Castrup's article in 2004 also offers an excellent and detailed review and comparison of some of the tools, covering both commercially available tools as well as freeware and providing review on their features, functionality, user-friendliness, technical support etc.

With the MCM's application starting to be discussed for uncertainty evaluation, software tools integrating MCM approach have been developed and made available in recent years. In particular, NPL Report MS1 of 2008 proposes a software application (NPLUnc) involving MCM. The software introduced in this report support both the GUM framework and the SP1 MCM framework and allows users to use these approaches to the four example problems presented in GUM SP1.

Matlab is used in NPLUnc to facilitate the MCM approach. Matlab is a powerful interaction computing environment and, in the mean time, a high level programming language. Matlab, originated from the words “matrix laboratory”, aims at an easy handling of matrix and complex arithmetic, although it is worth noting that Matlab is based on pure numeric computation. Since its initial appearance in the 1970s, this strong software package has seen a significant development of itself, gaining popularity in both the industrial and academic fields.

Although there are other software tools available, use of Matlab in measurement uncertainty evaluation is proposed in this thesis due to its advantages, especially its capability of implementing MCM (Fernández, 2009). Specifically, as MCM relies on repeated random sampling, the generation of random numbers is vital. In Monte Carlo simulation, it does not always require the random numbers to be truly random. In many cases, pseudo-random numbers are used and they are generated by algorithms instead of truly random processes. Matlab can easily generate pseudo-random numbers.

Besides, Matlab provides the users with an easy, friendly and interactive environment. It is also easily available at an inexpensive cost.

As mentioned earlier, Matlab forms part of the basis of NPLUnc thanks to its strengths. The web-based KB-DSS proposed in this thesis takes into account the review on the current software tools done by other researchers. The proposed system also takes advantage of Matlab’s strengths and bases its MCM approach on NPLUnc, with modifications and enhancement introduced.

2.6 Summary

In this chapter, the concept of measurement uncertainty and its importance are firstly outlined. It is particularly important to understand the difference between measurement errors and measurement uncertainties. Two most widely applied approaches for uncertainty evaluation, GUM uncertainty framework and GUM SP1 MCM framework, are discussed in the chapter with an overview of their concepts, history and an outline of their procedures.

It is outlined in this chapter that GUM has certain limitations, which resulted in the publication of SP1 to overcome some of these limitations. A brief comparison of these two approaches is made in this chapter which indicates that the SP1 is more general in terms of applicability. However, it is also emphasised that these two guiding documents should always be used in conjunction with each other.

In this chapter, a number of currently available software tools for uncertainty evaluation are briefly reviewed. The software made by NPL is of particular interest to this thesis, as it features the use of Matlab, which forms part of the basis of the web-based KB-DSS proposed in the thesis.

A section of this chapter is occupied to introduce SI, preparing for the more detailed discussion of the proposed system in later chapters.

Chapter 3 Expert System and Decision Support System

3.1 Introduction

Generally, information systems are the software and hardware systems that carry out applications involving data, for example, the collecting, storage, processing and communication of data and information.

In history, a few inventions, including the invention of movable type printing, a printing and typography system firstly invented in China around 1040, and the portable typewriter in the 19^{th} century, have boosted the early development of information system. The world's first large-scale mechanized information system was recognised to be Herman Hollerith's 1890 tabulating machine, which was used to tabulate the 1890 census of United States. This was a critical step in the development of automated information system. More recent developments in human history include the introduction of the first electronic computers around 1940–1945 and the appearance of personal computers in early 1970s, which made information system available to small businesses and individuals. In early 1990s, the World Wide Web was introduced and served as a major booster of information system for individual use. In today's daily life, the information system has become an indispensable part of an organisation's and an individual's life, examples of which include private email communication, commercial teleconferencing and online purchasing systems.

Information systems consist of a number of separate components, such as computer hardware and software, telecommunication systems, databases, human resources and procedures (Encyclopædia Britannica, 2009). Information technologies are critical part of information systems and there are a few types of information systems, which include transaction processing systems, expert systems, decision support systems, knowledge management systems, database management systems, and office information systems. Among these types of information systems, expert systems and decision support systems have particular advantages and they can be utilised to facilitate the evaluation of measurement uncertainties.

This chapter discusses the main features and characteristics of these two types of information systems and investigates into how their advantages and the integration of these two types of systems can be realised to optimise the software tools which help end users evaluate measurement uncertainty.

3.2 Expert System

3.2.1 History of Expert System

In people's daily life, they turn to experts if they are faced with a problem in a specific area which requires expertise to solve. Expertise is knowledge learned over a period of time which helps people solve problems. Developing a machine that can "think" like an expert has been the research topic for decades. Thanks to the effort of a large number of scientists and researchers, expert system has been introduced in a large number of areas and its application has been expanding ever since its introduction.

Expert system (ES) is software which solves problems in a way as a human expert would do by simulating or reproducing the performance of real human expert. It is normally related to a specific problem area or domain. It is tightly linked to artificial intelligence and has its root planted in cognitive science. Basically, an ES is a decision making and problem solving system, where the user provides input information by answering pre-designed questions and/or entering data. The ES will keep asking questions until it can reach conclusion(s), sometimes with a statement of confidence level.

There are different types of ES, rule-based ES, network-based ES and frame-based ES. In 1960s, Edward Feigenbaum, Joshua Lederberg and Bruce Buchanan started working on Dendral, the world's first ES at Stanford University (Lindsay, 1993). The aim of Dendral was to create an "artificial expert" which could determine the molecular structure of previously unknown chemical compounds (Shang, 2005). Dendral used inference rules and showed the significance of the domain-specific knowledge. In 1970s, Edward H. Shortliffe developed MYCIN, an ES involving diagnosing and treating certain infectious diseases (Shang, 2005). Using MYCIN, the user can input the information about a patient's symptoms into the system, which analyses the information and if the symptoms are related to any disease the system has knowledge for, the system will ask for more detailed input with the help of which the system can narrow down the possibilities until it reaches a diagnosis.

Nowadays, ES are commercially and academically available in a large number of industries and areas, for example:

- ONCOCIN was developed in late 1970s aiming to use artificial intelligence to assist physicians on medicines and their dosages and testing.
- Molgen which is used to assist biologists in planning DNA experiments.
- PUFF developed at Stanford in 1970s aims at diagnosing obstructive airway diseases.
- PROSPECTOR which became famous in 1980s was designed to assist geologists in the exploration of sites containing valuable ores.
- A quality function deployment (QFD) based ES was developed by Chakraborty and Dey for non-traditional machining processes selection (Chakraborty and Dey in 2006).
- Iqbal et al (2006) developed an ES for optimising parameters and predicting performance measures in hard-milling process using fuzzy logic.

A comprehensive review on ES and its applications was conducted by Liao featuring a review on a large number of different types of ES documented in publications from 1995 to 2004 (Liao, 2005).

3.2.2 ES Architecture

A typical ES consists of several basic parts: knowledge base, working memory, inference engine, user interface and knowledge acquisition system.

Knowledge base:

Knowledge base contains the knowledge obtained from human experts in a computer-readable form. It is basically a collection of rules and information derived from human experts. As knowledge base is one of most important parts of an ES, it is required that the knowledge base is usable, correct and complete.

The method of storing knowledge in a knowledge base is called knowledge representation. The common methods of representing knowledge include semantic networks, production rules, frames and uncertainty and fuzzy logic.

Working memory:

Working memory is the collection of data and information generated during the session of an ES, including the answers from end user, known facts, intermediate information generated by the inference engine etc.

Inference engine:

Inference engine carries out the work related to selection of appropriate knowledge and application of the selected knowledge. Inference engine tries to reach a decision by running inference rules. Inference rule is an *if*/*then* (antecedent/consequent) statement, which includes an *if* clause and a *then* clause. When using inference rules, two main reasoning methods are applied, forward chaining and backward chaining.

Forward chaining, also called data driven method, starts by applying already known facts and information to the inference rules. This process is repeated as additional facts are required as the process proceeds. This continues until a particular goal is reached. In this approach, as the start point is the known facts, the inference rules used are determined by the known facts.

In contrast, backward chaining, also called goal driven methods, starts with the goal(s) and by searching the inference rules looks for the if clause which is known to be true.

Either of these two methods has its own advantages and should be used with consideration of the characteristics of each particular case.

User interface:

The user interface is in its essence a "bridge" between the end user and the ES. It is an interacting point which shows the questions to the user and passes the user's response to the ES for the inference engine to run the inference rules. The user interface also functions as a "check point" where the answer of the user is checked and made sure it complies with the reasoning rules of the ES. Practically, the user interface can be dialog boxes, table or command prompts, depending on the design of different ESs.

Knowledge acquisition system:

The knowledge acquisition system is the part of the ES which obtains the initial facts and information from the human experts. Its aim is to change the obtained knowledge from its original form to a form that is readable to the machine and that can be used in the inference engine.

There are a number of techniques that can be applied to acquire knowledge from the human experts, for example interviews, reporting techniques, goal trees and decision networks to name a few.

The following diagram shows the architecture of a typical ES.

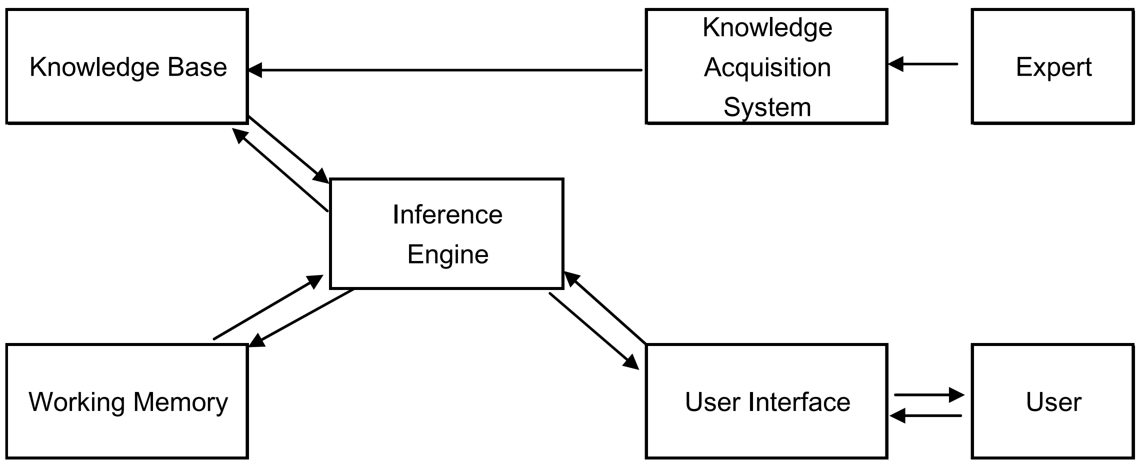


Fig. 3.1 Architecture of Expert System

3.2.3 Advantages and Disadvantages of ES

ES brings a number of advantages, including:

- It improves productivity by speeding up problem solving time.
- Due to its strong knowledge base it allows consistent answers and solutions to be provided for repeated or similar questions.
- High availability of expertise and no requirement for presence of real human experts – expertise are gathered during design stage and updated as the expertise in the specific domain develops, therefore expertise is accessible any time anywhere. This is particularly important when expertise is scarce or prohibitively expensive.

- Substantial cost saving to organisations, via a number of ways e.g. by reproducing copies of the ES which can then be used by multiple users and by cutting personnel costs.
- Serves as a reserve of human expertise and a strong and permanent database of information.
- It integrates the use of confidence factors, which is similar to confidences humans would use while reasoning.
- Offers clarity of logic or reasoning of decision making.
- It works according to strict mechanism thus eliminates common human mistakes.

Nevertheless, ES also has a few disadvantages and limitations:

- Knowledge acquisition has proven extremely difficult and time consuming during the past decades of development.
- It lacks common sense which is essential in some decision making – after all a machine cannot think as a human does.
- It is only useful when normal questions are asked and lacks creativity.
- Without regular updates to the knowledge base, the ES can easily be outdated.
- An ES is only effective in the specific domain and is often not able to detect a problem which is outside of the scope of the particular ES.

3.2.4 Recent Development of General ES

Originally, the design and development of ES was generally restricted to a few highly academic areas, such as biochemistry, physics and geology etc. As the ES developed, it has been used commercially in a large number of industries, including banking, insurance, computer games, military and space. The other main areas where ES is applied include interpretation, prediction, diagnosis, transportation, design and planning, monitoring and control etc.

1) Artificial Neural Networks

Artificial neural network (ANN) is a concept derived from the concept of Biological neural network (BNN). While BNN consists of real biological neurons and functions based on the central nervous system, ANN consists of a group of artificial neurons and is created by mathematical, electronic or other simulating methods to mimic the structure and function of BNN. Theoretically, the essence of ANN is how to realise the learning from BNN and use that learning to solve problems. This, in practice, is related to how ANN learns from a variety of data and makes correct decisions (i.e. solves problems) by simulating the function of BNN.

Although the history of ANN dates back to 1940s when Warren McCulloch and Walter Pitts proposed the first model of ANN (the famous McCulloch-Pitts Model) (Wang, 2007), the development of ANN entered its “full development period” in late 1980s and continues to be a major direction of the development of ES.

Traditionally, the main areas of application of ANN are pattern recognition, robot control, data processing and manufacturing. Recent directions of developments of ANN include radial basis function network (Moradkhani et al, 2004; Celikoglu and Cigizoglu, 2007) and others.

2) Fuzzy Expert System

For conventional ESs, one of the main bottlenecks is the difficulty of processing uncertain information or questions. Conventional ESs are used to process certain data such as choosing between two preset numbers, but when facing with a question involving uncertain data, the outcome are often unsatisfactory. This is where fuzzy logic was introduced into the area of ES.

Basically, a fuzzy expert system is an ES that uses fuzzy logic instead of Boolean logic

(Azadeh et al, 2008)). It is based on fuzzy collection theory and the inference process of a fuzzy ES is usually carried out at four stages: fuzzification, inference, composition and defuzzification. Fuzzy ES is widely used in the areas of linear and nonlinear control, financial systems and data analysis etc.

3) Hybrid Intelligent Systems

While a conventional ES, an ANN and a fuzzy ES all have their own advantages, each of them may not be sufficiently strong in processing vast amount of data and detecting trends in the data. Hybrid intelligent systems (HIS) combine the use of different methods and types of ES aiming at reaching decisions and solutions with better justification.

Typically, an HIS can combine conventional ES, fuzzy ES, neural networks, and neuro-fuzzy systems. Although HIS is a promising development direction, it is worth noting that how to make the knowledge representation in these different systems compatible to each other is an essential problem.

4) Other directions of recent and future directions

Apart from the directions and areas mentioned above, there are a few other directions as well.

Firstly, the ES has been used in corporate enterprise applications, such as corporate document management and office equipment optimisation. Secondly, ES is beginning to be used as a replacement of human teachers. This is being realised by combining multiple ESs which can share and combine their knowledge bases and present the knowledge in a learnable way to the students.

3.3 Decision Support System (DSS)

3.3.1 History and classification of DSS

DSS is an interactive computer-based system that assists people in decision making activities. Specifically, a DSS is a kind of information system which, by processing and analysing raw data and knowledge, helps users to identify and solve problems and complete decision activities (Power, 1997).

According to Power (2007), the origin of DSS lies in the 1960s, when researchers started to work on computerised models to help users make decisions and solve problems. The first computer aided decision system was introduced in late 1960s. Research done at Harvard University and Massachusetts Institute of Technology was considered to be a key point in the history of DSS. In 1970s, management decision system became the research topic of an increasing number of institutes and researchers. The term Decision Support System was then introduced in 1971. In late 1970s, books and materials pertaining to the design, engineering and implementation and development of DSS were made available. At its early stage, the research and development of DSS was mainly in the areas of business and management and focused on the theory development. For example, late 1970s and early 1980s have seen the publishing of a number of important books and articles covering the subject of DSS, among them including Keen and Scott Morton's DSS text book *Decision Support Systems: An Organizational Perspective* (Keen and Morton, 1978) and Sprague and Carlson's Book *Building Effective Decision Support Systems* (Sprague and Carlson, 1982).

Apart from the development of theories pertaining to DSS, the practical application of DSS also enjoyed a fast development in 1980s, when the scope of DSS expanded from the initial business and management areas to other areas (Power, 1997). It was also recoganised at that time that DSS could be developed to assist decision making at all levels of an organisation.

The development of DSS can be clearly seen in the historical development of five types of DSS, according to Power. Outlined below is a brief historical review based on Power's summarising article in 1997 (Power, 1997):

1) Model-driven DSS

Model-driven DSS focuses on the use of models where the data and parameters input by the users are processed to assist decision making activities. The amount of data needed for a model-driven DSS is often limited so there is no need for a large database. The development of this type of DSS saw a few important building tools introduced in 1970s and 1980s. Among these tools are Interactive Financial Planning System (IFPS) which was introduced in late 1970s and Expert Choice which was released in 1983. In a review paper published in 1988, it was indicated that model-driven DSS has certain limitations. A more recent development of Model-driven DSS includes the concept of model-driven spatial DSS (SDSS) which was introduced in the literature in 1995 (Crossland, Wynne, and Perkins, 1995).

2) Data-driven DSS

A data-driven DSS "emphasizes access to and manipulation of a time-series of internal company data and sometimes external and real-time data" (Power, 1997). Among the first data-driven DSS are the systems built at American Airlines in 1970-1974, and the executive information systems (EIS) and executive support systems (ESS) developed in 1979. The latter two were built on the basis of single user model-driven DSS and on the support of relational database products. The breakthrough development of Data warehousing and On-line Analytical Processing around 1990 saw the scope of DSS being broadened. Researchers at Procter & Gamble's and Xerox contributed significantly to the development of these systems in 1980s. A more recent example is the data-driven DSS developed by Wal-Mart in 1995-1997.

3) Communications-driven DSS

Communications-driven DSSs are DSSs that use communication technologies to assist users in decision making and problem solving. Communication technologies such as audio and video conferencing technologies are the fundamental part of communications-driven DSS. Discussions and researches about the type of DSS were noticed in the academic field as early as early 1960s. Dr. Douglas C. Engelbart, the inventor of the computer mouse and groupware is one of the pioneers in this area. In 1980s, Group Decision Support Systems were developed, which was initially not more than academic research but later evolved into commercial product in early 1990s. Later on, the development of internet helped boost the development of more advanced communications-driven DSS.

4) Document-driven DSS

Document-driven DSSs put emphasis on the retrieval, manipulating and analysis of documents, including scanned files, hypertext files, image and even videos, in assisting the decision making activities. The type of DSS is extensively used in commercial environment due to its capability of processing a large variety of documents, from corporate policies and business processes/manuals to daily memos and correspondence. Particularly due to the large scale of the documents database, an advanced search engine is of essential importance to a document-driven DSS. The earliest discussion about this type of DSS was seen in mid 1940s. The research about these systems continued throughout 1970s, when its use in managerial areas was one of the focuses of the research. The breakthrough in the development of document-driven DSS happened in 1990s when researchers identified solutions of how to find the right documents that can help the decision making more effectively. The Internet, again, is a significant booster of the development of this type of DSS, thanks to the vast resource of documents it can provide the users with.

5) Knowledge-driven DSS

Knowledge-driven DSS features an “expertise” database of a specific domain, which assists users in decision making and problem solving through rule-based reasoning functions. The type of DSS often resembles a typical ES, which is based on a knowledge base of a specific domain. Especially in recent years, the development of ES and knowledge-driven DSS are usually closely linked to each other. Similar to ES, this type of DSS is widely used in commercial areas such as manufacturing planning and financial activities.

3.3.2 DSS Architecture

According to Sage (1991), the main components of a DSS include a database management system (DBMS), a model-base management system (MBMS), and a dialog generation and management system (DGMS).

Database management system (DBMS)

The DBMS stores the data, information and document of a specific domain. It practically functions similarly as a knowledge base of a typical ES. A DBMS of a DSS should be able to advise the users of the data and documents available to the user and instruct how to access them.

Model-base management system (MBMS)

The MBMS of a DSS transforms data from the DBMS into information that can be used to assist the decision making and problem solving. In many cases, the users of DSS are faced with problems which are ill-structured, semi-structured or unstructured, thus the MBMS provides a platform where the user is assisted in model building.

Dialog generation and management system (DGMS)

The DGMS of a DSS resembles the user interface of a typical ES, however, user interface is generally considered to be a broader concept than DGMS. Its function is to provide the user with an interaction point with the system. The DGMS/user interface helps determine whether the particular DSS is suitable for the problem faced by the user and whether the results, which are decisions or options for solutions, will be worthwhile. With the aid of the DGMS, the user has access to the recommendations of the decision/problem solution.

While Sage's view about the structure of a typical DSS is widely recognised, another type of the structure is also widely used, which is database-model-user interface (Sprague and Carlson, 1982). Regardless of the types, a DSS gathers information, data and document which are used to support the technological and managerial decision making activities. A DSS always has inputs (data, information etc), user knowledge, outputs (data transformed by the MBMS) and decisions/recommendations to the solution.

The figure below outlines the architecture of a typical DSS:

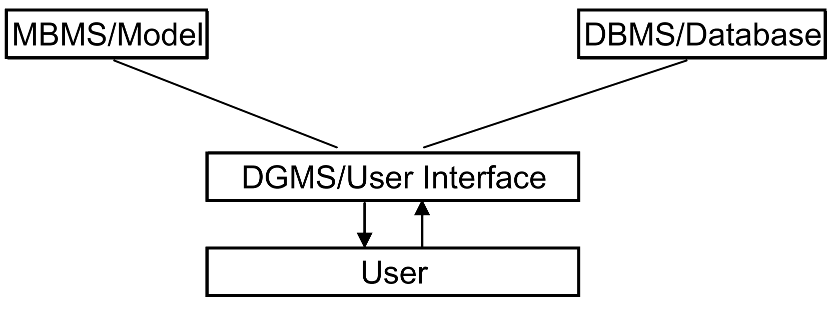


Fig. 3.2 Architecture of Decision Support System

3.3.3 Advantages and Disadvantages of DSS

A large number of researchers have provided valuable view in the advantages and disadvantages of DSS. The summary below is mainly based on the research and review done by Power (Power, 2002; Power, 2006).

The advantages of DSS are mainly seen in the following areas:

- Cost saving

Although decision making is the ultimate aim of a DSS, it has been widely recognised that DSS has brought a large number of companies and organisations huge cost saving. The saving is normally seen in reduction of number of staff needed and saving in infrastructure. This is particularly significant to an organisation where decision makers have high turnover rate and where training is slow or processes are poorly controlled.

- Time saving

Most of DSS have demonstrated their ability of reducing decision cycle time. The improved personnel efficiency when using DSS is also a main contributor of the time saving. In many cases, users find that reaching the decision using a well-designed DSS always proves more efficient than working with human experts.

- Improved quality of the decision

DSS not only assists user in reaching decision more cost-effectively, it also provided decision of high quality. Although this is documented via subjective perception of the decision recommended by a DSS, there have been reports which show that the main reason mangers use DSS is to obtain accurate information.

- Improved communications among decision makers

Thanks to the communications technologies, many DSS are shared and can be used by multiple users, in contrast to a single-user DSS. Communications among the decision makers are therefore enhanced by the enhanced accessibility and sharing of information, facts, data and documents. Based on Power's classification of DSSs, this advantage is mostly reflected in communications-driven DSS (Group Decision Support System), model-driven DSS and data-driven DSS (Power, 2006).

- Competitive advantages over competition

A DSS does not necessarily mean enhanced competitiveness to an organisation, however in recent years, especially with the aid of web-based DSS, competitiveness of an organisation can be achieved by DSS via costs reductions, supplier and customer relations and managerial effectiveness.

- Improved staff learning and training

While this is again not the original aim of DSS, it has been used as a useful tool for staff learning and training. Through DSS, members of staff learn new concepts/approaches and can have a better understanding of the business.

Apart from these main advantages, increased decision maker satisfaction and increased organisational control and targeted marketing are also advantages of DSS that are commonly recognised.

Apart from the advantages, the disadvantages of DSS are also discussed by researchers in a number of papers. The summary below outlines some of the main disadvantages of DSS (Power, 2006).

- Overemphasising decision making

As DSS are being used by an increasing number of organisations, some managers tend to use it whenever they think is suitable, in some cases without thoroughly considering the appropriateness of the DSS to the particular decision situation. This leads to overemphasising decision making and undermines the significance of other factors to the success of the organisation, such as political and social factors.

- Transfer of power

Generally, the use of DSS replaces a large part of the function of human in decision making. It however undermines the importance of human innovation in the decision making activities.

- Unanticipated effects

It has been documented that implementing DSS may in some cases bring unanticipated effects. An example is the DSS overload the users with information and that practically decreases the effectiveness of decision making.

- Information overload

The strong database of DSS means that the amount of information provided by DSS is substantial and managers are often overloaded with information, which is often misleading and reduces effectiveness. This is a common problem and it can often be corrected by monitoring and managing the information load.

Other disadvantages of DSS include obscuring responsibility, false belief in objectivity and status reduction. Many of the disadvantages of DSS can effectively be avoided or corrected by adequate staff training and regular monitoring and control.

3.3.4 Recent Development of DSS

As discussed in above sections, DSS has been widely used in business and management areas. In recent years, the implementation of DSS has extended to a large number of industries, and outlined below are a few examples of the development of DSS' implementation in some of the industries.

Energy Planning

In the energy industry, the operations often require the analysis of a large amount of information. Decision support system was proposed to be integrated into some of the operations, such as design of power plant, in 1990s (Dargam and Perz, 1998). In recent years, DSS has been used more widely in the industry. Ramachandra et al's research in regional domestic energy planning has proposed to analyse energy consumption at domestic sector using DSS, which provides a better understanding of the process and assists decision makers in timelier decisions (Ramachandra et al, 2005).

Manufacturing

DSS has been commonly used in the manufacturing industry, especially since it has been combined with the development of World Wide Web (WWW). Product design and manufacturing methods are changing constantly. Therefore, how to make effective decisions in these areas has become the focus of researches in recent years. Kengpol and O'Brien outlined in their article (Kengpol and O'Brien, 2001) that a decision support tool to select appropriate tools and technologies to achieve prompt product development, which would enable the organisation to gain competitiveness through the improved effectiveness. In more recent years, the research in the manufacturing industry has become more detailed. For example, in the product development area, knowledge-based DSS was proposed to assist rapid one-of-a-kind product development in 2006 by Xie (Xie, 2006). Production planning and scheduling has also been a main area where DSS is widely discussed and used (Jang et al, 1996; Ecker et al, 1997; Zhou et al, 2008).

Environment

DSS has been increasingly used in the research related to environment in recent years. For example, it has been considered to be used in assessing water quality control (Assaf and Saadeh, 2008). Environmental decision support systems have become one of the main sub-groups of DSS and continue to be one of the most discussed areas (Matthies et al, 2007; Booty et al, 2001).

Although DSS is mainly used in business and management areas, it is clearly documented in recent years that it is being utilised in an increasing number of industries and areas, including food production, agriculture, financial activities, production and operations management, strategic management and engineering etc.

3.4 Comparison and integration of ES and DSS

ES and DSS are widely used in a large number of industries in recent years, due to their strength in problem solving and decision making. They are also often compared with each other, due to the similarities of their characteristics. The disadvantages of these two kinds of information systems are discussed in the above section. In the expressing and evaluation of measurement uncertainty, ES and DSS need to be integrated in order to (a) combine the strength of both systems and (b) minimise the limitations of the systems. This section discusses the differences between the two kinds of systems and then discusses the integration of them.

3.4.1 Comparison of ES and DSS:

According to Finlay (Finlay, 1990), the components of ES and DSS and their methodologies are very similar, although the terminology used for them may be different. For instance, the dialog generation and management system (DGMS) of a DSS functions basically the same as a User Interface of an ES, although the latter is normally considered to be a broader concept.

Despite their similarities, ES and DSS do differ from each other in the following main areas:

1) Objectives and intents

A DSS is an interactive system that supports technological and managerial decision making using data and models, whereas an ES is often used in a specific (and often small) domain and aims at helping users solve problems by mimicing the function of human experts. A DSS is more likely used to solve ill-structured, semi-structured, or unstructured issues (Ford, 1985; Sage, 1991), often in the strategic and tactical areas, while an ES may be more often used in operational situations, which are often well-structured (Sage, 1991).

2) Users

The users of DSS may be the employee of the organisation, their customers, their suppliers or other stakeholders. In comparison, the ES allows human expert to input their knowledge in a specific domain into the system (a “teaching” process) so that the system can assist users who are often non-experts or less-expert.

3) Results from the systems

In a DSS, the users get information/reports which are analysed or generated by the system and make their own decision based on the information obtained. In contrast, in an ES, the inference process of the system aims at suggesting a final decision or several options to the users.

3.4.2 Integration of ES and DSS

As both ES and DSS have its own strength and limitations, in recent years, these two kinds of systems are often used as complementary approaches (Arentze et al, 1995). This complementarity has led to the integration of ES and DSS to achieve higher performance in decision making and problems solving. After the ES and DSS are integrated, it is commonly called Intelligent Decision Support System (IDSS) or Knowledge-based Decision Support System (KB-DSS). It is also called Intelligent Support Systems, Expert DSS (EDSS) and Expert Support Systems (ESSs) by different researchers.

In a KB-DSS, the ES often serves as one of the main components (Klein and Methlie, 1995). As discussed earlier in the thesis, ES is a one of the sub-category of AI. Thus, the ES component in the KB-DSS provides knowledge in a specific domain using AI to the DSS users. In accordance to the function of a typical ES, this component simulates reasoning and explains the reasoning as well as provides a result to the user, while the other main component, the conventional DSS, carries out data and model management and the decision methodology.

According to Klein and Methlie (1995), there are a number of different approaches to integrate DSS with ES, including incorporating expertise in a specific domain, explaining the reasoning and conclusion using ES technology and using ES to improve the intelligence of DSS to name a few. With the integrated functions, the KB-DSS can basically achieve all functions which traditional ES and DSS can achieve.

KB-DSS brings a number of benefits to the users. A few of its main advantages are outlined below:

1) The ES enables the systems developer to integrate expert knowledge to complement the existing models in a DSS (Arentze et al, 1995).

2) ES and DSS are often integrated where the functions of both kinds of systems are needed. Instead of using two separate systems in one common domain, the KB-DSS significantly improves productivity of the problem solving or decision making activities (Zopounidis et al, 1997).

3) As a consequence of the integration, KB-DSS saves the users not only the time needed to reach the final solution/decision but also the cost invested into the development of the system (one system vs. two systems).

4) Unlike traditional DSS, KB-DSS provides a large amount of expertise obtained from human experts to the end user. This knowledge database can also serve as a reservation of information and data.

3.5 Web-based ES and DSS

Before the appearance of the World Wide Web, the application of ES and DSS was restricted to a small group of people (standalone ES or DSS) and the sharing of information was restricted by the lack of a broader platform. The earlier standalone ES and DSS had a number of obvious limitations due to this. As ES and DSS share similar characteristics, such as requirement of a database and user interface, a number of limitations of earlier ES and DSS are discussed together in this thesis. Firstly, as the ES and DSS are based on independent PC platforms, it is difficult to obtain knowledge from different sources. Secondly, updating the system and its database and interface is made extremely difficult as it is often needed to be carried out by many separate steps and was very time consuming. Thirdly, dynamic information sharing was restricted by the lack of a shared platform. For ES, its limitations are even more obvious in this area, as the accessibility of the expertise provided by ES was very much limited to the use of the standalone system (Duan et al in 2004).

The breakthrough in this area came around 1995, when the Internet provided the technological possibility of expanding the accessibility and capability of DSS and ES (Power, 2007). Take DSS for instance, in 1995, a number of papers proposed the integration of Internet and decision support. Power and Kaparthi pointed out in 1998 that corporate Intranet and global Internet had become the focus of the development of DSS (Power and Kaparthi, 1998).

A web-based DSS is “a computerized system that delivers decision support information or decision support tools to a manager or business analyst using a Web browser such as Netscape Navigator or Internet Explorer” (Power, 1998; Power, 2000).

A web-based ES involves distributing the application of the ES via the Web (Dokas, 2005). As the main function of an ES is to simulate the function of human experts, web-based ES enables the knowledge of a specific domain to be delivered to users through web (Duan et al, 2004).

This section focuses on the concept, advantages and disadvantages of web-based ES and DSS. The building of web-based ES and DSS, their integration and application in the evaluation and expression for measurement uncertainty will be discussed in details in the following chapters.

In recent years, integrating web with ES and DSS has become a main direction in the industry. Web-based ES and DSS have brought substantial benefits to the users but also pose a few limitations.

3.5.1 Advantages of Web-based ES and DSS

Generally, both web-based ES and DSS enjoy the following advantages:

1) Enhanced system accessibility

Thanks to the development of Internet, the users no longer need to install the traditional standalone ES or DSS on their PC. Instead, they can easily access the system online, which give them the access to the most up-to-date version of the system (Power, 2000b). This is commonly recognised to be the most important improvement by integrating the Internet with ES and DSS.

2) A stronger database

Once the ES and DSS are provided on the basis of the Web, the human experts (for ES) and the users (for both ES and DSS) can submit the knowledge, information, data and feedbacks to the system via the Internet (Duan et al, 2005). This is also considered to be one of the most import contributions of the Internet to ES and DSS.

3) More effective system update and maintenance

The Internet allows the ES or DSS to be managed and monitored from a centralised location, where the operators are based. This location collects users' feedback and carries out regular evaluation of the system. This allows for more timelier and effective system update and maintenance.

4) Reduced technological barriers and the tasks are accomplished at lower cost

Power advised in his paper in 2000 (Power, 2000a) that the Internet technology has reduced the technological barriers between the users and the systems. Moreover, due to the effectiveness that the web-based ES and DSS provides, the decision making or problem solving processes can be completed for less costs.

5) Improved effectiveness of problem solving and decision making

All the above-mentioned advantages allow the Web-based ES and DSS to carry out their main function, problems solving (ES) and decision making (DSS) more cost-effectively. This is mainly due to the platform provided by the Internet, which enables the information, data and feedback of the system to be shared and distributed more effectively.

Apart from the above-mentioned general advantages, ES and DSS also bring specific benefits due to the differences in their architecture and design processes.

ES's unique benefits:

1) Knowledge acquisition is made easier since the Internet makes the knowledge source much broader and more diverse (Duan et al, 2004). The acquisition is also made feasible at a distance from the human experts in the specific domain.

2) Explanation and justification of result are also enhanced by the Internet. In further development of this, the future web-based ES should integrate an online real time communications between the user and the human expert (Duan et al, 2004).

DSS's unique benefits:

1) Due to the Web infrastructure, the systems can deliver the decision support to an organisation's customers, suppliers as well as other stakeholders (Power, 2000).

2) The web-based DSS enhances the consistence of the decision making for repetitive tasks. It also helps decision makers make decisions more quickly and more predictably. In addition, it boosts the dissemination of "best practice" analysis (Power, 2000).

3.5.2 Limitations of Web-based ES and DSS

Although the web-based ES and DSS have been developing rapidly in recent years, there is still need for more research and improvement in this area. The commonly discussed limitations of them are outlined below:

1) Information overload

The traditional standalone ES and DSS already have the problem of information overload. This problem is only enhanced by the integrating of the Internet, which provides a broader source for data and knowledge.

2) The compatibility with the Web

In Duan et al's paper (Duan et al, 2004) about web-based ES, they pointed out that traditional development methodologies that work well in the traditional ES may not suit the Internet platform. This applies to not only ES but also many information systems in general.

3) Verifying and filtering of expertise and knowledge

As expertise and knowledge are acquired via Internet, how to ensure the information, data and knowledge input by users (for both ES and DSS) or human experts (for ES) are accurate and appropriate is therefore made difficult and if it is to be done adequately, a special workforce needs to be established to monitor and control the knowledge flow, which requires both time and cost. This problem is particularly obvious for ES, because how to balance the different information from several online experts poses a challenge to Web-based ES (Duan et al, 2004).

4) Security concerns

As all information is communicated and distributed via Internet, ensuring the security of the information has become a main challenge to the developer of web-based ES and DSS. This is especially significant when the system is linked to an organisation's intranet, which may contain confidential information of the organisation.

Some of the disadvantages above may require a large amount of resource to tackle, while some of them can be optimised during the design stage or via closer monitoring during the implementation stage or through more adequate staff training. Most importantly, these limitations do not in any way undermine the advantages of using web-based ES and DSS, as their strengths significantly improve the performance of the systems.

3.6 Summary

In this chapter, the concepts of ES and DSS are discussed, accompanied with a detailed introduction and discussions on their characteristics, architecture, advantages and limitations, supported by articles and papers from the literature. A comparison between these two types of information systems is also carried out.

Based on the advantages and limitations of ES and DSS, this chapter introduced the concept of KB-DSS which integrates the two systems, incorporating the advantages of the systems for better performance.

The development of the Web has significantly boosted the performance of these systems. This chapter discusses the web-based ES and DSS, which forms the basis of the web-based KB-DSS proposed in this thesis for evaluation and expression for measurement uncertainty. The design and implementation of the proposed system is discussed in the following chapters.

Chapter 4 Methodologies

4.1 Introduction

According to the previous work (Wei et al, 2007, 2009) and the earlier literature review in Chapter 2, a brief introduction to the current software tools for uncertainty evaluation was outlined and the comparisons among them have been documented in the literature. Although there have been developments of several computer-assisted approaches in recent years, their wide applications have been impeded by some significant disadvantages, such as cross-platform compatibility, accessability, cost-effectiveness and user-friendliness. More importantly, the process of measurement uncertainty evaluation involves a number of critical analyses and decision making, including the following:

1. How to formulate the appropriate mathematical model, when the user has very limited knowledge;
2. How to decide which framework is more suitable for a specific case of measurement uncertainty, the GUM Framework or GUM SP1 MCM Framework;
3. How to provide the specific support for the user when a particular framework is followed.

In order to make uncertainty evaluation more user-friendly and easier to handle, the proposed Web-based KB-DSS approach is developed to streamline the evaluation of measurement uncertainty. Thanks to the advantages of KB-DSS discussed in Chapter 3, a Web-based KB-DSS can realise the following objectives:

1. The system can be accessed via a web browser, therefore overcoming the limitations of the traditional desktop application;
2. The high interactiveness between the user and the system improves the support to the user in problem analysis, decision making and calculation. The improved decision making consequently ensures the generation of more accurate results.

The methodology for the proposed system is shown in the figure below:

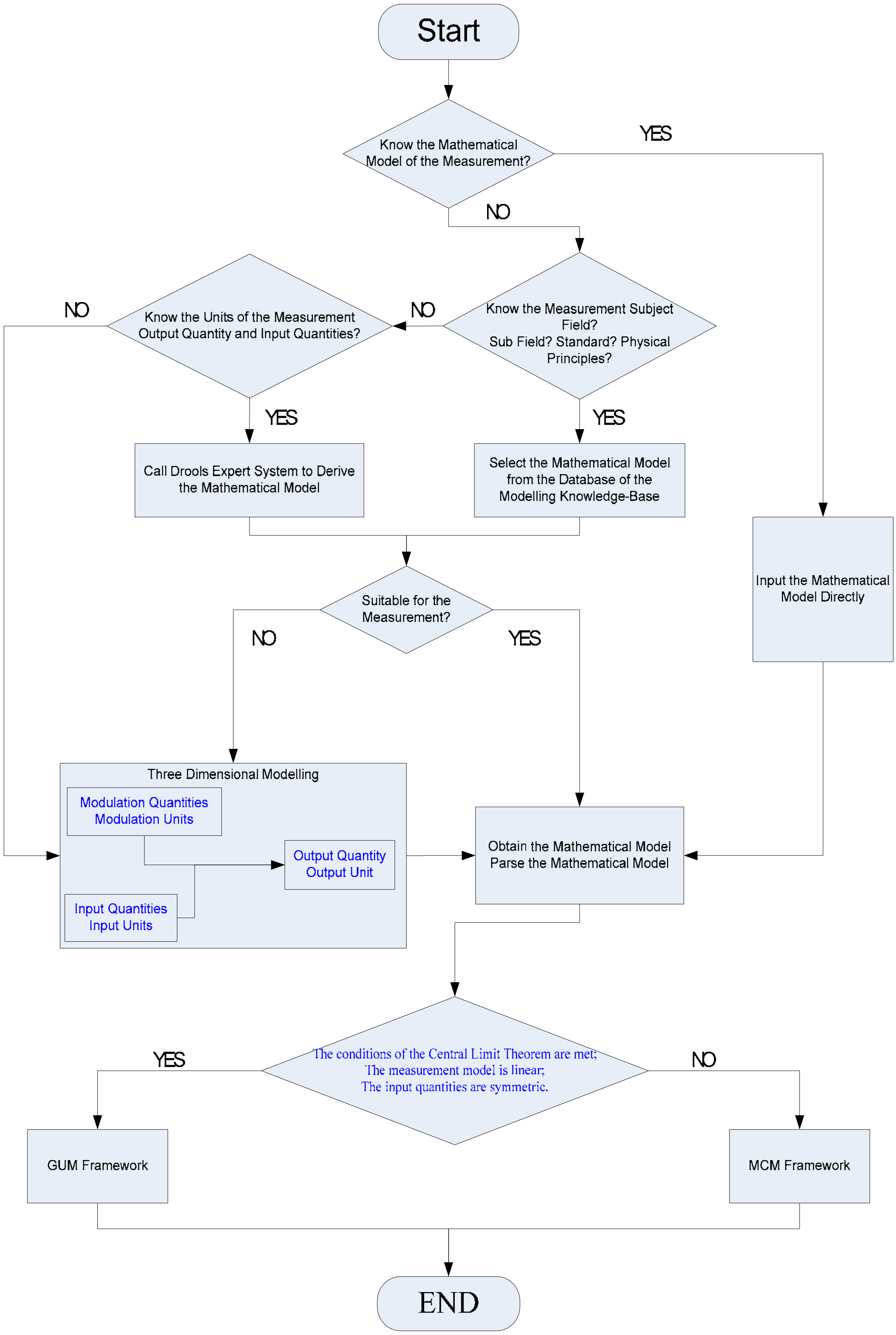


Fig. 4.1 The Methodology for the Proposed KB-DSS

In this chapter, the methodology of structuring this system is discussed.

Firstly, as a web-based application, the proposed system has adopted an architecture based upon the Apache Struts framework of Model-View-Control (MVC). Java programming is used due to its various advantages.

Secondly, in regard to the components of the proposed system, the thesis focuses on the knowledge bases, including Measurement Modelling Knowledge Base (MM KB), GUM Framework Knowledge Base (GUM KB) and MCM Framework Knowledge Base (MCM KB). These three knowledge bases are located on the server side to carry out the evaluation of measurement uncertainty.

Thirdly, on the client side, the user interface is realised by using Java Server Page (JSP). In the mean time, Asynchronous JavaScript and XML (AJAX) technology is utilised to process the requests and responses between the client and the server.

Lastly, as a complex system involves a large amount of data transfer, the efficiency and security of the data transfer is of critical importance. Java Transfer Object has therefore been used to handle the data transfer.

4.2 Development of the architecture of the web application of the proposed system

The World Wide Web has seen a breathtaking development of itself since its invention. In recent years, the term "Web 2.0" has started to be used to represent the second generation of web design and development, which enhances the information sharing and user-centred activities. The terms was proposed in 1999 (DiNucci, 1999), however, the wide recognition of the term associated with its current indication of the second generation of web development started only in 2004, after the O'Reilly Media Web 2.0 conference (O'Reilly, 2005).

It is worth noting that the term Web 2.0 does not mean the technologies associated with web development have been replaced by new ones. Rather, it emphasises on a change of how web developers and end users use and interact with the Web. Generally, Web 2.0 focuses on the user participation. The end users can not only obtain information from the web, they can also upload, manage and manipulate data on the Web, in contrast with traditional web application where the content could only be manipulated by the developer of the website. Examples of Web 2.0 include Flickr, Wikipedia, blogging websites and Youtube, to name only a few.

The user participation characteristics of Web 2.0 is often realised by technologies/technology groups such as Asynchronous JavaScript and XML (AJAX), and Adobe Flash. The development of Web 2.0 is one of main supports to the methodology proposed in this thesis. Some of its typical components (e.g. AJAX) also form a solid foundation of the methodology. The detailed discussion of this is included later in this chapter (section 4.4.2).

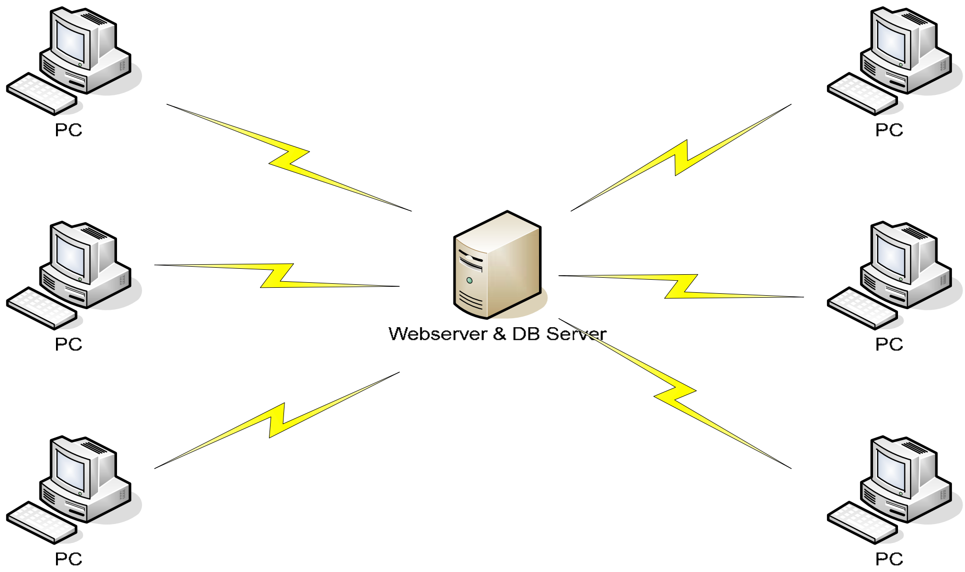


Fig. 4.2 Web 2.0 Client/Server Architecture

Fig 4.2 shows a Client/Server architecture of a typical web application. On the server side, the model-view-controller (MVC) architecture was developed, as shown in Fig 4.3. MVC was originally implemented by Reenskaug (Reenskaug, 2003). An MVC pattern is a software designing architecture where the business logic is isolated from the user interface (Java blueprints, 2002). In the proposed system, as a large amount of data transfer and different calculation methods are involved, this isolation helps in the sense that any change to the business logic does not affect the user interface, therefore significantly improving the efficiency of the system and making system maintenance easier.

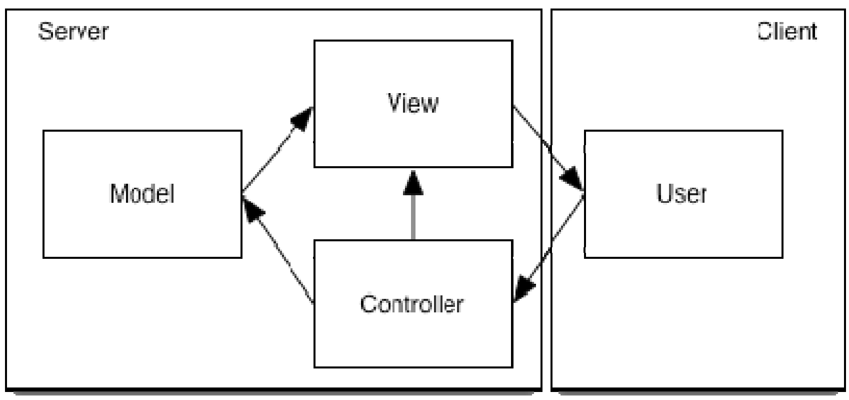


Fig. 4.3 MVC Client/Server Architecture (Steele, 2004)

As indicated by its name, a MVC pattern includes three components: model, view and controller (Eckstein, 2007). The model represents and manages data and the rules governing this data. The view renders the contents of a model into a form suitable for user interaction, i.e. managing the display of information. The controller transforms the information (input from the user) into actions for the model to perform (Steele, 2004). The figure below shows a common MVC pattern.

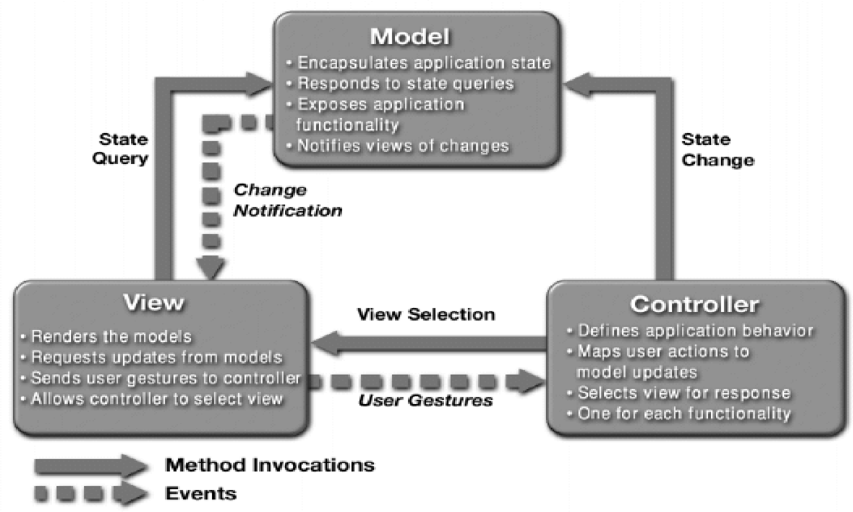


Fig. 4.4 The Model-View-Controller Pattern (Java Blueprints, 2002)

Due to MVC's strong capacity for web-based software designing, it has been widely used in developing web-based software. In the proposed system in this thesis, the Apache Struts web application framework which implements the MVC architecture pattern is used.

Apache Struts is a well-designed web application framework incorporating Java Servlet API and MVC. The figure below clearly demonstrates the relationship among the components of a typical Apache Struts framework. As a main characteristic of MVC framework, the business logic is separated from the user interface logic. Equally importantly, one of the most critical parts of the Apache Struts framework, the client/server exchange, is realised by a Java-enabled HTTP layer. Fig 4.5 outlines the structure of a typical Struts framework.

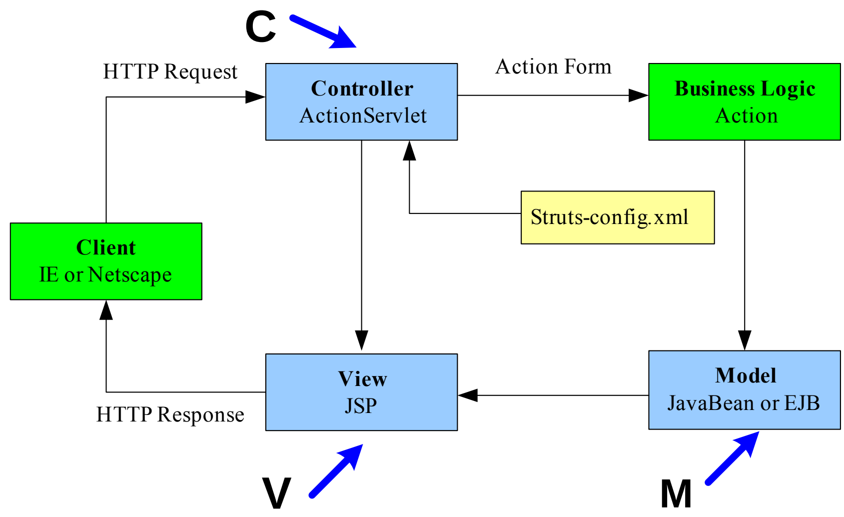


Fig. 4.5 A Typical Apache Struts Framework (Davis, 2001)

- Controller: In Struts, the controller of the MVC architecture is a Servlet, the ActionServlet. The ActionServlet provides an entrance for the HTTP requests from the client browser. It receives and distributes the requests to corresponding locations. The controller is configured using Struts-config.xml files.
- View: Practically, the View is JSP files. Struts provides a vast JSP tag base including HTML, Bean, Logic and tiles.
- Model: Here, the Model shows the state of the application. It exists in the form of Java Bean(s).
- Business logic: In this thesis, the Business logic includes the three knowledge bases, which are discussed separately below.

In this thesis Java programming is used to realise the function of the proposed system.

Java, developed by Sun Microsystems, is a high-level object-oriented multi-platform programming language influenced by other programming languages such as C, C++ and Objective-C etc (Gosling and McGilton, 1996). The initial concept of Java started to appear in early 1991 and the first publishing came as Java 1.0 in 1995. Java is famous for "Write Once, Run Anywhere" which makes it feasible to run it on multiple platforms (Sun Microsystems, 1999). A better understanding of Java can be achieved by a brief look into its main characteristics and advantages:

- Portability: As clearly conveyed by "Write Once, Run Anywhere", Java can be run on a variety of computer operating-system platforms, since Java codes are complied to bytecode, enabling it to be run regardless of computer platform (Gosling and McGilton, 1996).
- Java is influenced by other programming languages. However, Java is simplified so that it can eliminate many of the common programming errors.
- Java has strong capability of being integrated into web applications. JSP is a perfect example or Java's application associated with the web (Sun Microsystems, 1999).
- Java has nurtured the development and introduction of a number of further platforms widely used in the industry, including J2EE and J2ME, focusing on enterprise applications (Alur et al, 2003) and mobile applications respectively.

4.3 Development of the Knowledge Base

This section discusses separately the key components of the proposed Web-based KB-DSS, the three knowledge bases: Measurement Modelling Knowledge Base (MM KB), GUM Framework Knowledge Base (GUM KB) and MCM Framework Knowledge Base (MCM KB), as shown in Fig.4.6.

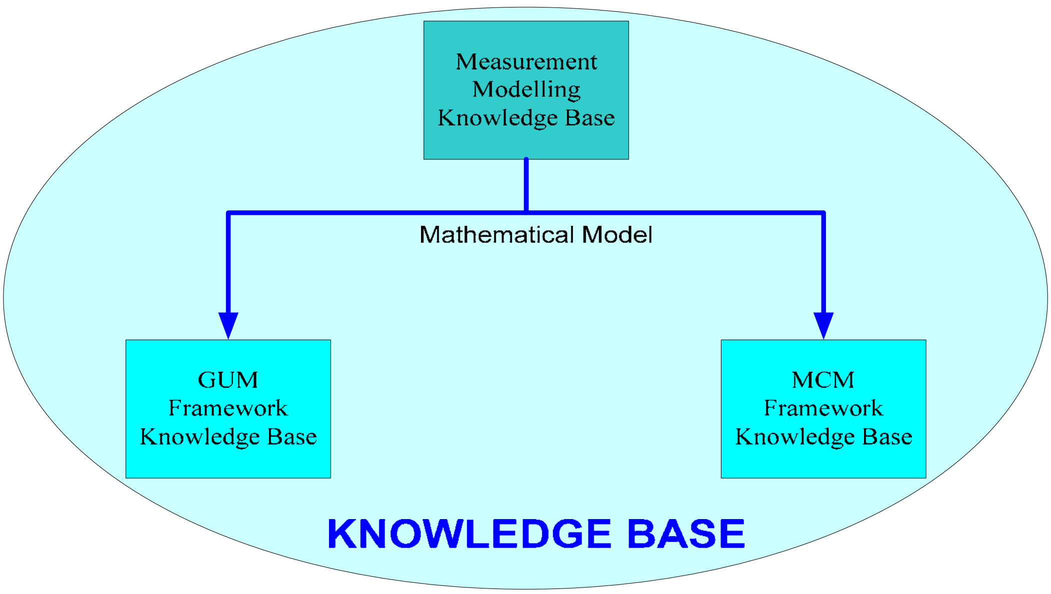


Fig. 4.6 Relationship of proposed Knowledge Bases

4.3.1 Measurement Modelling Knowledge Base

For measurement uncertainty, the establishment of the measurement mathematical model is of critical importance. A measurement modelling knowledge base is needed for this purpose and it consists of four modules. The first module, Measurement Standards Modelling Module (MSMM), provides a database of a variety of known models, from which the user can choose. If the models obtained using the first module is not appropriate to the user's specific measurement situation, the second module, SI Unit Modelling Module (SUMM), can be applied. This module asks for specific input and output information, which assists in deriving the appropriate model. Based on the theories of modern measurement system (Bentley, 1988), a third module is proposed, the Three Dimensional Modelling Module (3DMM). Compared with the first two modules, this module is more complex and needs a bigger database. In this measurement system – oriented module, the user inputs the information related to the input quantity, output quantity and the modulation to establish the mathematical model.

The models obtained via these three modules can be the same or different. These models are used in the fourth module, User Input Modelling Module (UIMM), where the user is allowed to modify the mathematical models obtained earlier, according to the specific case. Alternatively, in case where none of the first three modules can help the user locate the most appropriate model, UIMM provides the user with a platform where a complete model can be input into the module directly by the user.

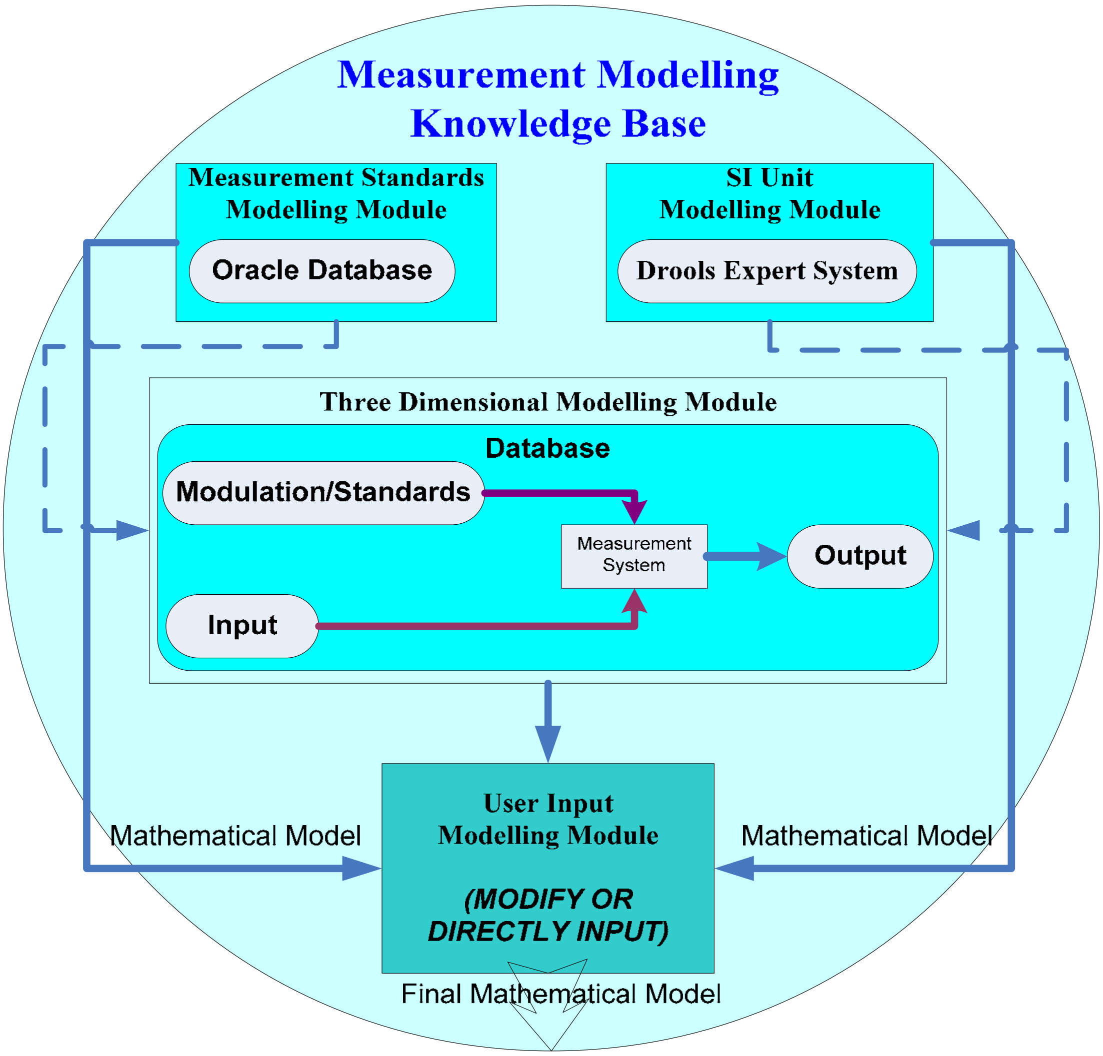


Fig. 4.7 Sub-modules of Measurement Modelling Knowledge Base

The four modules are discussed in details as follows.

1) Measurement Standards Modelling Module (MSMM)

This module aims at providing the user with a comprehensive database of models, sorted by the subject fields in metrology. It is particularly helpful to users who do not know the model for the evaluation of measurement uncertainty.

According to the European Association of National Metrology Institutes (EURAMET), metrology is divided into a number of main technical subject fields, based on BIPM's work and EURAMET's extension (Howarth, Redgrave, 2008). Fig.4.8 below outlines the subject fields of metrology, their subfields and corresponding measurement standards. It is worth noting that although the main fields are commonly recognised, the subfields are never defined and accepted officially globally.

Based on different physical principles, measurement methods, and measurement instruments, every measurement standard may correspond to multiple mathematical models. This thesis proposed to categorise the mathematical models under each measurement standards according to the physical principles. After the user has chosen the measurement standard, the appropriate mathematical model can be located by choosing the appropriate physical principle.

For example, assuming the user does not know the model but knows that the model needed is related to mass calibration, the selection process is carried out according to steps as follows:

Step 1: The user chooses “Mass and related quantities” from the “Subject Field”.
Step 2: The user chooses “Mass measurement” from the “Subfield”.
Step 3: The user chooses “Mass standards” from the “Important measurement standards”.
Step 4: The system will now display a few models from the mass standards including the mass calibration.

Step 5: Take the example in GUM SP1 9.3 for example: Based on the specific measurement, assuming the user knows that the measurement is based on Archimedes' principle，the user now can locate the model for mass calibration from the few models displayed. The model takes the form

$$m_W \left(1 - \frac{\rho_\alpha}{\rho_W}\right) = (m_R + \delta_{mR})(1 - \frac{\rho_\alpha}{\rho_W})$$

GUM SP1 9.3.1.1 explains the detailed meaning of the symbols.

Subject Field	Subfield	Important Measurement Standards
Mass and related quantities	Mass measurement	Mass standards, standard balances, mass comparators
	Force and pressure	Load cells, dead-weight testers, force, moment and torque converters, pressure balances with oil/gas-lubricated piston cylinder assemblies, force-testing machines, capacitance manometers, ionisation gauges
	Volume and density Viscosity	Glass areometers, laboratory glassware, vibration densimeters, glass capillary viscometers, rotation viscometers
Electricity and magnetism	DC electricity	Cryogenic current comparators, Josephson effect and Quantum Hall effect, Zener diode references, potentiometric methods, comparator bridges
	AC electricity	AC/DC converters, standard capacitors, air capacitors, standard inductances, compensators, wattmeters
	HF electricity	Thermal converters, calorimeters, bolometers
	High current and high voltage	Measurement transformers of current and voltage, reference high voltage sources
Length	Wavelengths and interferometry	Stabilized lasers, interferometers, laser interferometric measurement systems, interferometric comparators
	Dimensional metrology	Gauge blocks, line scales, step gauges, setting rings, plugs, high masters, dial gauges, measuring microscopes, optical flat standards, coordinate measuring machines, laser scan micrometers, depth micrometers, geodetic length measuring tools
	Angular measurements	Autocollimators, rotary tables, angle gauges, polygons, levels
	Form	Straightness, flatness, parallelism, squares, roundness standards, cylinder standards
	Surface Quality	Step height and groove standards, roughness standards, roughness measurement equipment
Time and frequency	Time measurement	Caesium atomic clock, time interval equipment
	Frequency	Atomic clock and fountain, quartz oscillators, lasers, electronic counters and synthesisers, optical combs
Thermometry	Temperature measurement by contact	Gas thermometers, ITS 90 fixed points, resistance thermometers, thermocouples
	Non-contact temperature measurement	High-temperature black bodies, cryogenic radiometers, pyrometers, Si photodiodes
	Humidity	Mirror dew point meters or electronic hygrometers, double pressure/ temperature humidity generators
	Absorbed dose – Medical products	Calorimeters, Ionisation chambers
	Radiation protection	Ionisation chambers, reference radiation beams/fields, proportional and other counters, TEPC, Bonner neutron spectrometers
	Radioactivity	Well-type ionising chambers, certified radioactivity sources, gamma and alpha spectroscopy, 4 Π detectors

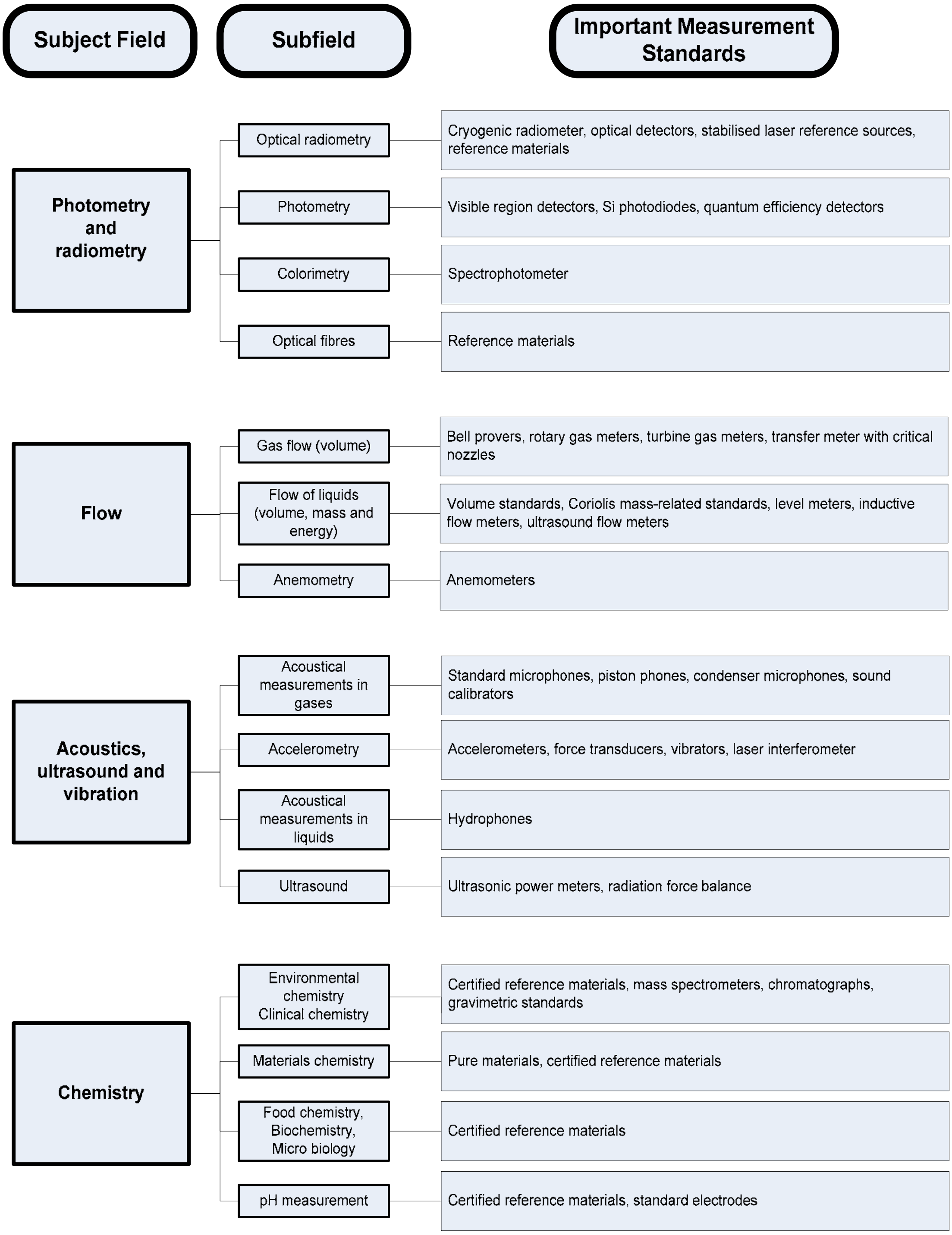


Fig. 4.8 Classification of Important Measurement Standards

(Howarth, Redgrave, 2008)

The critical goal of this module is to build a database. In the following chapters, the development of this database using Oracle relational database management system (Oracle RDBMS) is discussed in more details.

As mentioned above, the user goes through the "Subject field - Subfield - Important Measurement Standards - Physical Principle - Mathematical Model" process which assists the user in making the decision about the appropriate mathematical model on the basis of the specific measurement situation.

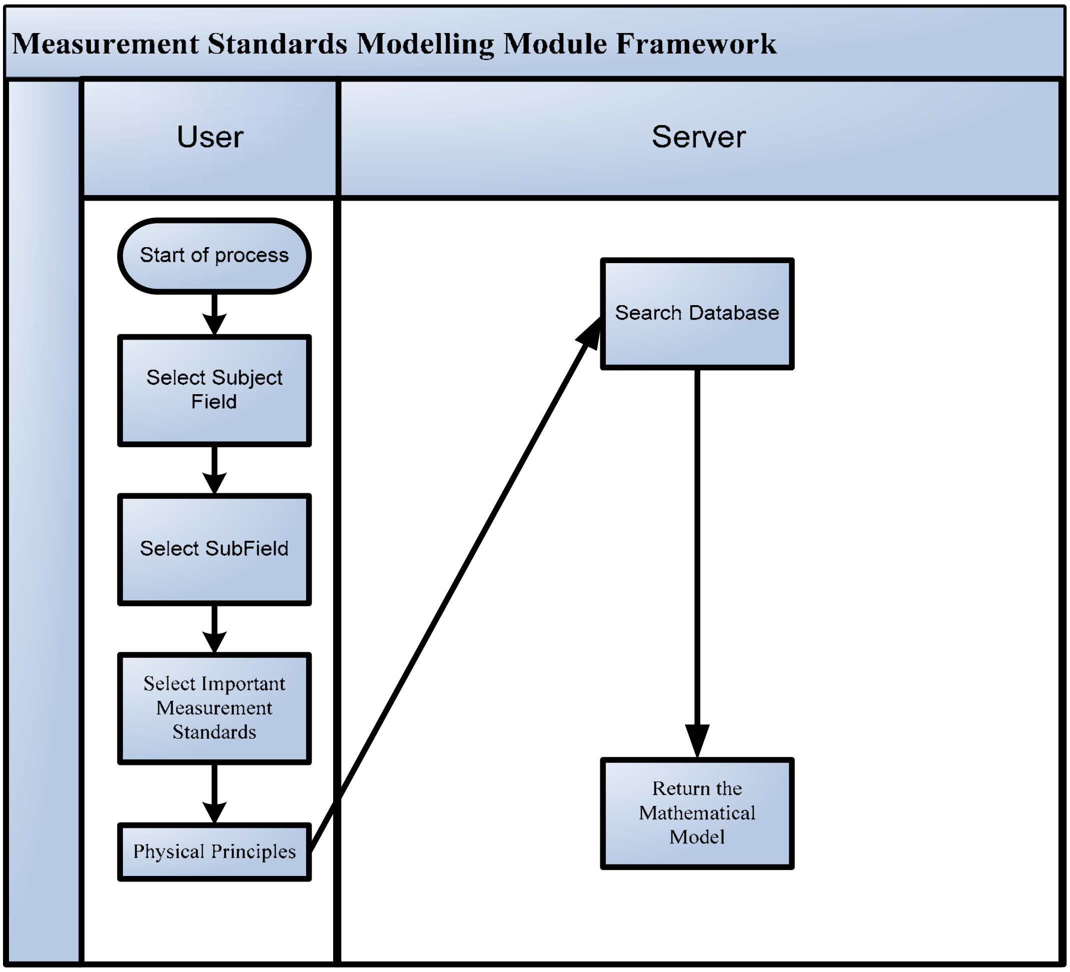


Fig. 4.9 Measurement Standards Modelling Module Framework

In many cases the user can easily find the appropriate model and therefore does not need to input further information. This decision support undoubtedly makes the decision making activity much more efficient.

Another major advantage of this module is that the database of the models can be expanded easily. During the maintenance, the database can be updated regularly to include more up-to-date or new models. The design and implementation of the database are discussed in Chapter 5.

2) SI Unit Modelling Module

The SI Unit Modelling Module is based on the International System of Units (SI), closely tied to the relation between SI base units and derived units. Due to this derivation relation, this thesis innovatively integrates this relation with an ES approach to effectively assist the user with the measurement modelling. A brief introduction of the history of SI and its architecture is discussed in Chapter 2. Fig 4.10 demonstrates the relations among the based units and derived units of SI.

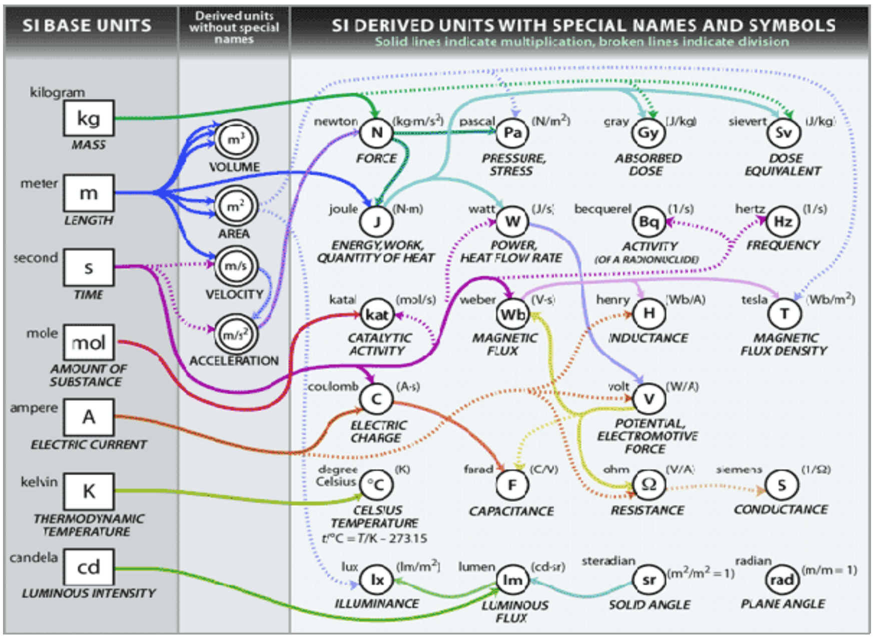


Fig. 4.10 Relationship of SI Units (Source: NIST)

An Expert System based on SI

In SI, each derived unit can be seen reflected in an expression/a model in terms of other units, either the base units or derived units. If the user does not know the mathematical model, but knows the units of the input quantities and also the unit of the output quantity, then the model can be determined based upon the derivation system of the SI. For example, if the user knows the input unit is Ω (ohm) and the output unit is A (ampere), then based on SI, the model the user is seeking after is most likely I = *V*/R. There are two things worth noting when applying this technique. Firstly, all units used in this technique must be from the SI. Secondly, if the information from the user is very limited, there may be more than one possible model. For example, in reality, there may be more than one input unit for a specific measurement, but the user is only able to provide one of them. In that case, the user needs to evaluate the result models and choose the appropriate one. Therefore, the more information the user can provide, e.g. more input units, the more likely that the appropriate mathematical model can be determined. In Fig. 4.11 below the framework of this module is outlined.

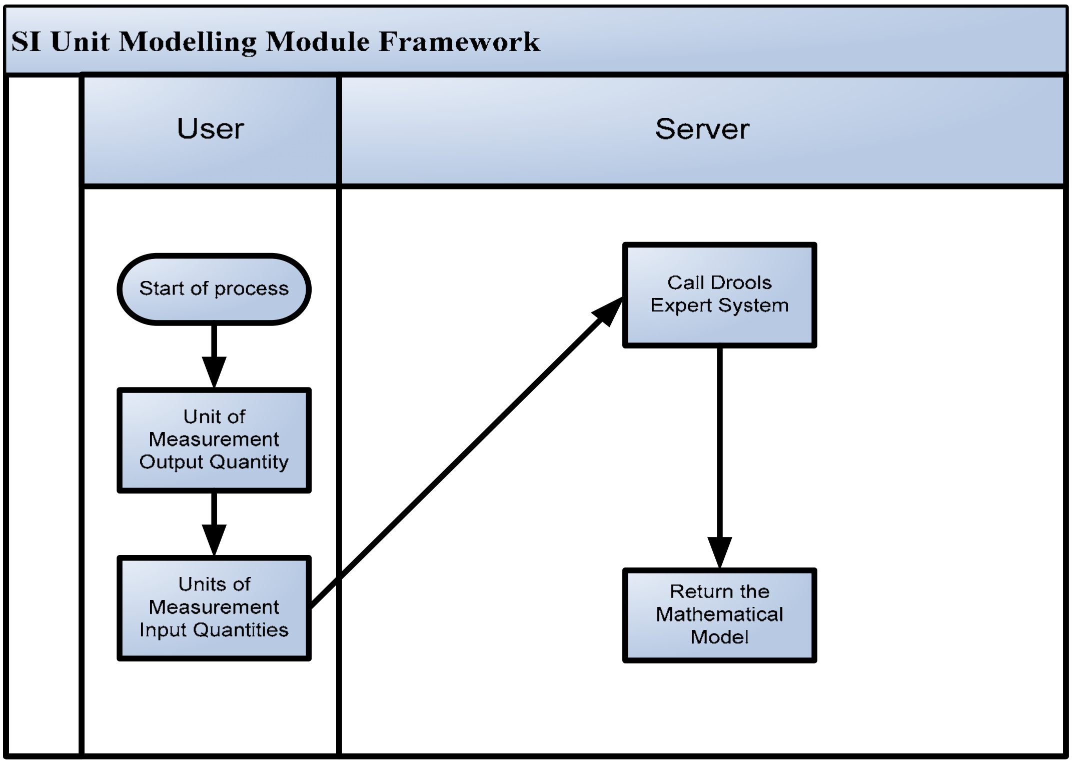


Fig. 4.11 SI Unit Modelling Module Framework

In the methodology proposed in this section, the SI unit technique is realised by using the methodology of ES. The expressions/models from the SI are included in the knowledge base. The unit information from the user is put in the working memory. In the inferring process, the typical “when/then” (the same as IF/THEN) clauses are used. To build the ES based on SI, firstly the relationship among the SI units are analysed using the binary tree data structure. Fig. 4.12 takes Ω for instance to show how this is done. The derivation of the model is carried out using Tree-traversal approach. The associated design and implementation are discussed in the following chapter.

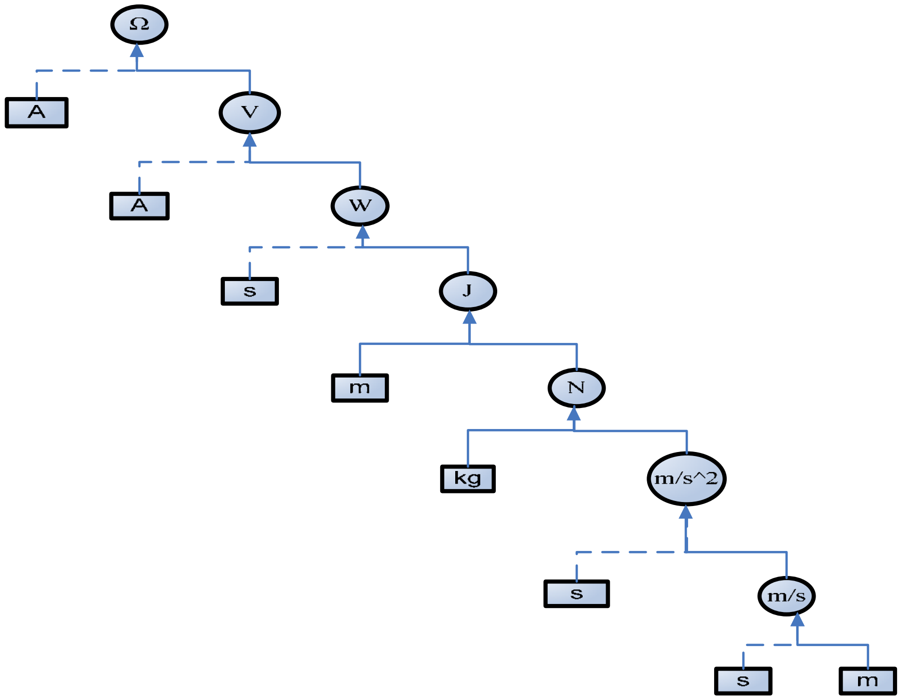


Fig. 4.12 Binary Tree of the Unit Ω in SI

Below are several examples to demonstrate the inferring process in this technique:

Example 1:

when

input unit = m *and* output unit = m

then

$Y = X_1 + X_2 + \ldots + X_n$ (This is a typical Additive model.)

Example 2:

when

input unit = m *and* output unit = m²

then

$Y = X_1 \cdot X_2$

Example 3:

when

input unit = Ω *and* output unit = A

then

$Y = \frac{X_1}{X_2}$ This is based on the classic Ohm's law $I = \frac{V}{R}$.

In this case, there is more than one result model, therefore the user needs to choose the most appropriate one based on the specific situation.

To sum up, the SI Unit Module proposed above integrates the ES with the SI, aiming at assisting the user with modelling, when the user needs to find out the correct model using limited information. This is realised by establishing a knowledge base with models based on SI. User's input into the working memory and the "when/then" inferring process enable the system to generate model (or a selection of most likely models from which the user can choose) to the user. This technique particularly helps when the users do not have all necessary information for modelling.

3) Three-Dimensional Modelling Module (3DMM)

Although the first two modules provide great user support on measurement modelling, the mathematical models contained in these two modules are more suitable to be used by testing labs. A more general modelling module intended for industrial fields is therefore needed for wider practical applications.

According to the 3rd edition of VIM (JCGM, 2008) a measurement system is "a set of one or more measuring instruments and often other devices, including any reagent and supply, assembled and adapted to give information used to generate measured quantity values within specified intervals for quantities of specified kinds". Fig. 4.13 outlines the structure of a typical measurement system.

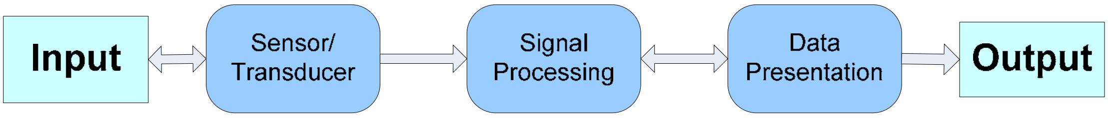


Fig. 4.13 A Typical Measurement System Architecture

An object-oriented approach to the description and modeling of measurement systems was proposed by Yang in 1998 (Yang and Butler, 1998). This framework can be used for the classification and systematic organisation of measuring devices or systems. Furthermore, in 2005, Yang presented a novel representation, which is essentially a transducer conversion logic or language (TCL) in a measurement system. Using two-port and three-port transducers as basic building blocks, it can be utilised to model any measurement system (Yang, 2005). Based on this approach, the intermediate stages in Fig. 4.13 can be characterised by modulation, as shown in Fig. 4.14, i.e. a 3D architecture.

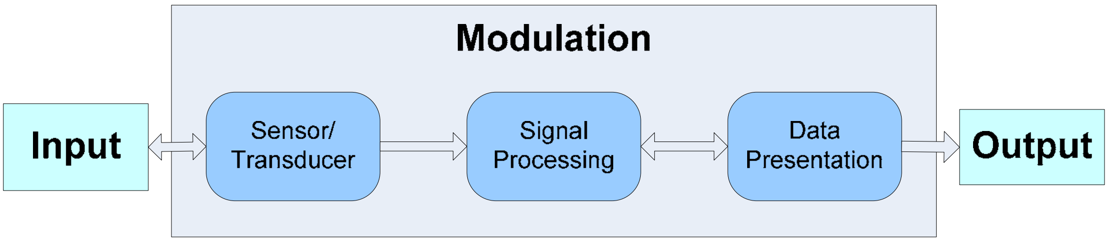


Fig. 4.14 Three Dimensional Measurement System Architecture

Once the quantities and units of the output, input and modulation are determined, the mathematical model is determined accordingly. For example, in the DC Current measurement system:

Firstly, the output quantity is determined as Current with corresponding unit A;
Secondly, the input quantity is determined as Volt with corresponding unit V;
Lastly, the general modulation quantity is determined as Resistance with corresponding unit Ω.

With the above information, the mathematical model can consequently be determined as

$I = \frac{V}{R}$. Table 4.1 demonstrates the procedure of this example:

Known Information						**Result**
Output Information *(Single)*		Input Information *(Can be Multiple)*		Modulation Information *(Can be Multiple)*		Mathematical Model *(Single)*
Quantity	Unit	Quantity	Unit	Quantity	Unit	$I = \frac{V}{R}$
Current	A	Volt	V	Resistance	Ω	

Table 4.1 Example of 3D Modelling Procedure

A complex database can be established with sub-databases corresponding to Input, Modulation, Output and the functional representation of the measurement system/mathematical model. By visiting this module, the user can submit required information and then this database is visited. The logic of this database is shown in Fig. 4.14.

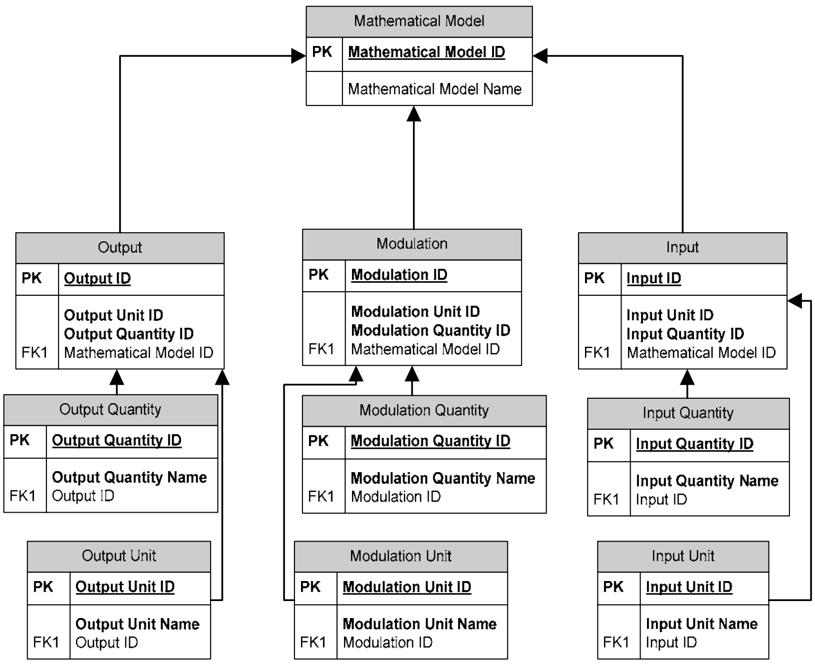


Fig. 4.15 Entity Relationship Diagram of 3DMM Database

The 3DMM provides great potential for extensibility, i.e. it can be used in conjunction with the MSMM and SUMM. More specifically, for SUMM, the output unit and input units provided by the user can be seen as the determination of the output and input information in a 3DMM. Similarly, for MSMM, the measurement standard chosen by the user can be seen as the determination of the input and modulation information in a 3DMM.

4) User Input Modelling Module

As mentioned earlier in this chapter, this module allows the user to either modify the mathematical model obtained from the other three modules, or directly input mathematical model into this module.

Basically, the UIMM serves as an interface between the modelling modules and the following knowledge bases, the GUM Framework Knowledge Base and the MCM Framework Knowledge Base, by providing the mathematical model to them. The interface function is realised by different methodologies for the GUM and MCM KB. For GUM KB, the UIMM is enabled by integrating symbolic computation in the GUM framework. For MCM KB, it is realised by transferring the mathematical model into a format which is readable to Matlab. The design and implementation is discussed in details in the following sections.

4.3.2 GUM Symbolic Computation Knowledge Base

In this section, the evaluation of measurement uncertainty is implemented based on the GUM framework. Since its publication, a number of software tools based on GUM uncertainty framework have been developed to evaluate measurement uncertainty, with numeric computation being utilised in many of the current software tools.

Numeric computation, as the traditional computation method widely used, has a number of limitation when it is used alone. According to the previous work by the author (Wei et al, 2007), the traditional numeric computation posed the following limitations to the evaluation of measurement uncertainty:

Firstly, as defining the mathematical model for the measurand is one of the most critical steps of the evaluation process, it is highly desired that the software tools not only allow the use of symbols predefined by the tools, but also allows the user to enter other common symbols and define any other symbols that may be required in the evaluation process. Evidently, the latter is fundamentally limited by the use of pure numeric computation.

Secondly, in the intermediate calculation during the evaluation, the use of user-defined symbols is very limited, thus reducing the efficiency and user-friendliness of the tools.
Thirdly, the algorithms and rounding errors during the numeric computing can significantly affect the correctness of the final result. This is particularly obvious during the calculation of the sensitivity coefficients, which often involves highly complex partial derivatives.

Apart from these, using pure numeric computation also brings other limitations. For example, it requires repeating of the calculation if the value of any of the variables is changed. Also, using numeric computation makes is hardly feasible to see the influence of each variable to the final result.

Generally, numeric computation makes many of the current software tools less efficient and less user-friendly. Specifically, in many of the current software tools for uncertainty evaluation, the numeric computation significantly limits the capacity of the tools in expressing and processing the mathematical models for the measurand.

This section, therefore, proposes a methodology of integrating symbolic computation in the evaluation and expression of measurement uncertainty on the basis of GUM framework.

Symbolic computation aims at representing the processing information in the form of symbols, instead of using their numeric value, with the aid of machines (e.g. computers). Its origin dates back to 1960s when its original idea was initiated. The development of symbolic computation has been substantial in the past four decades and is now used in a number of industries including digital computation, simulation application and computer-aided design. Its utilisation in engineering industry has also seen a rapid development in the recent years.

As symbolic computation's main goal is to automate the mathematical calculation, it is able to provide a number of advantages, which overcome the main limitations of numeric computation.

Generally, as it directly processes the symbols instead of values, symbolic computation simplifies the calculation process, making it much more efficient.

With the symbols used in the calculation process, the logic and analytical relations among the symbols and their relation to the final result is easily seen in every step of the calculation.

When integrated into the evaluation of measurement uncertainty, symbolic computation brings a number of benefits, which makes the evaluation process more efficient and user-friendly.

First of all, using symbolic computation, the symbols can be pre-defined or user-defined, meaning the user has the freedom of defining the symbols whichever is needed or appropriate for the calculation process. The mathematical model can be $\frac{A}{B}$ or $\frac{X}{Y}$, with the symbols defined by the user to represent an input quantity. This provides significant flexibility to the users when they define the mathematical models for the measurand concerned. The symbolic representation can be easily and readily understood by both the user and the tools. This is also considered as an important booster to the user-friendliness.

Secondly, according to the GUM framework, the calculation of sensitivity coefficient is one of most critical steps of the evaluation process of measurement uncertainty. Sensitivity coefficients are partial derivatives of the input variables. The calculation is often complex, thus using symbolic computation which automates the generation of sensitivity coefficients enormously simplifies the process and improves its efficiency.

Last but not least, the utilisation of symbolic computation in the measurement uncertainty evaluation minimises the involvement of numeric values, especially in the intermediate steps. Practically, all the intermediate steps in the evaluation process can be carried out in symbolic forms and numeric values are only required in the final step. This therefore eliminates the possibility of calculation errors (e.g. rounding errors) in the intermediate steps.

To sum up, symbolic computation makes the evaluation process more user-friendly, accurate and efficient. In symbolic computation, the use of symbols, including user-defined symbols are maximised and the process is automated and simplified.

In order to realise the symbolic computation capacity of the evaluation process, a powerful symbolic computation engine is needed. In this section, the use of JSCL-Meditor (Java Symbolic Computing Library and Mathematical Editor) is proposed. JSCL-Meditor is an important computation engine written for Java, manipulating information in symbolic form. It is a free, open source software which can be utilised in the measurement uncertainty evaluation process to implement the symbolic computation.

JSCL-Meditor is proposed in this section due to its strong advantages, including portability among different computer platforms, readability and symbolic ability. Specifically, the symbolic capabilities of JSCL-Meditor now include polynomial system solving, vectors and matrices, factorization, derivatives, integrals (rational functions), Java code generation and graphical rendering of math expressions etc (Jolly, 2009). Using JSCL-Meditor, the mathematical expressions input by the user can be readily understood by the symbolic computation library model.

Regarding how JSCL-Meditor's symbolic capability is utilised in the evaluation process, this is discussed in details in the following chapter.

4.3.3 MCM Framework Knowledge Base

The Monte Carlo Method is the basis of the GUM SP1 Framework for the evaluation of measurement uncertainty. In this thesis, the MCM KB is developed based on NPL's software tool, NPLUnc (NPL Report MS1), which is suitable for both GUM and GUM SP1 Framework. The MCM Calculation approach of NPLUnc is integrated into the proposed system in this thesis and used to implement the MCM during the evaluation process.

As mentioned in Chapter 2, NPLUnc is Matlab-based and it includes the four example problems presented in GUM SP1. As a desktop application, it requires Matlab to be installed on the user's PC. It provides five probability density functions (PDFs).

In this thesis, an interface is established to connect Java and Matlab. By doing so, the user is able to visit and use the system via Internet, i.e. there is no need to install Matlab on the PC, as it is installed on the server side of the system. The detailed implementation is discussed in Chapter 5.

4.4 Development of the User Interface

In the proposed Web-based KB-DSS, the UI is realised using JSP, through which the user submits requests and obtains the responses. AJAX, which optimises the performance of the UI, plays a key role in this development process.

4.4.1 Java Server Pages (JSP)

Web applications have developed from static to dynamic ones. This change originated in the limitations of earlier web site design and the necessity to simplify application design. JSP, developed by Sun Microsystems, is a technology based on Java that allows software and Web developers to develop and maintain robust and dynamic Web pages (McPherson, 2002). This object-oriented scripting language is widely used not only in development of dynamic web sites but also in development of database driven web applications, due to its powerful server side scripting support.

JSP is one of the key components of Java 2 Platform, Enterprise Edition (J2EE), which is a collection of Application Programming Interfaces (API), a powerful and highly scalable platform for enterprise applications. Java Servlet technology serves as a basis for the JSP technology (McPherson, 2002). Although the latter is considered as a successor to the former, they can be and are often used in conjunction to offer the web developers an enhanced performance (e.g. ease of administration and maintenance, excellent extensibility etc).

Generally speaking, JSP provides web and software developers with an excellent platform and framework for building dynamic web pages containing HTML, DHTML, XHTML and XML (McPherson, 2002). A typical JSP page is plain HTML containing special JSP tags containing the Java source code, which is the source of the dynamic ability of the page. A JSP page is established on the support of a JSP engine that can be linked with a web server or reside within a web server. When a JSP page is sought after by a user, the JSP engine delivers the request to the JSP page, where the request is processed and generates a response for the client in a communicable way. This response is then delivered to the engine, which further delivers the response to the client (via HTML). These procedures depend on the fundamental layer of JSP, the servlet.

The JSP technology has been gaining breathtaking popularity in recent years, thanks to it strength in a number of ways. Outlined below are a few of its main advantages and benefits to the web and software developers:

- It is highly portable and platform independent, which means the JSP pages have excellent performance on any platform (including Linux) and run on/can be accessed from any web server. The famous feature of Java, ‘write once, run anywhere' (Sun Microsystems, 1999), is perfectly reflected by JSP. The cross-platform nature of JSP is particularly obvious in comparison with Active Server Pages (ASP) and Hypertext Preprocessor (PHP).

- The components of JSP pages are highly manipulable and reusable, which significantly decreases the development time. In JSP, the reuse of components are realised by using JavaBeans and Enterprise JavaBeans (EJB).
- It simplifies the development process, as it allows the Java source codes to be directly inserted into JSP file, which also significantly facilitates its maintenance.
- In JSP, the presentation, logic and data are separate (McPherson, 2002). In other words, the presentation layer and the implementation layers are completely separated from each other. Specifically, the HTML, visible to the user, is the presentation layer, while the JSP on the server is on the implementation layer. This, practically, means that the user interface and the content of the page are separate. The efficiency and ease of use of JSP is enhanced by this characteristics as the developers are allowed to modify the page layout without changing the dynamic content.

All these above-mentioned benefits make the JSP a powerful technology to rapidly develop dynamic web pages that are faster and easier to build, use, maintain and manage, cross-platform, highly extensible and with excellent performance.

4.4.2 Asynchronous JavaScript and XML (AJAX)

AJAX is a group of client-side technologies that are used in developing interactive web applications. The term AJAX was coined in 2005 by Jesse James Garrett (Garrett, 2005). Traditionally, when a request is sent to the server, the server processes the request and returns the response to the client via HTML. This undermines its efficiency as the user needs to wait every time when the server is processing the request. AJAX was proposed in a response to eliminate this disadvantage of traditional web application. Using AJAX, an additional intermediate layer is established between the user and the server. This fundamental difference between the traditional web application and AJAX is clearly shown in the Fig. 4.16 (Garrett, 2005).

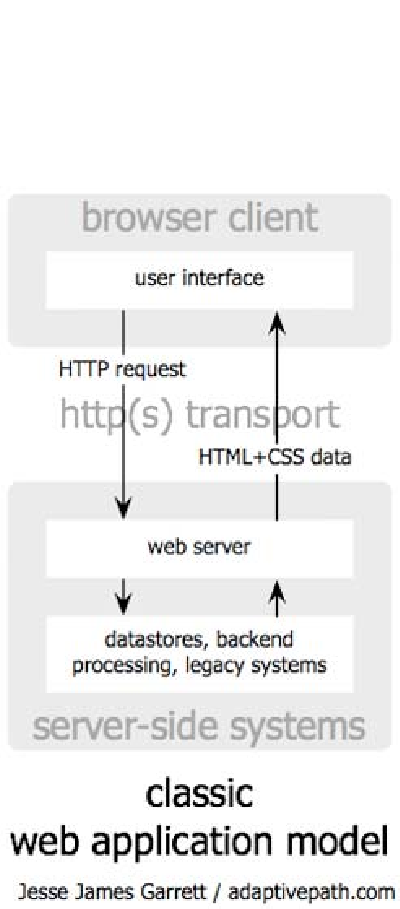


Fig. 4.16 Comparison of architecture of the traditional model for web applications (left) and an AJAX model (right). (Garrett, 2005)

As mentioned, AJAX is a group of pre-existing technologies. According to Garrett, typically, an AJAX application is a combination of the following:

- Presentation using Extensible Hypertext Markup Language (XHTML) and Cascading Style Sheets (CSS);
- The Document Object Model (DOM) which dynamically displays and interacts with the information presented;
- Asynchronous data exchange and retrieval between browser and server using XMLHttpRequest (XHR);
- data interchange and manipulation using XML and XSLT;

- and JavaScript, a client-side scripting languages.

In AJAX, the intermediate layer, the AJAX engine, communicates with both the user and the server. As indicated by its name, AJAX allows the communication between the user and the application to happen asynchronously unaffected by the communication between the engine and the server.

As can be seen in the figure, instead of a traditional HTTP request (used in traditional web application), a JavaScript call is sent to the server, which makes JavaScript an important component of an AJAX application.

AJAX is becoming increasingly extensively used in a large number of web applications in recent years, including Google Maps and Gmail. Its advantages and benefits are widely recognised in recent year, including the following:

- AJAX makes a web-based application much more responsive and faster, by using asynchronous requests. Page reload/refresh is therefore made unnecessary, consequently meaning that the sections of pages are reloaded individually. The elimination of this lag period, page reloading delays, is widely considered as one of the essential strengths of AJAX. In other words, this advantage significantly improves the user experience.
- An AJAX application is more secure compared with traditional web application as the web pages are not stored on a computer, which protects the application from attacks by virus.
- Traditionally, the information on a web page is reloaded upon receipt of every request. In contrast, using AJAX, only specific contents are requested as needed, consequently reducing bandwidth use and load time.
- AJAX allows separation between the content and the structure (i.e. data and display) of the page, making the web application to be much more efficient.

In short, AJAX is becoming a powerful approach that allows efficient designing and developing of more responsive, interactive and secures web applications, due to its excellent asynchronous mode.

JavaScript:

As mentioned above, JavaScript is an important component of an AJAX application. Originally developed by Brendan Eich at Netscape under a different name, JavaScript is a client-side scripting language particularly well-known for its use in web pages. The name JavaScript was introduced as a result of a co-marketing deal between Netscape and Sun, although it is important to note that JavaScript itself is essentially not associated with the Java programming language (Flanagan, 2006).

JavaScript is typically utilised as a component of the web browser, making it one of the most popular languages used in the development of dynamic websites, with its popularity further enhanced by the development of technologies such as AJAX. It has officially been approved by European Computer Manufacturers' Association (ECMA) as an ISO standard, with the standardised version named ECMAScript.

A number of characteristic of JavaScript contribute to its wide recognition and popularity (Flanagan, 2006):

- JavaScript is easy to learn and use, making it popular among developers not specialised in programming.
- JavaScript responds promptly to a user’s request, as it can run locally in the user’s browser, consequently improving its responsiveness and efficiency.
- It offers strong capability of working in the environment of HTML pages and interacts with the Document Object Model (DOM) (Eich, 1998).
- There is no need for specific tools to write JavaScript and a plain text or HTML editor is sufficient.

4.5 Development of Data Transfer Object

The proposed system in this thesis involves a large amount of data transfer, which makes the efficiency and security of the data transfer of significant importance to the performance of the system. In this thesis, Java Transfer Object is used to handle the data transfer and ensure its efficiency, stability and security. Highlighted below is a brief introduction to the concept of Transfer Object and it advantages.

In a software application, there are often needs to transfer multiple data elements across a tier or between the subsystems of the application. Traditionally this would be done via multiple remote requests/calls, which not only causes high network traffic, but also consumes unnecessary resources.

Transfer object (TO), formerly know as Value object, brought a solution to this type of problems. Transfer object brings a phase into the data storage and retrieval process, where multiple data elements are assembled, encapsulated and stored in TO and carried across a tier, often carried to the presentation tier. When the client makes request for data, instead of receiving multiple data elements, the client receives the TO. The data/individual attribute values transported by the TO are then accessed locally by the user, meaning remote calls are replaced by local calls.

As one of the design patterns introduced in the book, *Core J2EE Patterns: Best Practices and Design Strategies,* authored by experts from the Sun Java Center, TO resolves a number of problems associated with the traditional method of data transfer (Alur, 2003), e.g. the data being not serialisable and high network traffic. To be more specific, TO is a booster to the system's performance in a number of means:

- Applying TO significantly reduces the number of remote calls when multiple data need to be transported to the client, consequently decreases network overhead, network traffic and usage of the server resource as well as simplifies the remote interface.

- It allows more data to be transferred in fewer remote calls, increasing the overall efficiency of the system.
- A few strategies can be applied in conjunction with TO, further enhancing its performance. For instance, using one of the strategies allows the data in the TO to be modified and updated by the client, and then transported back from the client.

The integration and utilisation of TO in the proposed system is further discussed in Chapter 5.

4.6 Summary

In this chapter, the proposed methodology of this thesis is outlined with an explanation of the key technologies, theories and approaches involved. The proposed system, a Web-based KB-DSS, uses a MVC architecture, which is appropriate due to the web-based nature of the system. From the main components of the system, three knowledge bases are proposed to assist the user in mathematical model establishment and during the evaluation process of the measurement uncertainty.

The user interface of the system, which enables the interaction between the user and the system, is realised applying the JSP and AJAX technologies. Transfer Object is utilised in the proposed system to carry out date transfer, due to its strengths for systems requiring substantial amount of date transfer.

Chapter 5 System Design and Implementation

5.1 Introduction

In software engineering, system design encompasses a chain of processes, ranging from theory analysis, structure design, definition of system components and modules, data management to system optimisation and debugging. Typically, the design of a system involves a few common main stages:

1) Defining system architecture
2) Logical and physical design
3) Computer programming
4) Optimising the system
5) Testing the system

Depending upon specific cases, there may be additional steps required in the design of a system, such as analysis of the requirements of the system and its end users. It is also worth noting that the detailed design of the system includes both logical and physical design, the former is mainly concerned with the definitions of the system structure, input, output and procedures while the latter deals with the actual physical and process design.

In Chapter 4 (Fig. 4.1), the methodology of the proposed Web-based KB-DSS for the evaluation of measurement uncertainty has been discussed. The functional requirements of the system and subsystems have also been identified.

This chapter details the design of the proposed system and its implementation. As each sub-system has its own user interface, the discussion on each UI is included in the section for each sub-system. More specifically, this chapter discusses the following:

- Design and the implementation of the proposed System Framework and Data Transfer Object
- Design and the implementation of the Measurement Modelling Knowledge Base
- Design and the implementation of the GUM Framework Knowledge Base
- Design and the implementation of the MCM Framework Knowledge Base

5.2 Design and Implementation of the System Framework and Data Transfer Object

This section explains the design and implementation of the framework of the proposed system and its data transfer.

5.2.1 Design of the System Framework

As the proposed system applies MVC architecture, each subsystem/knowledge base is realised by a set of files listed in Table 5.1.

File Type	MVC Property	Function
*.JSP	View Component	Providing the User Interface.
*Action.java	Control/Action Component	Realising the action of the user on the UI, i.e. the web page, such as submission of data and page forwarding. *Mapping path needs to be set in the Struts-Config.xml*
*TO.java	Model Component	Transporting data among the components.
*Cmd.java	Business Logic Component	Corresponding to the commands of the programs and are the essential programs to realise the function

Table 5.1 MVC File Properties

Fig. 5.1 depicts an overview of the framework of the proposed Web-based KB-DSS:

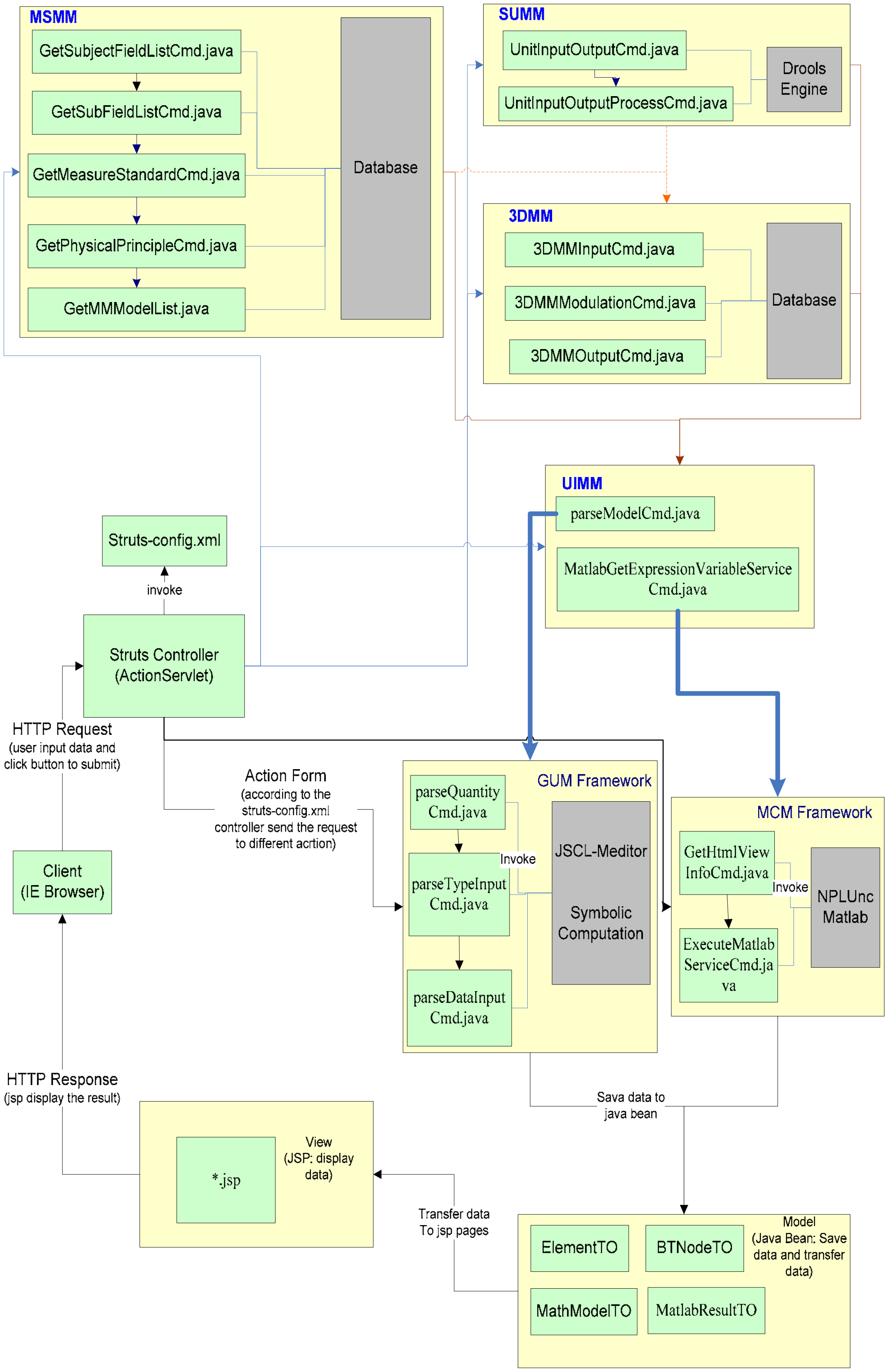


Fig. 5.1 Web-based KB DSS Framework

5.2.2 Data Transfer Object

Chapter 4 has discussed the concept of Java Transfer Object. The detailed implementation of TO in the proposed system is shown in Fig. 5.2.

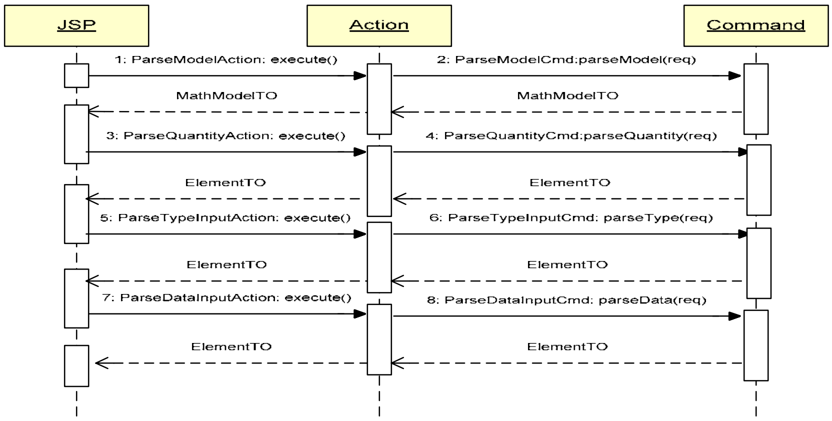


Fig. 5.2 Data Transfer Object Architecture

The user submits data via the JSP page to Action, which calls Command to carry out the processing of the data. Command stores the processed data in TO, which is returned to Action. Action then proceeds to the next JSP page based on the mapping path of struts-config.xml.

5.3 Design and Implementation of the Measurement Modelling Knowledge Base

As discussed in Chapter 4, the Measurement Modelling Knowledge Base includes four sub-modules: MSMM, SUMM, 3DMM and UIMM. Among these four sub-modules, the MSMM, SUMM and 3DMM can be considered as sub knowledge bases. MSMM and 3DMM can be categorised as database-driven knowledge base, whereas SUMM is a rule-driven knowledge base.

5.3.1 Design and Implementation of the Measurement Standards Modelling Module

In the MSMM, a large variety of measurement standards and mathematical models are stored and maintained. The function of this module is to help the user decide which mathematical model to use, given the specific measurement process. According to the logic relationship presented in Fig. 4.7, (section 4.3.1), a well-designed database is the critical stage of MSMM to ensure that the mathematical models can be easily and accurately retrieved by the system.

Database design typically involves a few the following steps:

- Determine the purpose of the database.
- Determine the tables in the database.
- Determine the fields in the tables.
- Identify fields with unique values.
- Determine the relationships between the different data elements
- Add data and create other database objects.

Database management system (DBMS) is a common approach applied in database design and development. Generally, DBMS is a software tool utilized to store data for further retrieval. Its main functions include creating, managing and using the tables in a database, as well as maintaining and updating them.

Currently, there are a number of DBMSs available in the market, among them Oracle is one of most widely used systems. Oracle Database, also commonly referred to as Oracle Relational Database Management System (RDBMS), is a cross-platform DBMS (Gavin, 2006).

Oracle allows a large amount of data to be stored and new systems to be easily and quickly developed from the pools of components. Oracle has both logical and physical structures, which are separate from each other, with the data logically stored in the form of tablespaces while physically stored as data files. This feature offers a benefit that the logical structures are not affected by the management of the physical storage of data. Oracle generally excels in senses that it offers high integrity, consistency and security of data and fast recovery from human errors and technical failures. It is also well-known for its user friendliness, versatility, capacity of handling large scale of database usage and consistently introducing new features to improve its performance (Gavin，2006).

For the above reasons, Oracle Database has been chosen to be used in the MSMM. Five tables are constructed in this database to manage *Measurement Subject Field, Measurement Sub Field, Measurement Standards, Physical Principle* and *Mathematical Model* respectively. The relationship among the five tables is shown in the following figure.

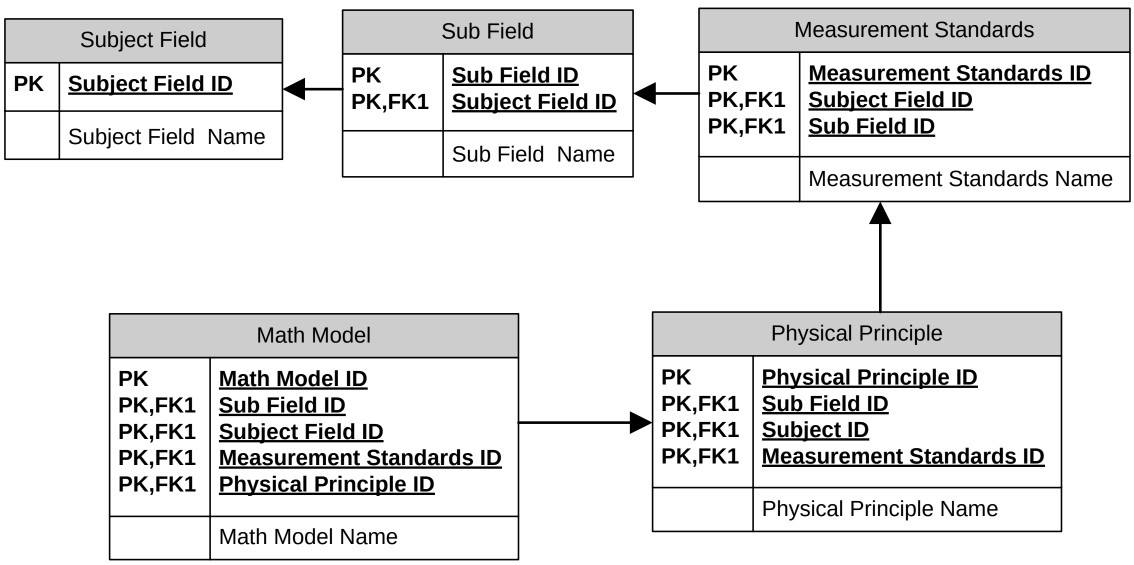
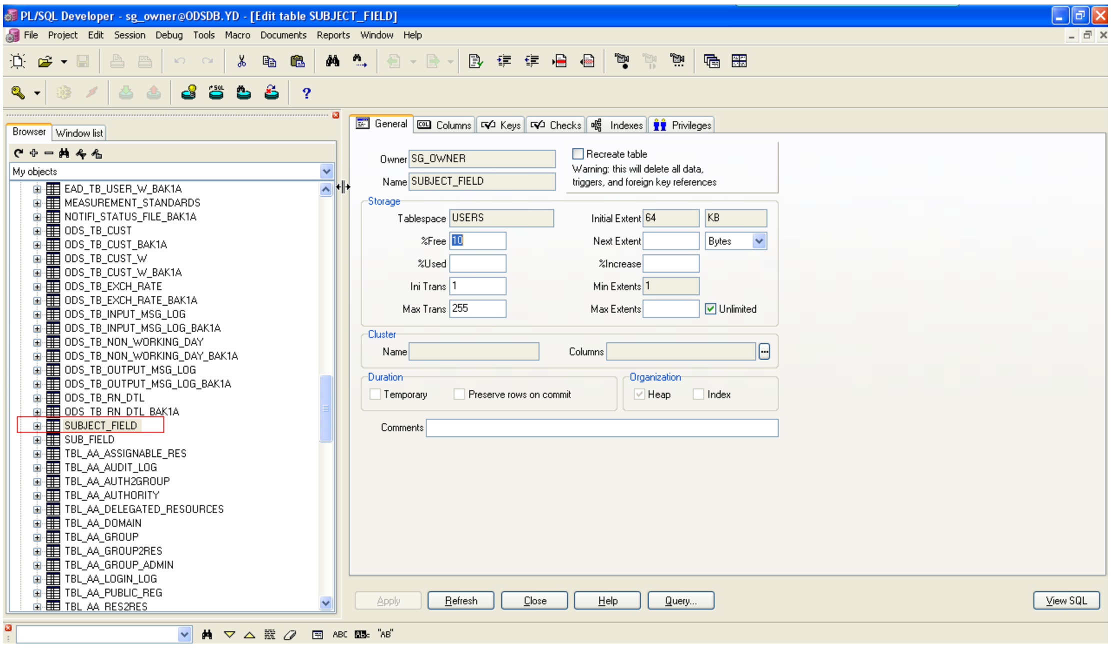


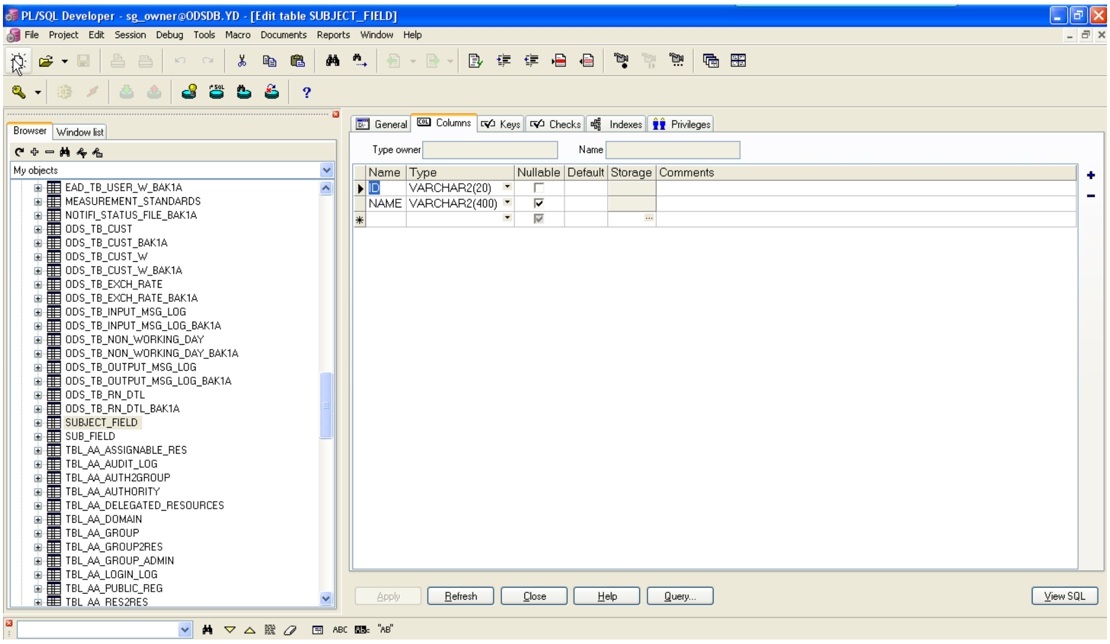
Fig. 5.3 MSMM Database Entity Relationship Diagram

The development of the database is conducted in the environment of PL/SQL Developer, a procedural extension language to SQL in the Oracle database. Taking the SUBJECT_FIELD TABLE for instance, the main procedures carried out are as follows:

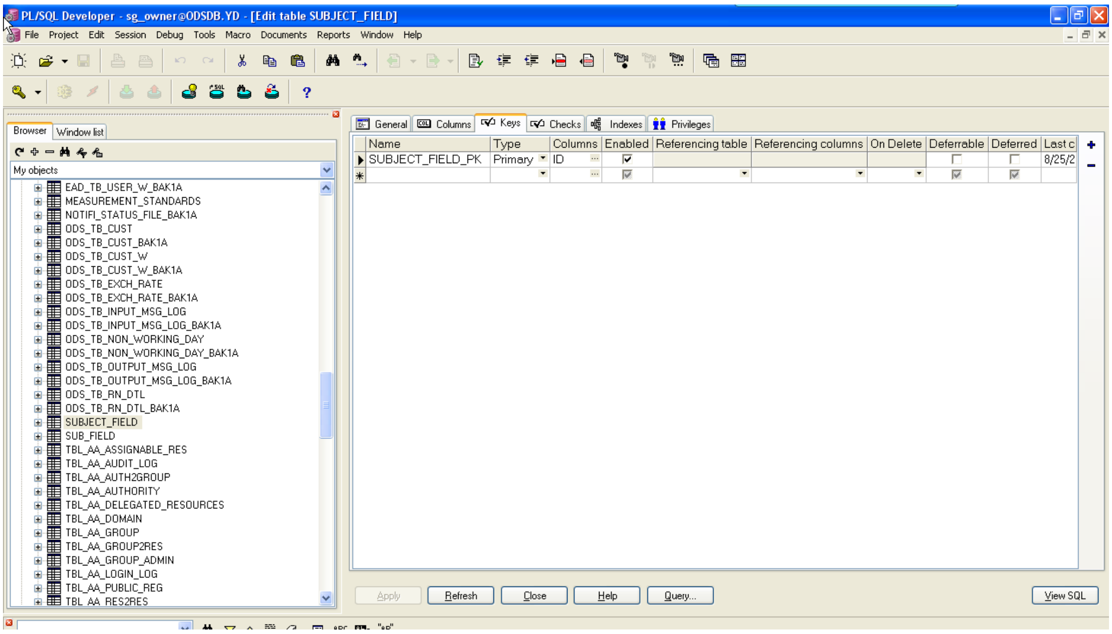
A) Create the name of the table and the general info (Fig.5.4):



B) Set the Column's property, name and type:



C) Set the Primary Key:



D) Create the Table

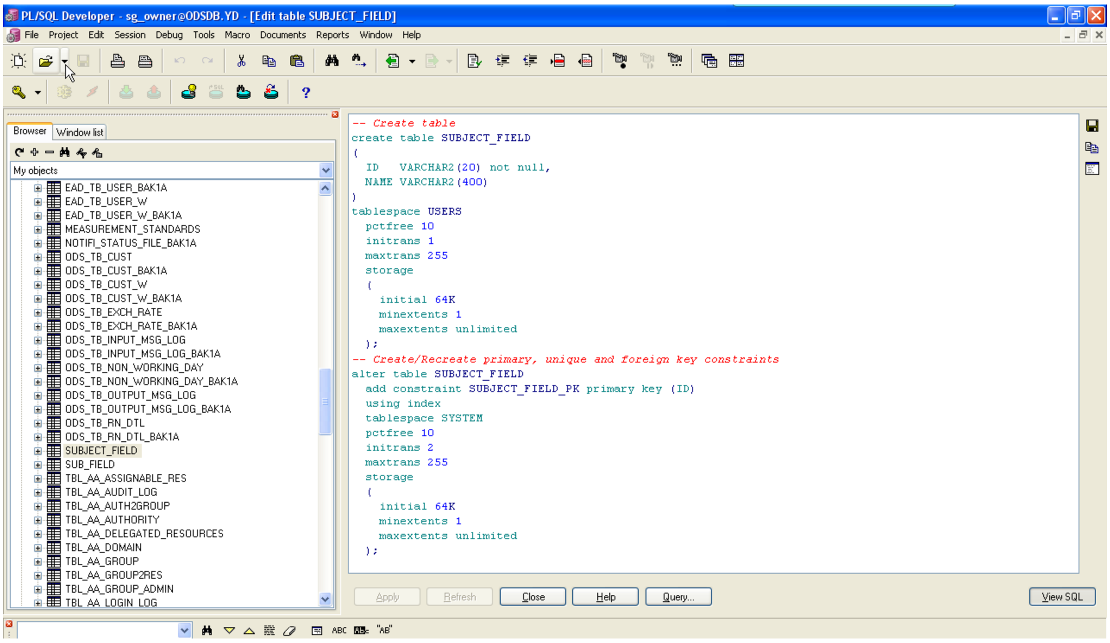


Fig. 5.4 Steps of Creating Database Table

Repeating the above procedures for each of the five tables generates the core database of the MSMM and the SQL codes.

The development of the database is only a part of the sub knowledge base, MSMM. Following the creation of the database, the integrated use of the MSMM database in the system is realised using Java programming language. Fig. 5.5 shows the general picture of the implementation of the MSMM.

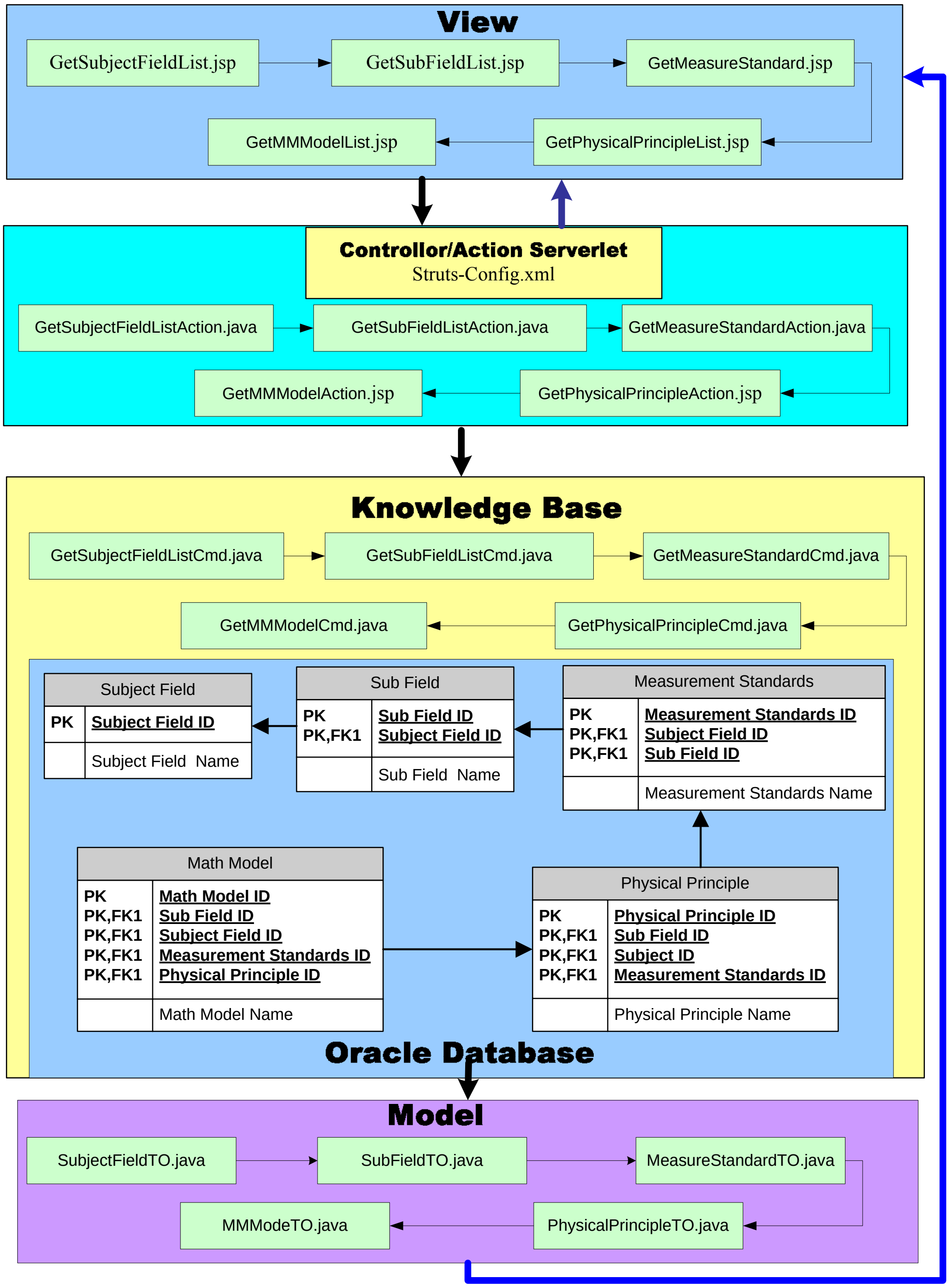
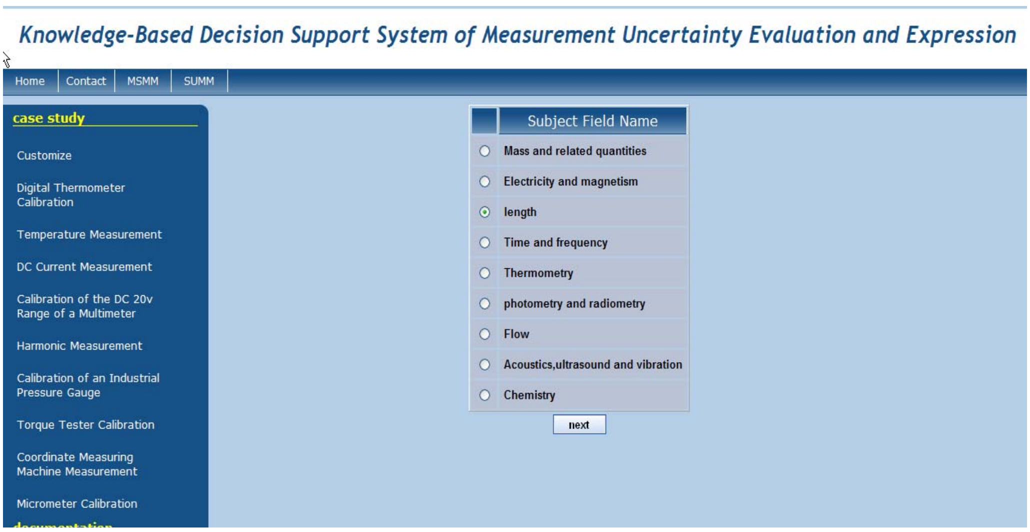


Fig. 5.5 Implementation of MSMM

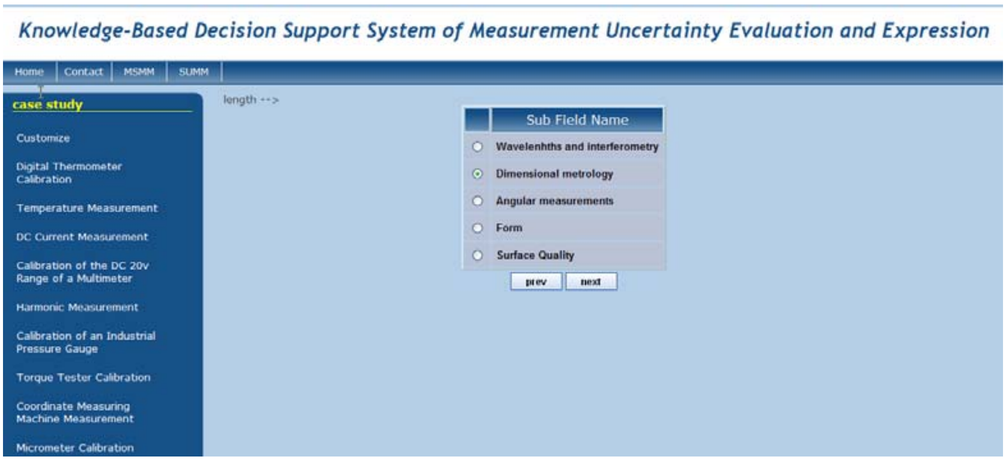
Apart from the *.JSP and *Action.java files, which have been introduced in the earlier section 5.2.1, *Cmd.java files correspond to the commands of the programs and are the essential programs of the database. Using GetSubFieldListCmd.java as an example, the program code is as follows:

```
public class GetSubFieldListCmd extends BaseCmd {
	private static final String GET_SUB_FIELD_LIST = "getSubFieldList";
	/*
	 * According to the parent id to get the child model list
	 * @param id - the primary key for get record
	 * @return List - the result of search
	 */
	public List getSubFieldList(String id){
		List list = null;
		try {
			if(getAppSQLTemplate() != null){
				list =
getAppSQLTemplate().queryForList(GET_SUB_FIELD_LIST,id);
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
		return list;
	}
```

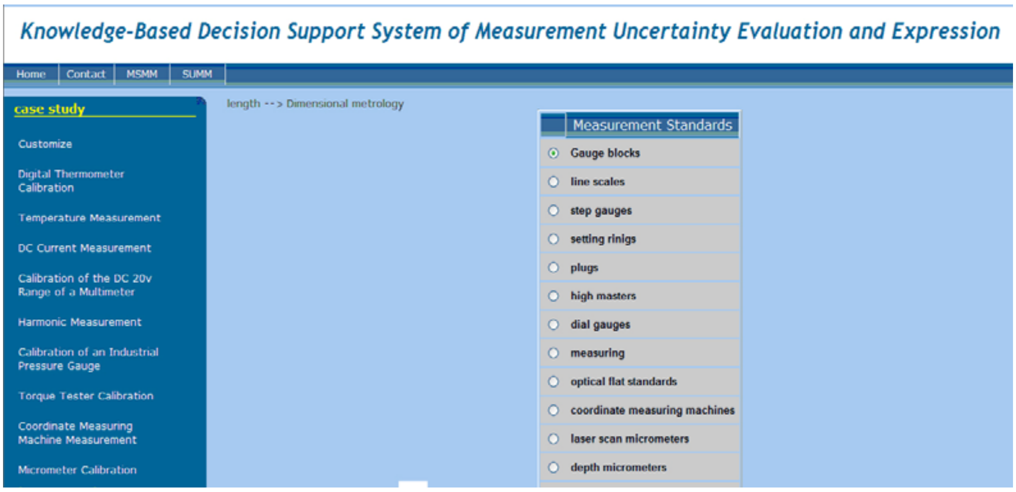
According to the above procedures, the user visits the knowledge base by choosing the right radio button and clicking the "Next" button on the web page. On completion of these procedures, the corresponding mathematical model is retrieved and transferred into the UIMM. Fig. 5.6 shows the user interface of MSMM.



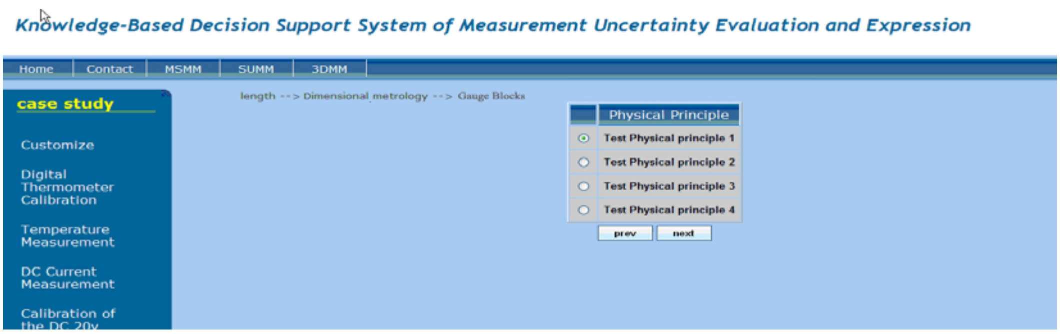
1)



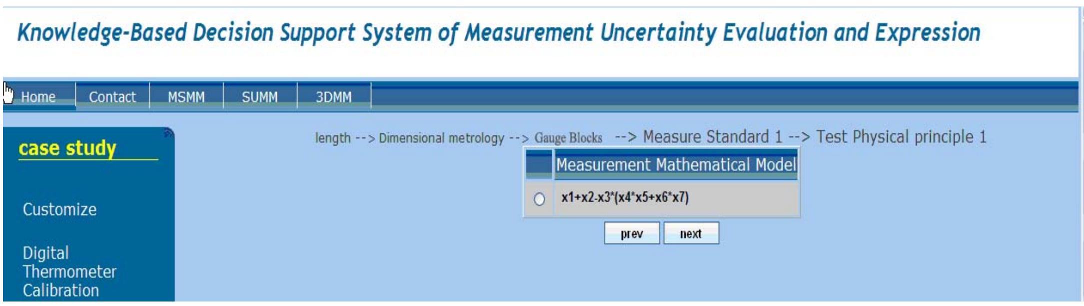
2)



3)



4)



5)

Fig. 5.6 User Interface of MSMM

5.3.2 Design and Implementation of the SI Unit Modelling Module

In Chapter 4, it is explained that the base units and derived units in the SI have certain derivation relationship. In SUMM, the mathematical models of the measurement can be derived based on the units of the input quantities and the output quantity provided by the user. This is a typical rule based knowledge base. The derivation of the measurement models has been performed using ES approach based on the Drools rule engine. As SI involves a large number of units and expressions, Ω as the output unit is used as an example in this section to demonstrate the principle of this module.

Drools

Drools is an open source rules engine to implement a rule based ES. It is a cross-platform business rule management system (BRMS) written in Java and is also known as a production rule system. Drools is widely recognised as a powerful framework for implementing complex business logic.

The heart of a production rule system is an inference engine, which matches the data with production rules to infer conclusions. This matching process is called Pattern Matching. Drools features a forward chaining inference rule. In Drools, a production rule is made up by two parts. Similar to the *if/then* statement structure of a typical ES, Drools utilises a structure of *when/then* statement.

when (condition)

then (action/conclusion)

Typically, a rule in Drools would look like this:

```
rule “name”
  ATTRIBUTES (optional)
  when
    (condition)
  then
    (action/conclusion)
end
```

A few algorithms can be used in a production rule system, such as linear, Rete, treat and leaps. Drools uses both Rete and leaps algorithms. Rete algorithm developed by Charles Forgy (1982) is an important basis of Drools as well as of other popular ES shells including CLIPS and Jess etc.

With the specific structure and technologies, Drools offers a number of benefits to its users, including the following:

- Drools offers excellent flexibility, e.g. it is able to associate the problem domain with domain specific languages.
- Drools allows the users to express the business logic in a declarative manner, making it easier to learn and use.
- It is capable of handling multiple languages, such as Java, Python and Groovy etc.
- Drools is significantly adaptive to changes.
- Being open source, Drools is available to all users free of charge. Also, by simplifying the components, Drools helps reduce the maintenance cost.

The design and implementation of SUMM Drools Expert System follows the procedures outlined below:

1. Analyse the relationship among the SI units using the concept of Tree Data Structure. Using Ω as the unit of the output quantity , it is a typical binary tree structure as shown in Fig. 4.12. The nodes with children (shown as circled nodes) are the Derived Units, whereas the nodes without children are the Base Units (shown as rectangular nodes). The solid line represents multiplication and broken line indicates division.
2. As shown in the figure, the binary tree is divided into seven layers. The possible mathematical models are analysed using binary tree traversal.

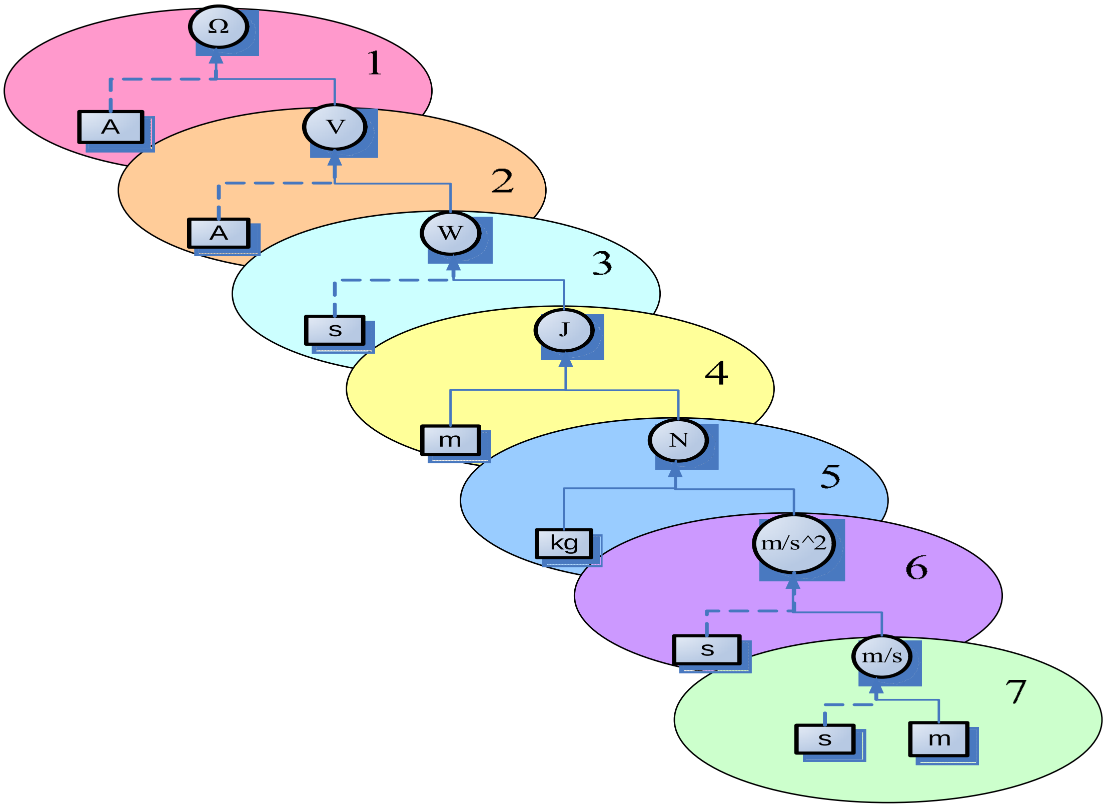


Fig. 5.7 7 Ties of Binary Tree Analysis for Ω

3. Table 5.2 shows the result of using traversal in this case:

Layer	Unit of Output Quantity	Units of Input Quantities	Mathematical Model
1	Ω	A, V	Ω=V/A
2	Ω	A, A, W	Ω=W/A/A
3	Ω	A, A, s, J	Ω=J/s/A/A
4	Ω	A, A, s, m, N	Ω= m*N/s/A/A
5	Ω	A, A, s, m, kg, m/s^2	Ω= m*kg*(m/s^2)/s/A/A
6	Ω	A, A, s, m, kg, s, m/s	Ω= m*kg*(m/s)/s/s/A/A
7	Ω	A, A, s, m, kg, s, s, m	Ω= m*kg*m/s/s/s/A/A

Table 5.2 Relationship of Input Quantities Units and Output Quantity Unit Ω in SI

When the information related to the input units is sufficient, the models in the last column, Expected Mathematical Model, can be reached. For instance, if the user provides three inputs units, A, s, and J, it results in the following model:

$$\Omega=J/s/A/A$$

If the user can only provide one unit of the input quantity, e.g. A, the first seven models in the table above satisfy the criteria. Also, there may be special cases, e.g. both the input unit and the output unit are Ω, in which case it is most likely that the model is an additive model. In these special cases, the user needs to base the decision on the specific situation.

4. 1) Based on the analysis logic above, the rule is created using ES approach. The critical principle of the algorithm is:
   - To put all mathematical models concerned into Drools working memory;
   - Once the user has chosen the unit of the output quantity (left side of the expression) and the unit of the input quantity (right side of the expression), the rule looks for mathematical model in the working memory which has the corresponding units.

   2) Specifically, the rule is created in Expert.drl file, named List. As the input and output units are chosen in one submission, the property of the rule is no loop.

```
rule "List"
    no-loop true
```

   3) In order to eliminate the redundancy of the rule, rule condition is used as a general condition for all units.

```
when
    btNodeTO: BTNodeTO( key == MathCommon.LOAD_LIST )
then
    System.out.println("Load formula list");
```

4) Vector object is created to store the mathematical models concerned. The Vector object is then stored in Drools working memory.

```
Vector vt = new Vector();
vt.add("O=V/A");
vt.add("O=W/A/A");
vt.add("O=J/s/A/A");
vt.add("O=m*N/s/A/A");
vt.add("O=m*kg*(m/s^2)/s/A/A");
vt.add("O=m*kg*(m/s)/s/s/A/A");
    vt.add("O=m*kg*m/s/s/s/A/A");
```

5. UnitInputOutputCmd.java and UnitInputOutputProcessCmd files are the critical part of the ES. This stage works as follows:

```
List formulaList = new ArrayList();
 try {
     String[] inputIdList = req.getParameterValues(MathCommon.INPUT_ID);
//obtain input units from the User Interface
     String outputId = (String)req.getParameter(MathCommon.OUTPUT_ID); // obtain
output unit from the User Interface
// BTNodeTO, encapsulate data into TO and put the TO into the Drools inference engine
     BTNodeTO btNodeTO = new BTNodeTO(MathCommon.LOAD_LIST);
//Obtain Drools Engine working memory
     WorkingMemory workingMemory = getAppWorkingMemory();
//Execute the rule
     workingMemory.fireAllRules();
// Obtain all lists
     Vector vt = btNodeTO.getFormulaList();
// Execute Traversal of vt
```

```
if(vt != null && vt.size() != 0){
    for(int a=0;a<vt.size();a++){
        String f = (String)vt.get(a);
// Using "=" to separate the output unit (left side) and the input expression (right side).
Execute Traversal of the formula list
        String []oFormula = f.split("=");
        if(oFormula[0].equalsIgnoreCase(outputId)){
            int counter = 0;
            for(int j=0;j<inputIdList.length;j++){
                if(oFormula[1].indexOf(inputIdList[j])>-1 ){
                    counter = counter + 1;
                }
            }
        //If the units match the units in the expression, put this expression into the
formula list
            if(counter == inputIdList.length){
                formulaList.add(f);
            }
```

Figure 5.8 below shows the overview of the implementation of the SUMM:

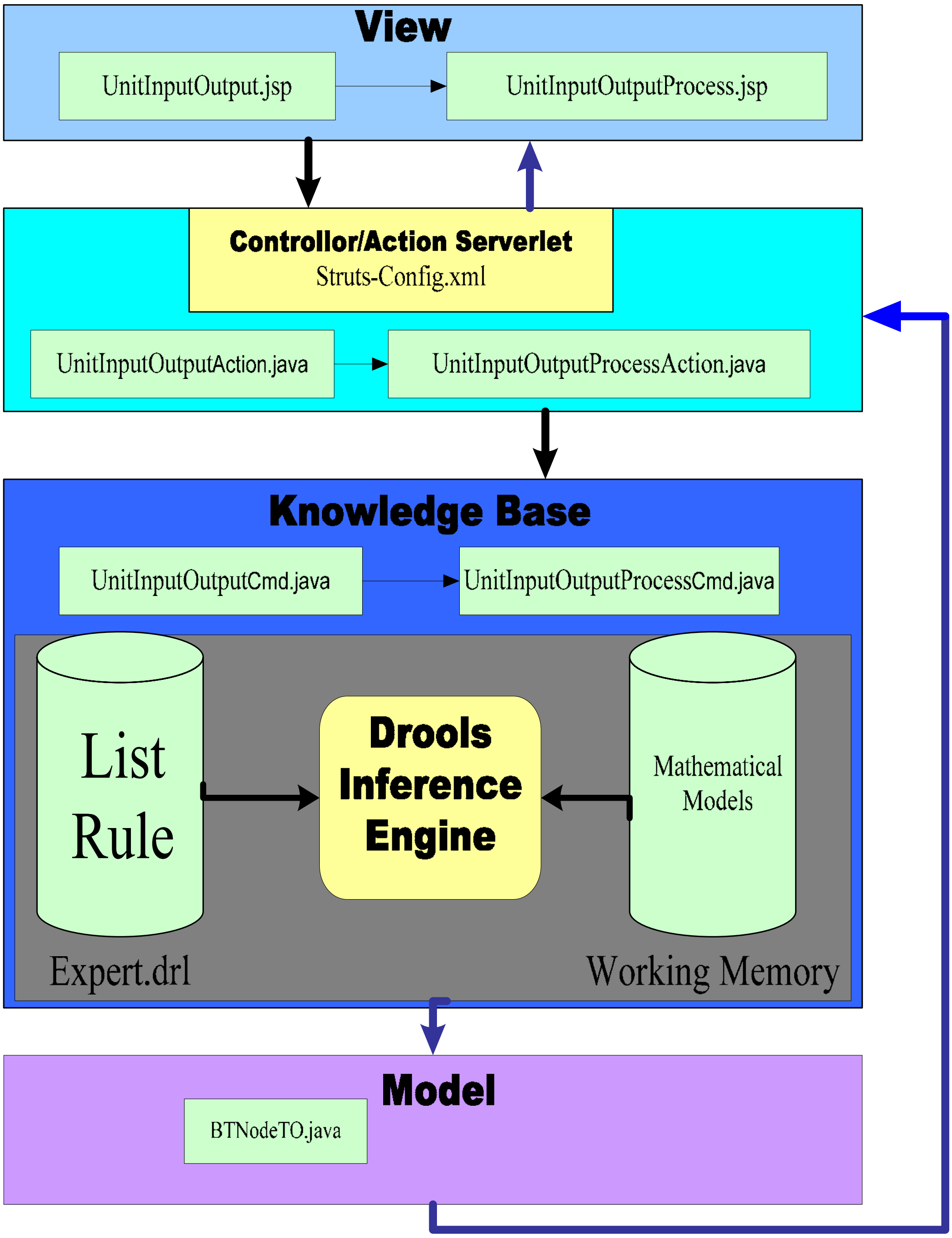
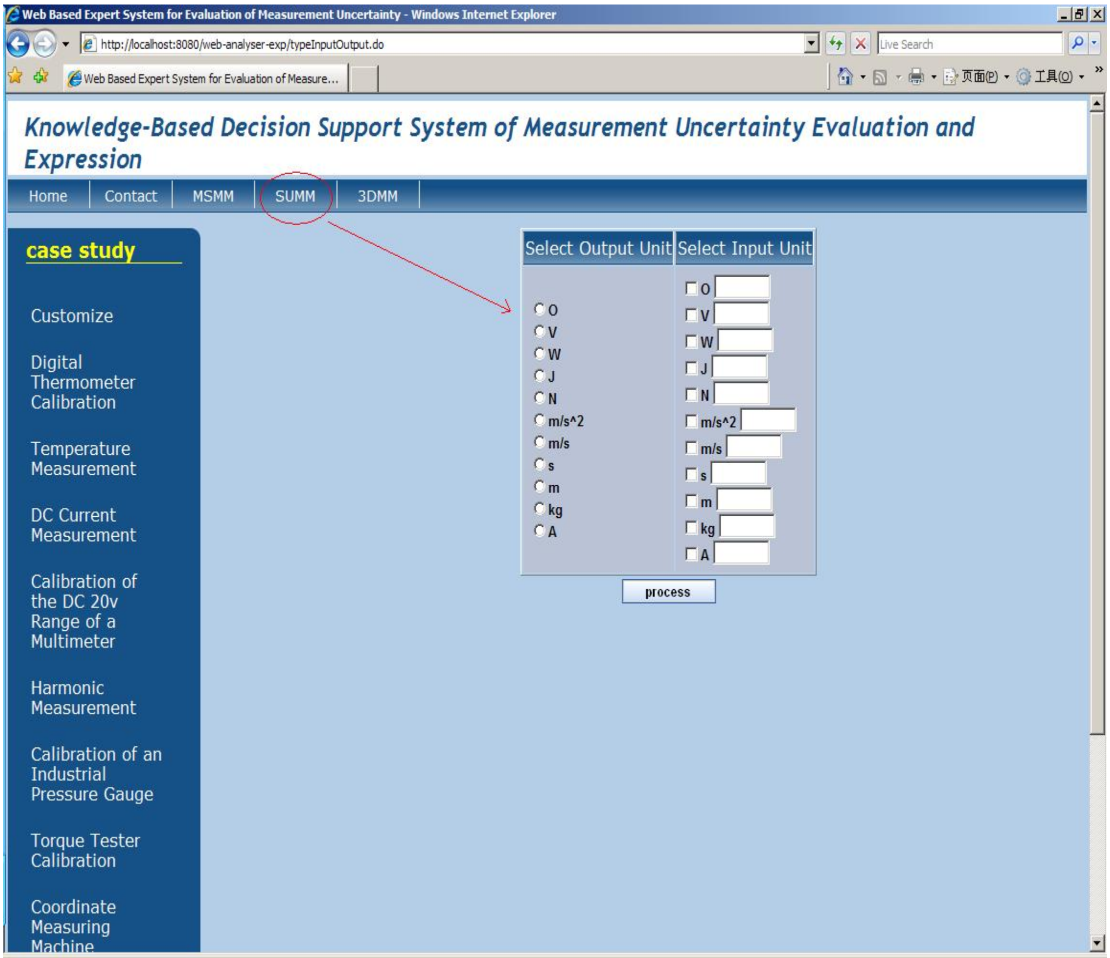
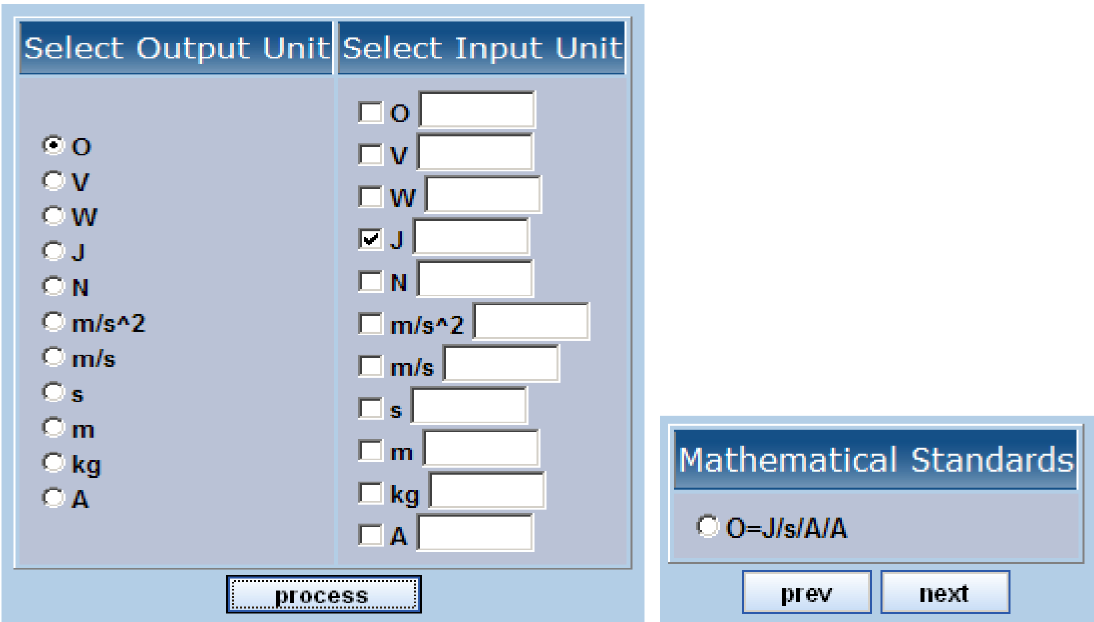


Fig. 5.8 Implementation of SUMM

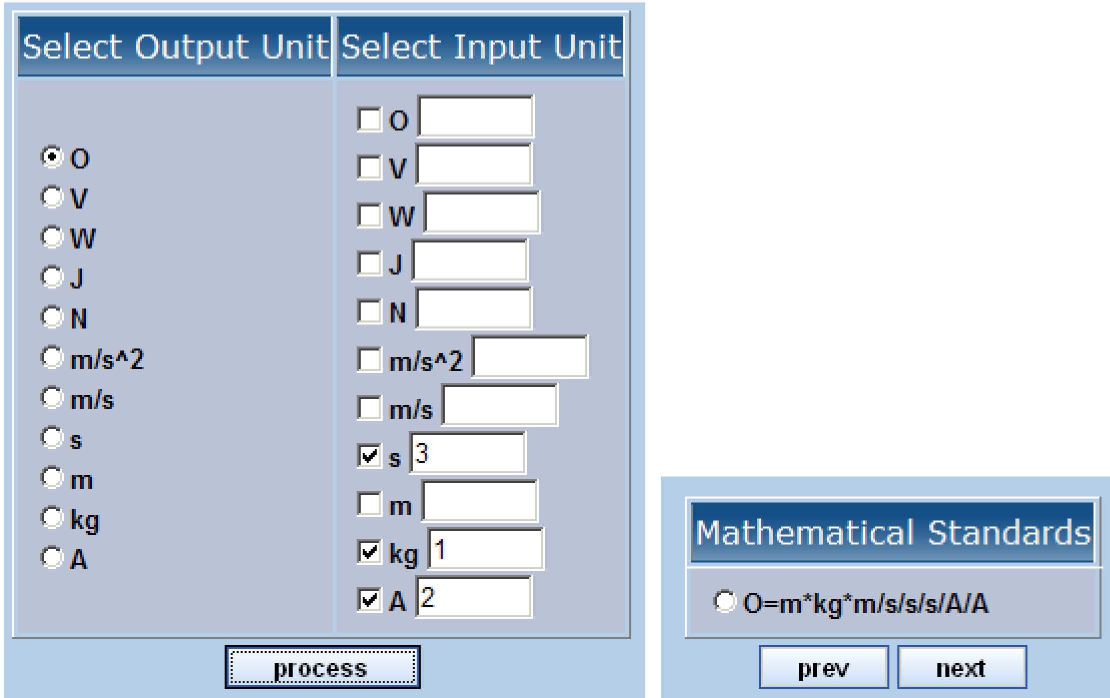
Following the procedures described above, the user chooses the radio button on the web page. By clicking the "Next" button, visit to the knowledge base is made feasible. Consequently the associated mathematical model is located and retrieved into the UIMM.


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
http://localhost:8080/web-analyser-exp/typeInputOutput.do
Knowledge-Based Decision Support System of Measurement Uncertainty Evaluation and Expression
Home
Contact
MSMM
SUMM
3DMM
case study
Customize
Digital Thermometer Calibration
Temperature Measurement
DC Current Measurement
Calibration of the DC 20v Range of a Multimeter
Harmonic Measurement
Calibration of an Industrial Pressure Gauge
Torque Tester Calibration
Coordinate Measuring Machine
Select Output Unit
Select Input Unit
O
V
W
J
N
m/s^2
m/s
s
m
kg
A
process

1)


Select Output Unit
Select Input Unit
O
V
W
J
N
m/s^2
m/s
s
m
kg
A
process
Mathematical Standards
O=J/s/A/A
prev
next

2)



3)

Fig. 5.9 User Interface of SUMM

5.3.3 Design and implementation of the Three Dimensional Modelling Module

The methodology of the 3DMM is discussed in Chapter 4 and the feasibility of this module is validated with examples. The implementation of the 3DMM will be very similar to the MSMM as described in section 5.3.1. Of course, complete implementation of the 3DMM will require significantly more efforts and time. For this reason it will be suggested as the future work.

5.3.4 Design and Implementation of the User Input Modelling Module

This module receives the mathematical models from MSMM, SUMM and 3DMM. The models are displayed on the web page to be modified by the user depending upon the specific measurement problem. Alternatively, the user can enter the mathematical model in this module directly. The design of this module is based on the AJAX technology. In accordance with the different requirements that the GUM Framework Knowledge Base and MCM Framework Knowledge Base have for the mathematical models, the models go through a pre-processing, the results from which are then transferred to the server, intended for the uncertainty evaluation carried out later. As AJAX is used, there is no need for page forwarding or refreshing. Once the user has finished inputting, the server starts the processing.

GUM KB uses symbolic computation. When the user chooses to use GUM Framework to conduct the further calculation, the symbols representing the input quantities, common letters and other symbols, including mathematical operation symbols such as +, -, *, /, ^2 and relational symbols such as (), [], {}, are parsed. The parsed variables are transferred to the GUM Framework.

The implementation of this is broken down into three steps:

1）All symbols other than the symbols for the input quantities are replaced by comma "," and all symbols for the input quantities are assigned;

2）A second assignment for the symbols which are not for the input quantities, e.g. brackets and calculation symbols etc, is carried out using stack algorithm;

3）The results from the two assignments are transferred to the GUM Framework.

As the MCM KB uses the computing engine of Matlab, when the user chooses to use MCM Framework for the further calculation, the mathematical models are transferred into a format which is readable in Matlab via the interface between Java and Matlab, instead of going through an assignment process for the symbols.

The implementation consists of the following steps:

1) The mathematical model entered by the user in the user interface is defined as "myExpression";

2) All variables in the model are obtained using MatlabGetExpressionVariableService.

3) Expression readable in Matlab is obtained using getExpressionVariable and getReplaceExpressionVariable. The expression is then transported into MCM Framework.

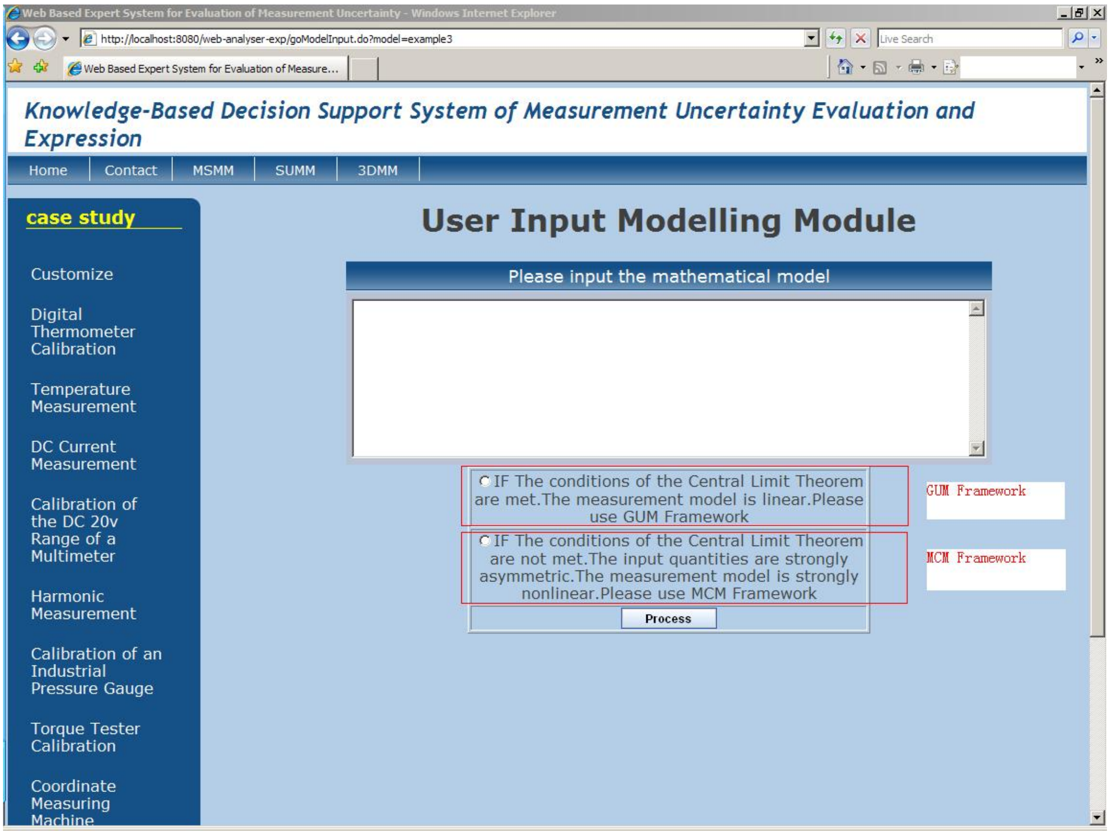


Fig. 5.10 User Interface of UIMM

5.4 Design and the Implementation of the GUM Framework Knowledge Base

The goal of the GUM KB is to evaluate measurement uncertainty by using the GUM Framework approach (Fig. 2.1) to process the data and information associated with the measurement that the user provides. This KB is a typical Java-based knowledge base. All processing logic, algorithms and procedures are programmed in Java and the calculation is conducted using symbolic computation.

The calculation process involves a large amount of data inputting, therefore the following points must be taken into consideration during the design:

1) User Friendliness

To achieve this, the web page must be succinct and straight forward; the page forwarding needs to be kept to a minimum level; the inputting process is better performed by using tables, buttons, radio buttons and dropdown lists. This not only improves the user friendliness but also facilitates the modification process in case there are mistakes during data inputting as page forwarding is kept to minimum.

2) Efficiency

All data elements obtained during the inputting and intermediate stages are encapsulated into TO.java. The Transfer Object containing these data elements is called in the final calculation. This is the underlying logic of Java Transfer Object.

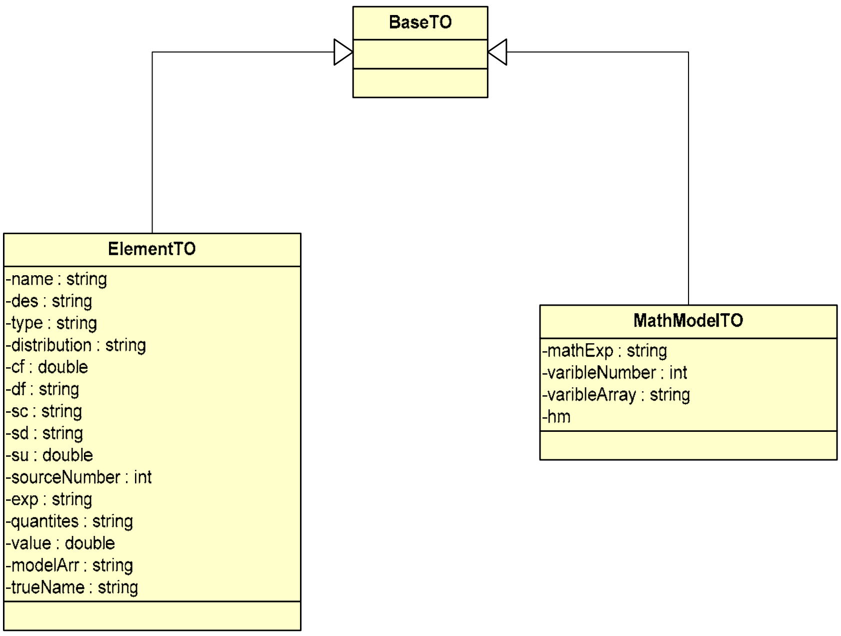


Fig. 5.11 GUM KB Transfer Object Relationship

As shown in Fig. 5.12, the GUM KB is based on the MVC.

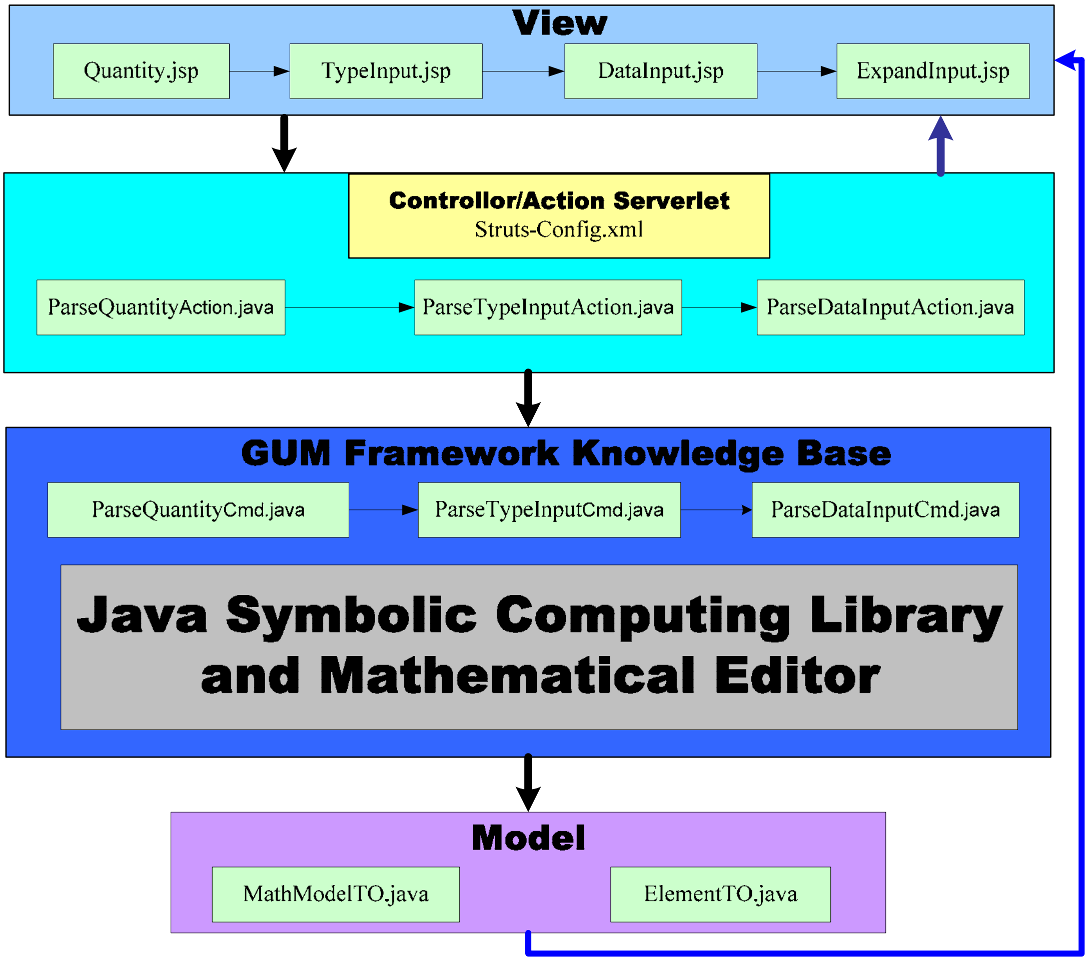


Fig. 5.12 Implementation of GUM KB

In general, the implementation of the GUM KB is carried out in the following four stages:

5.4.1 The pre-processing of Input Quantities

After the mathematical model is parsed in UIMM and transferred to GUM KB, the first step is to pre-process the Input Quantities and identify the number of the uncertainty sources for each input quantity and the value. The core files are:

- Quantity.jsp
- ParseQuantityAction.java

- ParseQuantityCmd.java

The implementation of the pre-processing is as follows:

1) Create a table each row of which corresponds to each of the input quantities of the mathematical model.
2) The first column is the input quantities obtained from the models parsed by UIMM.
3) The second column is the values of the input quantities provided by the user. Based on specific case, the value can be a group of measurement results or pre-specified values, for example, of the measurement instruments.
4) After the user has input the values in the second column, the third column automatically produces the arithmetic mean value. The codes are as follows:

```
function averge(obj){
        arr = obj.value.split(",");
            for (i=0;i<arr.length;i++) {
                if (isNaN(parseFloat(arr[i]))) {
                    alert("invalid input, please input number");
                    obj.value="";
                    document.fm[avrg].value = "";
                    return false;
                }else{
                    sum = sum + parseFloat(arr[i]);
                }
            }
            sum = sum/arr.length;
            }
```

5) In reality, for each unit of the input quantities, different decimal submultiples and multiples of the same unit may be used by the user. For example, the user

may have provided values in both cm and m. It is required that the same decimal multiple of one unit must be used for further calculation. Therefore, the fourth column allows the user to adjust the values by converting them into the same decimal multiple.

6) According to NPL's Measurement Good Practice Guide No. 11, each input quantity has one or more than one uncertainty sources. The common uncertainty sources are listed in the fifth column where the user can select from the sources. The number of the sources is then calculated automatically.
7) Upon completion of the above steps, the values of the input quantities provided by the user and the arithmetic mean value are encapsulated into corresponding MathModelTO.java files and ElementsTO.java files. The codes are as follows:

```
MathModelTO mmt = new MathModelTO();
mmt.setMathExp(model);
mmt.setVaribleArray(arr);
mmt.setVaribleNumber(ar.length);
mmt.setHm(hm);
req.getSession().setAttribute(MathCommon.MATH_MODEL_TO, mmt);
```

8) If the user is sure that the input is correct, the next step can be accessed by clicking the "next" button. If not, modification can be made by returning to the previous step by clicking the "prev" button. If the input information is not complete or there is input error, the system will generate error messages.

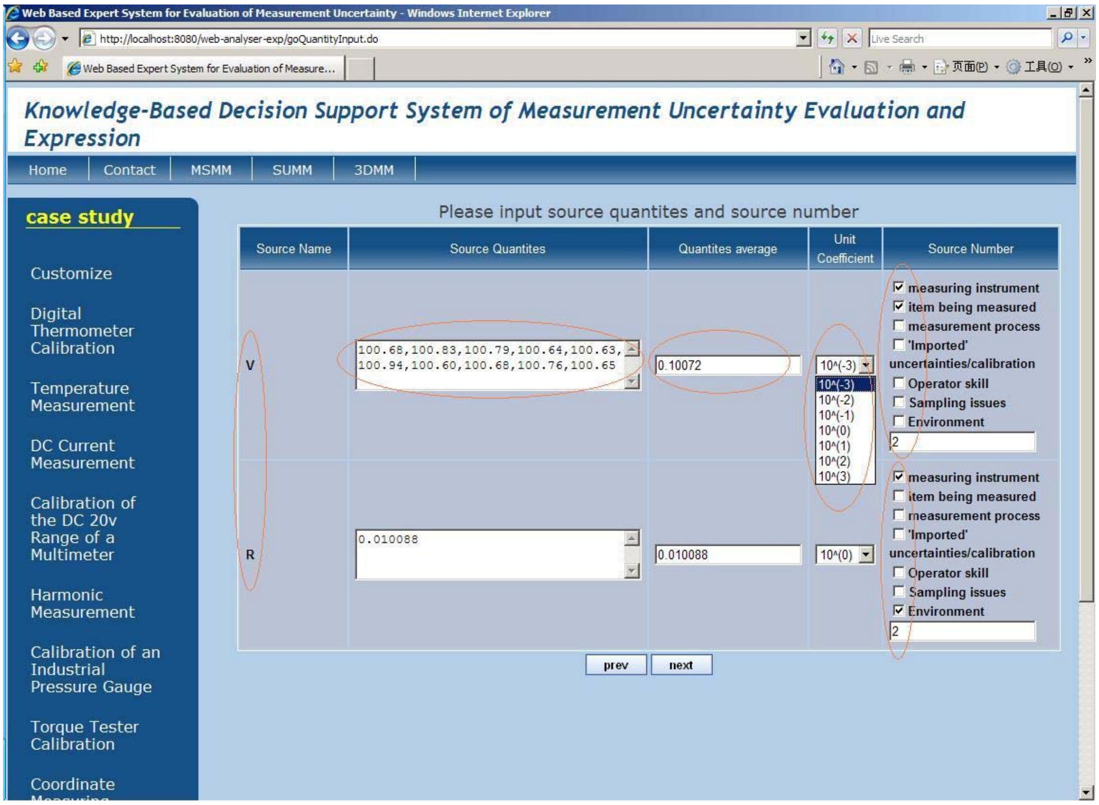


Fig. 5.13 Uncertainty Sources from Input Quantity

In this step, the support to the user is provided by automatically generating the arithmetic mean value, adjusting the decimal multiple of the input unit, providing information regarding the uncertainty sources and automatically calculating the number of sources.

5.4.2 Types and Degrees of Freedom of Uncertainty Sources

Once the value and number of uncertainty sources have been identified for each input quantity, each source is processed based on the appropriate measurement type by identifying the type and Degrees of Freedom (DOF) for each uncertainty source of each input quantity. The core files are:

- TypeInput.jsp
- ParseTypeInputAction.java,
- ParseTypeInputCmd.java,

The implementation of this step is carried out as follows:

1) Create a table each row of which corresponds to each uncertainty source identified in the first step.
2) To avoid confusion caused by the indication of the symbols, the second column allows the user to include description of each uncertainty source.
3) The third column determines the type of each uncertainty source.
4) After the type is determined, the DOF will be generated automatically by the system. If it is Type A, the DOF is *n*-1 (*n* is the number of the samples); If it is Type B, the DOF is infinity. The codes are as follows:

```
public String getDof(String quantity){
String arr[] = quantity.split(",");
      if(0==arr.length-1){
       return String.valueOf(1);
 }else{
       return String.valueOf(arr.length-1);
 }
}
```

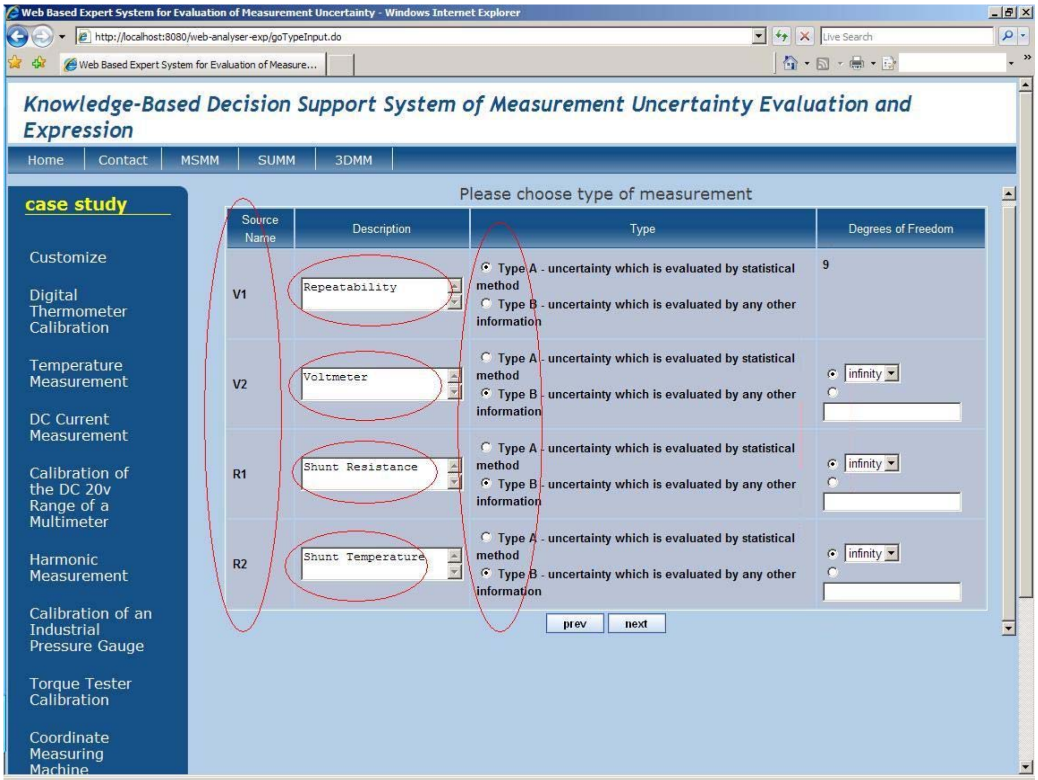


Fig. 5.14 Type and Degrees of Freedom of each Uncertainty Source

5) All above-mentioned steps are entered by the user. Modification is possible by returning to the last step by clicking the "prev" button.

6) Once the user clicks the "next" button, the programs on the server will carry out the calculation for the DOF and sensitivity coefficient of each uncertainty source. The core logic of this is to use API of the JSCL-Meditor to calculate the derivative of the mathematical model. The results from the calculation of the sensitivity coefficient are stored in TO and displayed on the web page in the next step. This is the most critical part of the proposed approach of GUM Framework using symbolic computation. Below is an example of a complex mathematical model:

$$T_m = M(1 - \frac{\rho_a}{\rho_m}) \cdot g \cdot d_{20} \cdot (1 + \alpha \cdot \theta)$$

In this case, the user can input a simplified model in the UIMM (the symbols can be replaced as needed as long as the repeated use of the same symbol is avoided), m*(1-A/B)*g*d*(1+X*Y). This provides significant flexibility. The variables from the model, m, A, B, g, d, X, and Y, are put into TO. ParseTypeInputCmd.java then calls JSCL-Meditor to calculate the derivative of each variable.

```
    // get the expression from a String. Here is “d(m*(1-A/B)*g*d*(1+X*Y), B)” which
means it computes derivatives.
    String str = "d(m*(1-A/B)*g*d*(1+X*Y), B)";
try {
//then put the expression value into the Generic object’s an instance, here is ‘ge’
Generic ge = Expression.valueOf(str);
//then JSCL-Meditor call its inner functions to compute the expression. The main function is
‘expand()’
Generic rt = ge.expand();
//The result may not be the simplest. So call the function ‘simplify()’ to simple it.
rt = rt.simplify();
//then output the result on the console
System.out.println(rt.toString());
} catch (Exception e) {
e.printStackTrace();
}
```

The output is m*（1-A/(B^2)）*d*(1+X*Y)*g, which is the sensitivity coefficient of input quantity B.

5.4.3 Data Input of Uncertainty Sources

The user then chooses the distribution of each uncertainty source in the next table to determine the Coverage Factor. For input quantities of Type B, the user also needs to input the Quoted Uncertainty. The corresponding files are:

- TypeInput.jsp
- ParseTypeInputAction.java,
- ParseTypeInputCmd.java,

The implementation of this step is conducted based on the following(see Fig.5.15)

1) Choose the distribution. The graphs of the distributions are provided on the page for the user to choose the appropriate one. The description of each distribution appears when the computer mouse is placed on the graph.

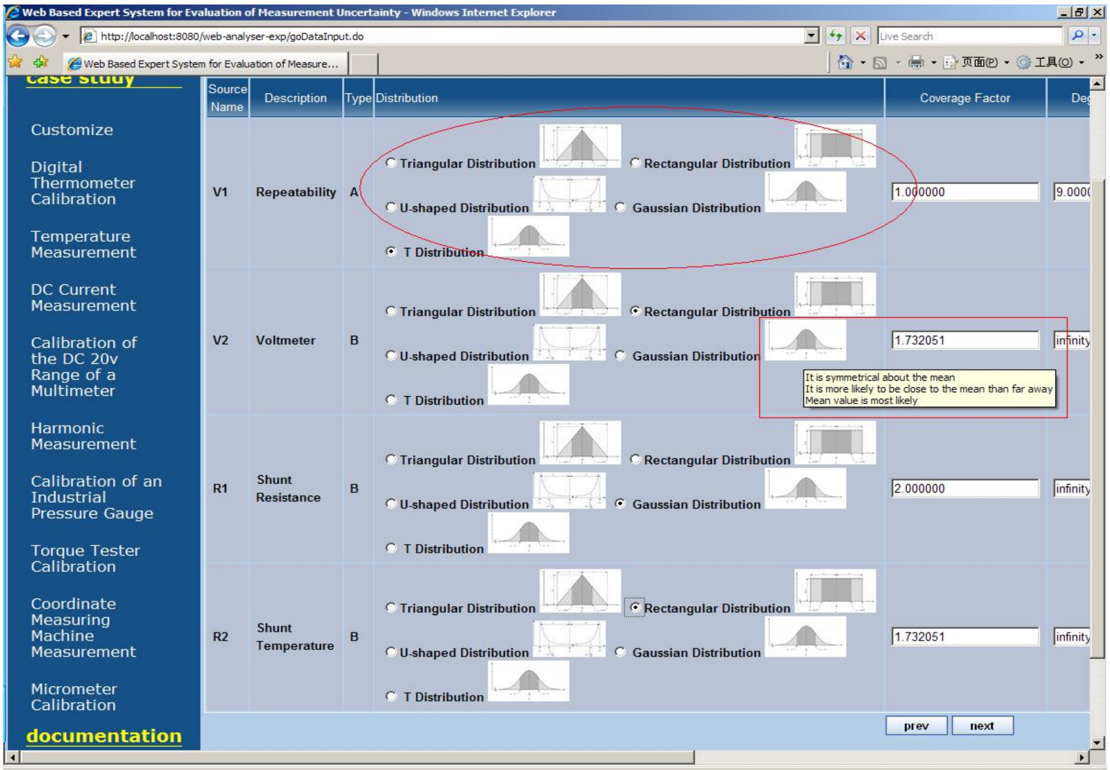
2) The Standard Deviation of sources of Type A is calculated automatically by the system. After the distribution is chosen by the user, the Standard Uncertainty will be displayed on the page immediately. As AJAX is used, there is no need for page forwarding or refreshing.

3) With regard to sources of Type B, the user needs to input the quoted uncertainty, after which the corresponding Standard Uncertainty will also be calculated and displayed on the page.

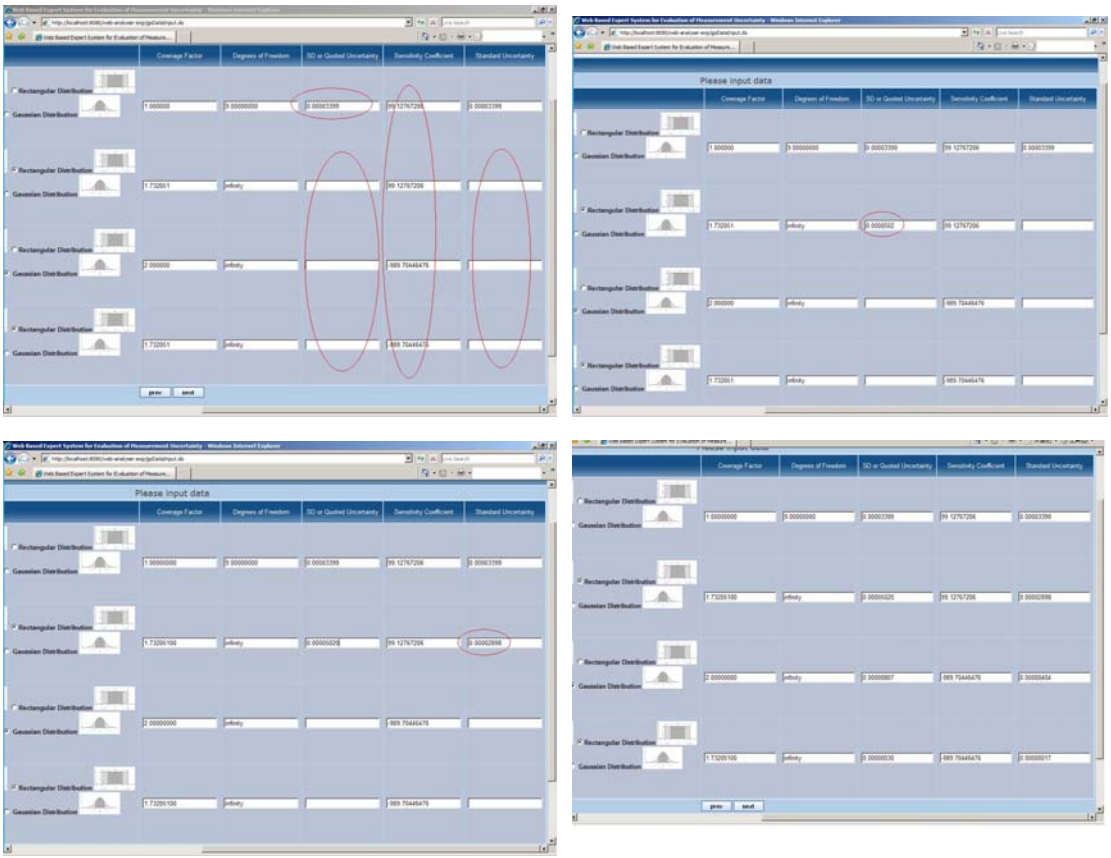
4) The user then clicks the "next" button. The TO of the previous steps are all called for calculation of combined uncertainty on the server.

The codes are as follows:

```
//compute the combined standard uncertainty
comSu = comSu +
    ((((ElementTO)vt.get(i)).getSu())*(((ElementTO)vt.get(i)).getSu()))*((Double.parseDouble(((ElementTO)vt.get(i)).getSc()))*(Double.parseDouble(((ElementTO)vt.get(i)).getSc()))));
}
comSu = Math.sqrt(comSu);
req.getSession().setAttribute(MathCommon.COMSU, String.valueOf(comSu));
```



1) Select Distribution of each Uncertainty Source



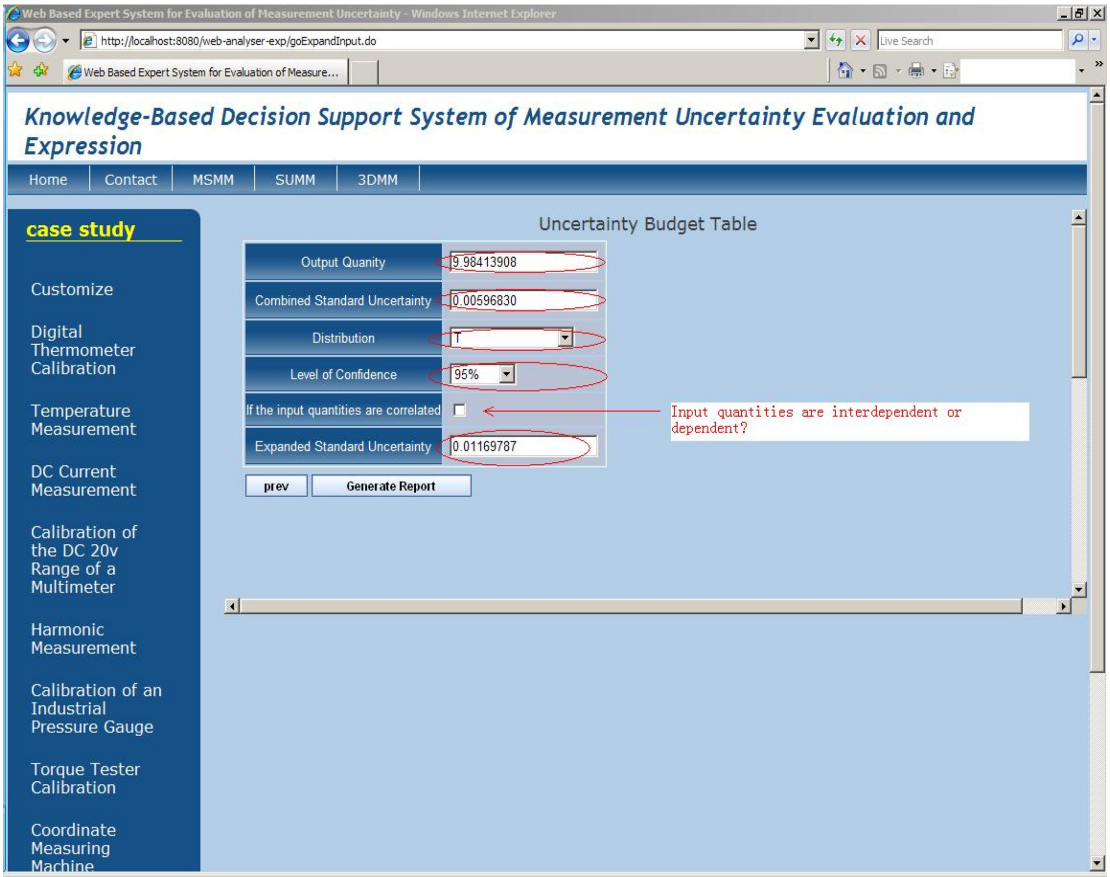
2) Input the Quoted Uncertainty of Type B sources

Fig. 5.15 GUM Uncertainty Data Input

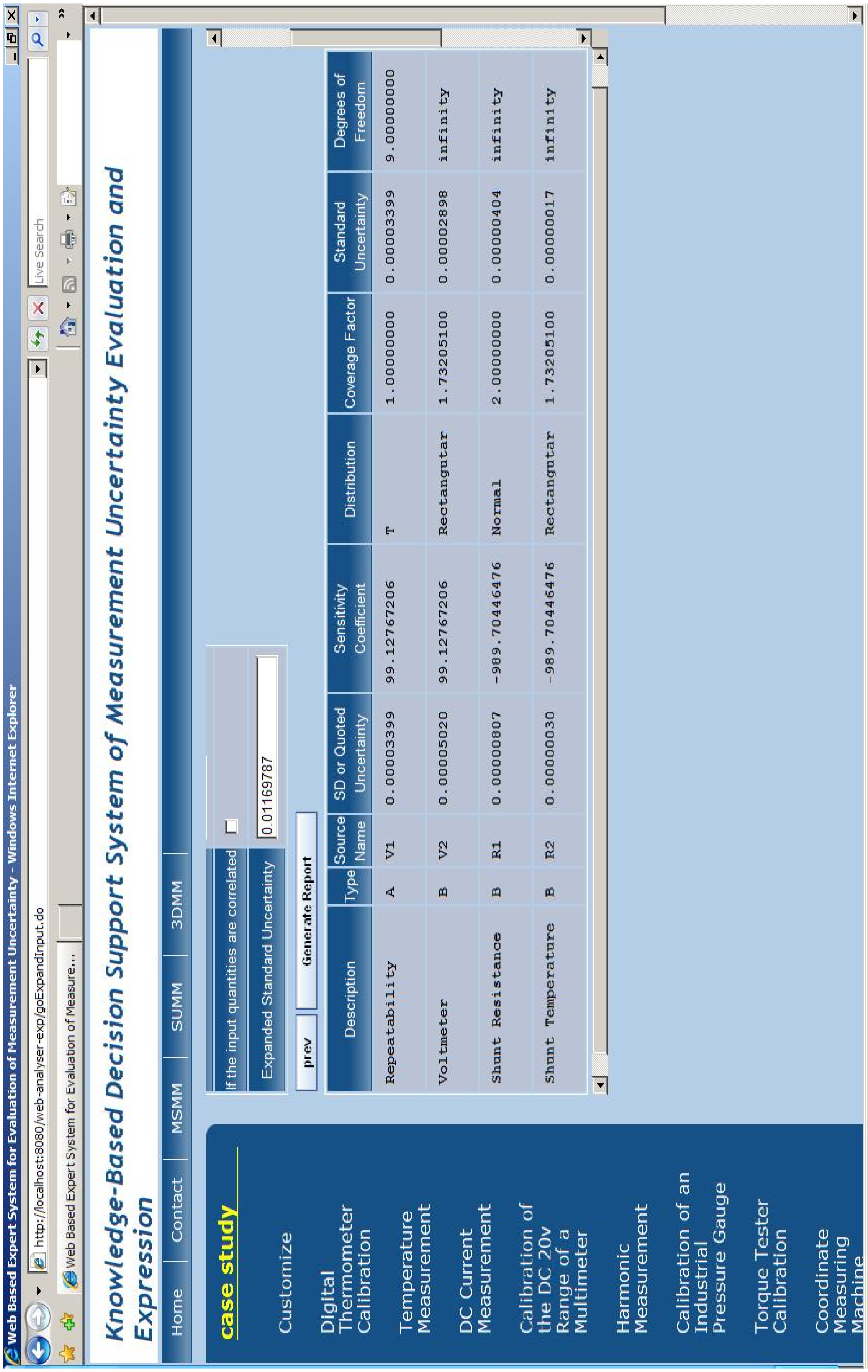
5.4.4 Expanded Uncertainty and Uncertainty Budget Table

Through the previous steps, the value of the output quantity and the combined uncertainty have already been calculated. This is the last step of calculating the uncertainty based on GUM Framework. This step only involves the user interface, i.e. ExpandInput.jsp.

In this step, the user needs to determine the distribution of the output quantity and the level of confidence, in order to calculate the Expanded Uncertainty. In the mean time, the system assumes that all input quantities are uncorrelated with each other. If they are correlated, the user can modify this in this step. If the user chooses that they are correlated, the system will calculate the combined uncertainty again on the server and return it to the page. Lastly, the user clicks the “Generate Uncertainty Budget” button and obtains the Uncertainty Budget Table as shown in Fig.5.16.



1)



2)

Fig. 5.16 GUM Uncertainty Budget Table

5.5 Design and the implementation of the MCM Framework Knowledge Base

MCM Framework Knowledge Base is a Matlab-driven knowledge base. Its function is to realise the evaluation of the measurement uncertainty based on GUM SP1 MCM Framework. NPLUnc (NPL Report MS1)'s MCM calculation program based on Matlab is used in this knowledge base. After the user has chosen or entered the mathematical model, UIMM transforms the model into a format readable in Matlab. In the mean time, the browser sends the information entered by the user on the web page to the web server. Upon the receipt of the information, the web server transfers the information to the programming interface of Matlab. All calculations are done in Matlab, which sends the result via the interface to the web server, which in turn transports the result to the web browser, enabling the user to view the result via the browser.

As mentioned in Chapter 2, modifications are made to NPLUnc's MCM calculation programming codes (see Appendix III). The modifications are outlined below:

1) Changed yr = eval([model, '(xr)']) to yr=eval (myExpression). As stated earlier, the "myExpression" here transforms the mathematical model from MMKB into a format readable in Matlab. This significantly broadens the range of models that can be processed in the system, no longer restricted to the four example problems of GUM SP1.
2) Based on GUM SP1 6.4.10 and 6.4.11, more distributions are added. The codes are as follows:

```
case {6}     %exponential distributions
               x=pdf{k,2};
               xr(k,:)=-(x*log(rand(1,M)));
case {7}     %gamma distributions
               q={k,2};
               p=rand(1,M);
```

```
x=p(1);
for i=2:(q+1)
x=x*p(i);
end
xr(k,:)=-(log(x));
```

3)The modified *.m file is transformed to a *.Jar file using JABuilder of Matlab and put in Java library of the system.

4) Create GetHtmlViewInfo.java files, collect the information submitted by the user via the JSP page and transform it into a format which is readable by Matlab. This is then transferred into NPLUnc via the interface between Java and Matlab.

5) Create ExecuteMatlabService.java files. When the user clicks the "Execute" button on the page, the data will be computed in NPLUnc. The result of the computing is then returned to the user on the next page.

The User Interface of MCM KB is shown in Fig. 5.17:

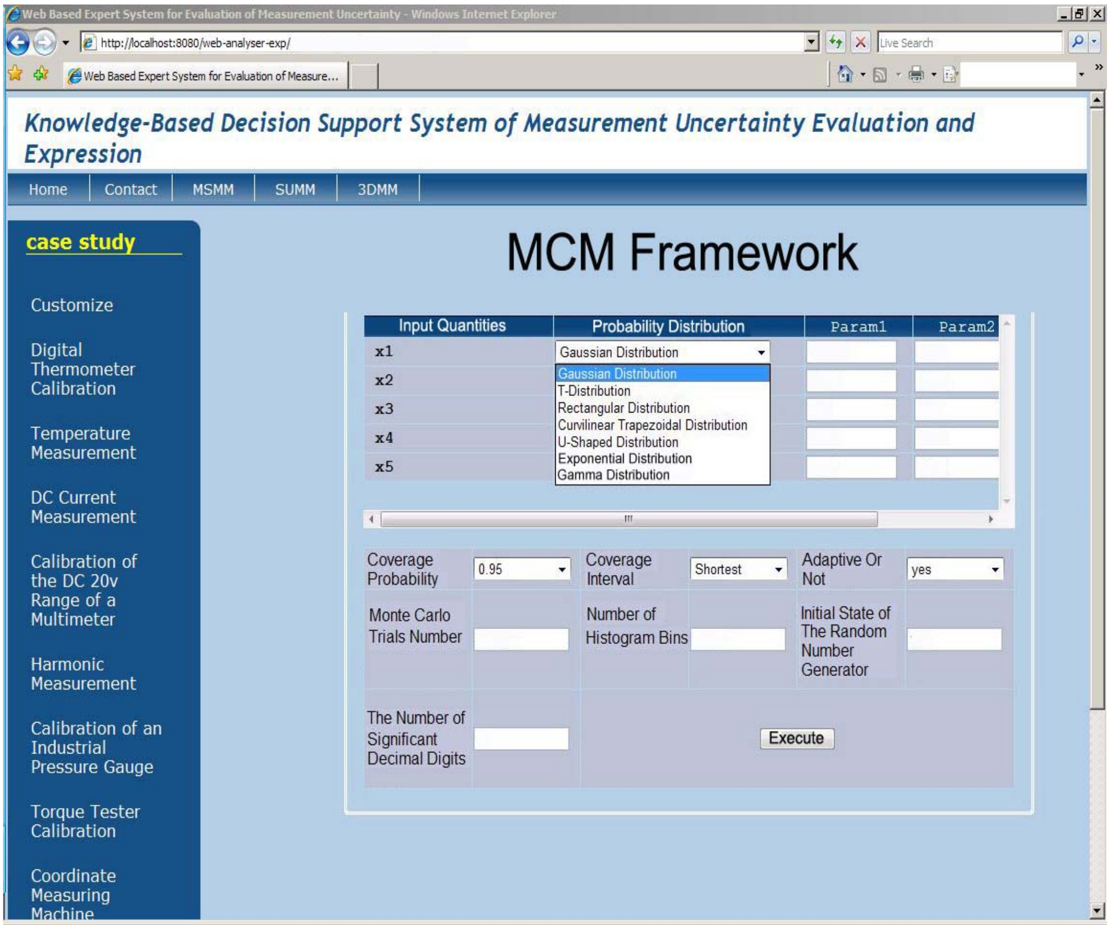


Fig. 5.17 User Interface of MCM Knowledge Base

5.6 Summary

In this chapter, the design and implementation of the proposed web-based Knowledge-Based Decision Support System is elaborated. The proposed system is based on a MVC architecture using TO to carry out the data transfer among the sub-systems. The design of the three sub-systems, MMKB, GUM KB and MCM KB, are discussed, accompanied with detailed explanations and example of their implementations.

Chapter 6 System Testing Result and Case Studies

6.1 Introduction

This chapter presents the testing results of the proposed web-based KB-DSS for measurement uncertainty evaluation through two case studies. Since each knowledge base/sub-system has been tested separately in the last chapter, this chapter focuses on the evaluation of the system. Two of the important mathematical models in modern measurement system are used in the case studies. The first case study concerns the Coordinate Measuring Machine (CMM) measurement, whilst the second case study is related to Mass Calibration.

Before the system testing, the follow talks need to be done on the Server computer:

1) Install the Java JDK and setup J2EE Environment variable *JAVA_HOME*. Also the setup of PATH environment variable to *%PATH%; %JAVA_HOME%\bin* is needed. Fig. 6.1 outlines this step:

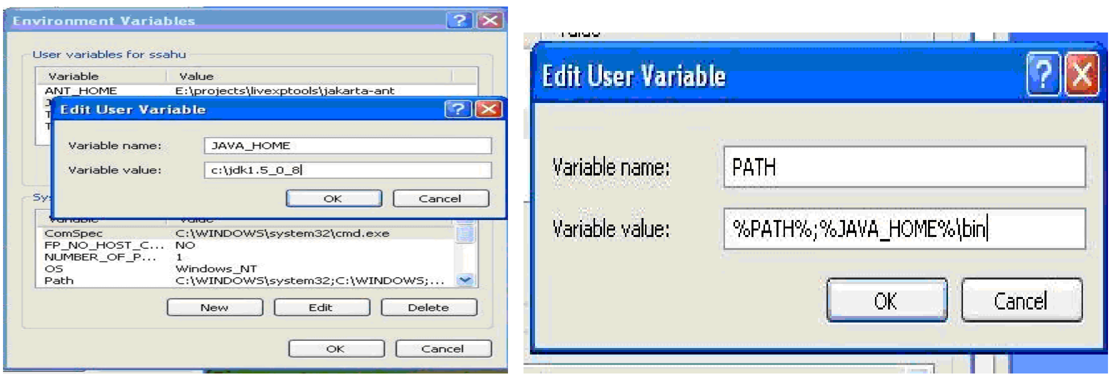


Fig. 6.1 Set J2EE Environments

2) Install Apach Tomcat servlet container;

3) Install Oracle Database 11g;

4) Install Matlab 2008a.

6.2 Case Study I: Coordinate Measuring Machine Measurement

A CMM, as shown in Fig. 6.2, is a machine used to measure the geometry of parts for almost any shape to high precision and is a common and important measurement instrument widely used by testing laboratories.

Fig. 6.2 Coordinate Measuring Machine

This measurement case and related data is adopted from Singapore Laboratory Accreditation Scheme Guide 1 (SINGLAS, 1995). In this case, a CMM is used to measure the length of the cylindrical rod from the centre point of one end to another, as shown in Fig. 6.3.

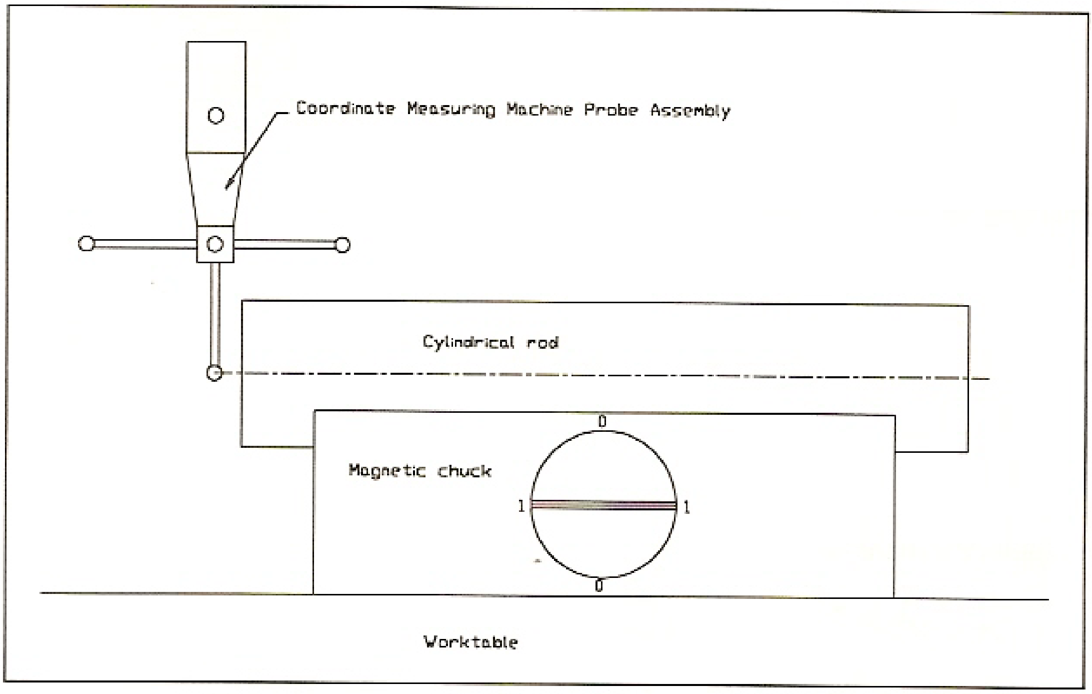


Fig. 6.3 Example of CMM Measuring

Assuming the user know the mathematical model

$$L_r = L_m(1+\alpha_m \Delta T_m - \alpha_r \Delta T_r)$$

Where

L_r is the length of cylindrical rod at 20℃;

L_m is the length as measured by coordinate measuring machine;

α_r is the thermal expansion coefficient of cylindrical rod;

α_m is the thermal expansion coefficient of CMM scales；

ΔT_m is the temperatures deviation of the CMM scales , from during measurement process；

ΔT_r is the temperatures deviation of the cylindrical rod at 20℃ during measurement process.

Source of Uncertainty	Type	u_i	Uncertainty value	Sensitivity coefficient	Probability Distribution	Coverage Factor	Standard Uncertainty	Degrees of Freedom
Repeatability	A	$u(L_{m1})$	-	0.999999	t-distribution	-	0.6μm	9
CMM specification	B	$u(L_{m2})$	±(1.2+3 ℓ)μm	0.999999	Normal	1.96	1.07μm	∞
Expansion of CMM scale	B	$u(\alpha_m)$	$\pm 2 \times 10^{-6}$ °C^{-1}	30.00008	Rectangular	√3	1.2×10^{-6} °C^{-1}	∞
Temperature of CMM scale	B	$u(\Delta T_m)$	±0.5°C	0.0015	Rectangular	√3	0.3°C	∞
Expansion of rod	B	$u(\alpha_r)$	$\pm 2 \times 10^{-6}$ °C^{-1}	-30.00008	Rectangular	√3	1.2×10^{-6} °C^{-1}	∞
Temperature of rod	B	$u(\Delta T_r)$	±0.5°C	-0.00345	Rectangular	√3	0.3°C	∞
Combined	-	$u_c(l_r)$	-	-	t-distribution	-	1.67μm	526
Expanded	-	U	3.3μm	-	t-distribution	1.96	-	526

Fig. 6.4 Original Uncertainty Budget Table from SINGLAS

In the UIMM proposed in this thesis, the user can input the mathematical model directly. Thanks to the symbolic computation approach proposed in this thesis, in the UIMM, the user can either input the original mathematical model Lm*(1+Km*Tm-Km*Tr), or input it as A*(1+B*C-D*E), as shown in Fig 6.5.

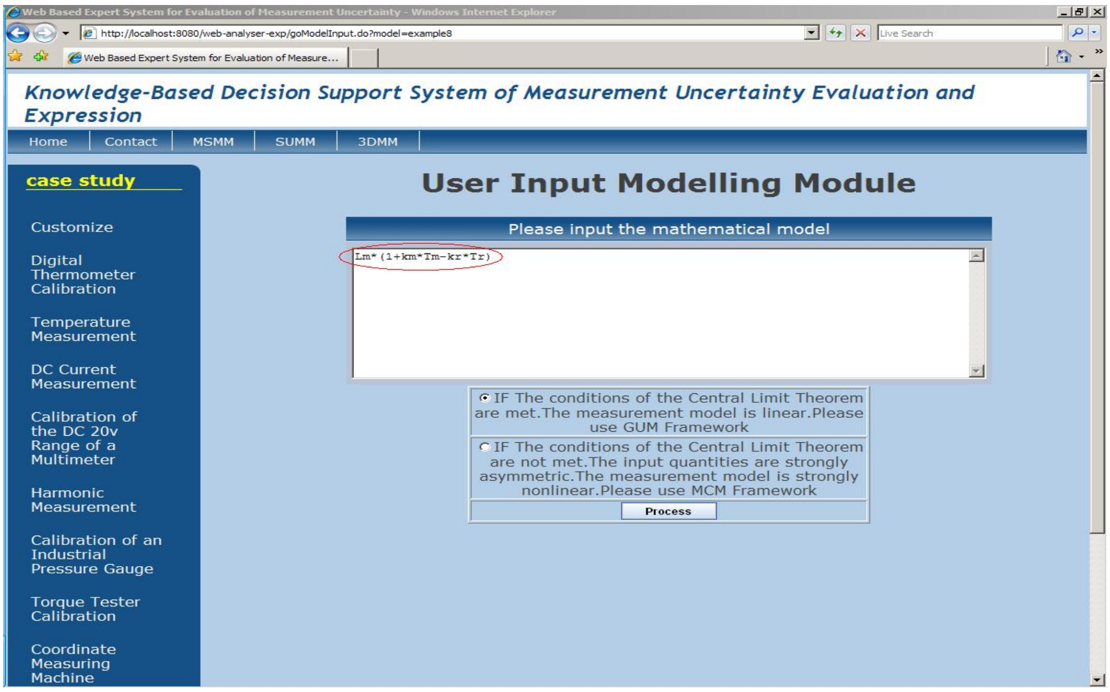


Fig. 6.5 Input Mathematical Model

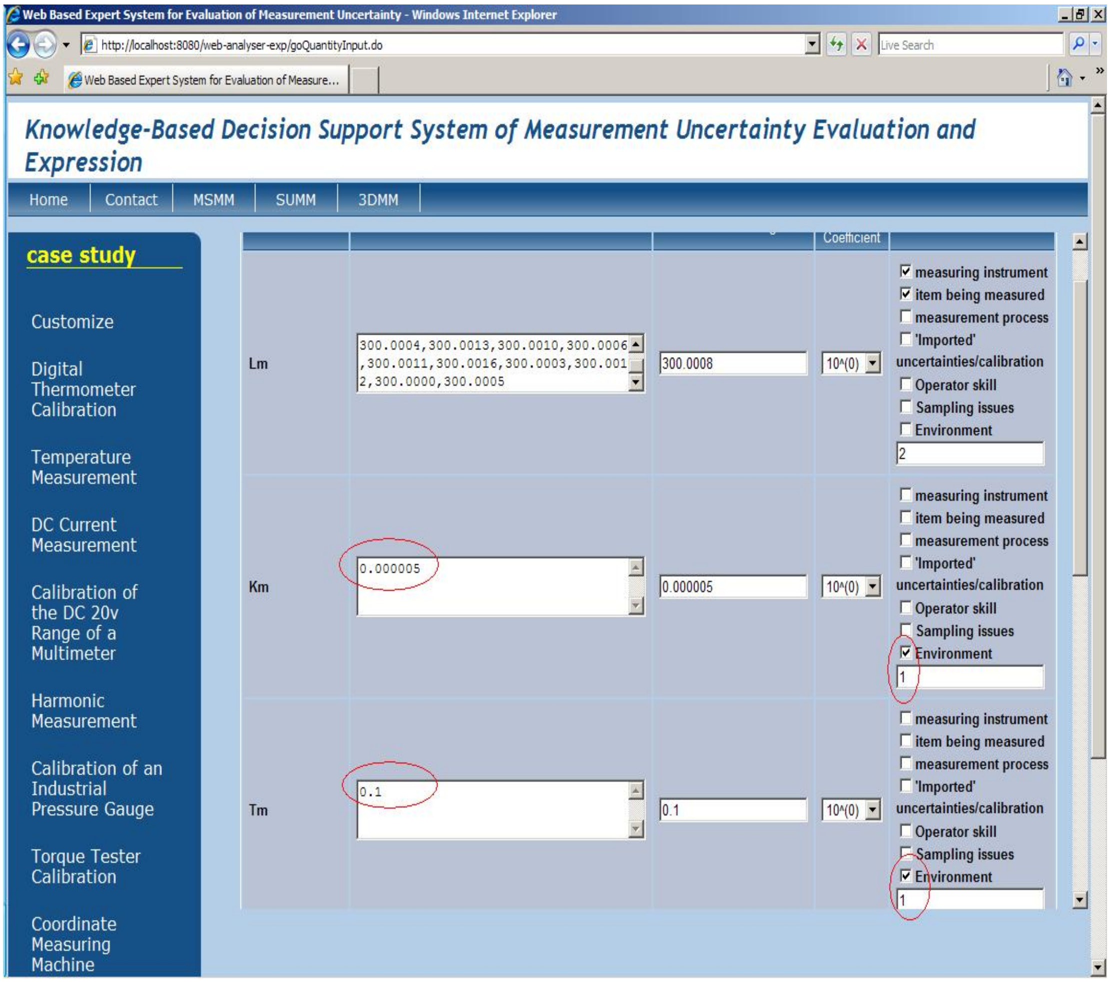
Once the mathematical model is entered, the original data associated with the measurement is processed following the procedure outlined below (Fig. 6.6 shows the procedure):

1) Input the value of each input quantity and the number of uncertainty sources for each of them;

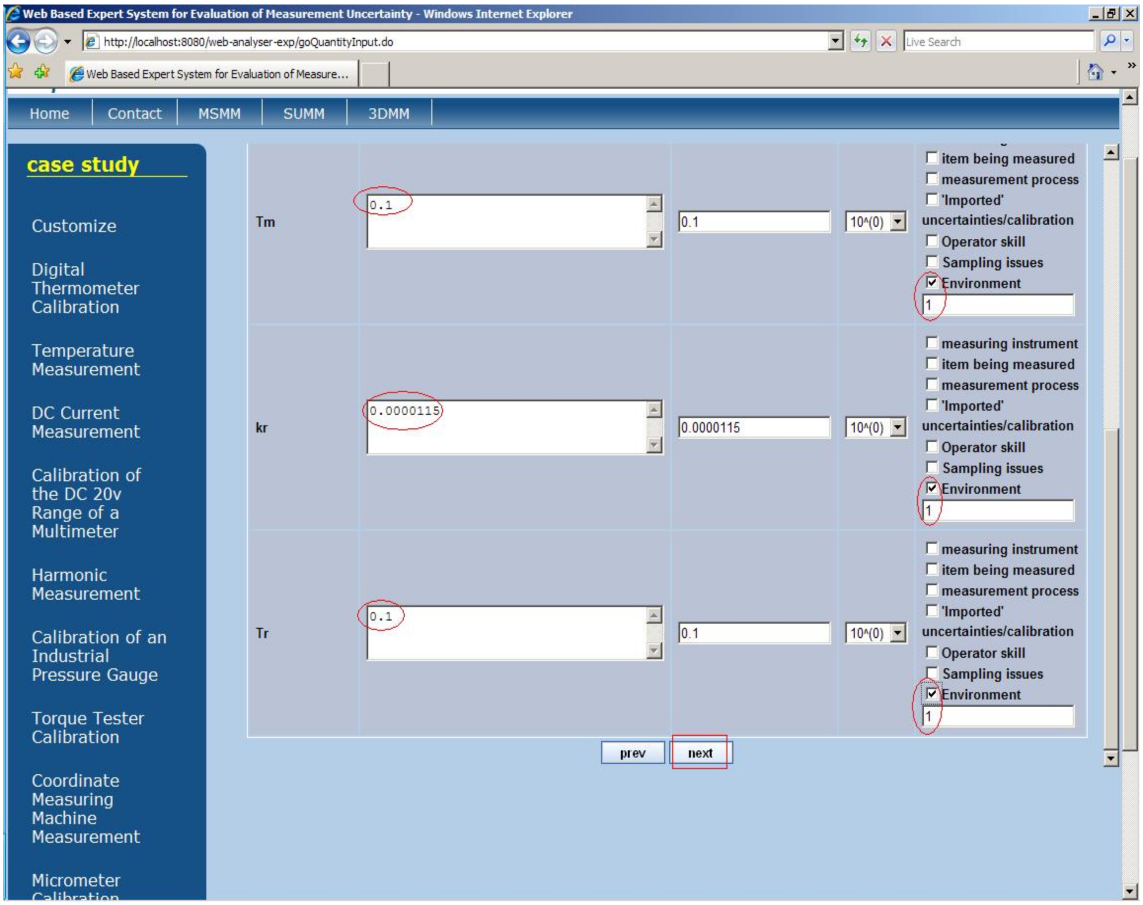
2) Determine the Type of Evaluation and Degree of Freedom for each input quantity;

3) Choose the distribution and input the Quoted Uncertainty for each source;

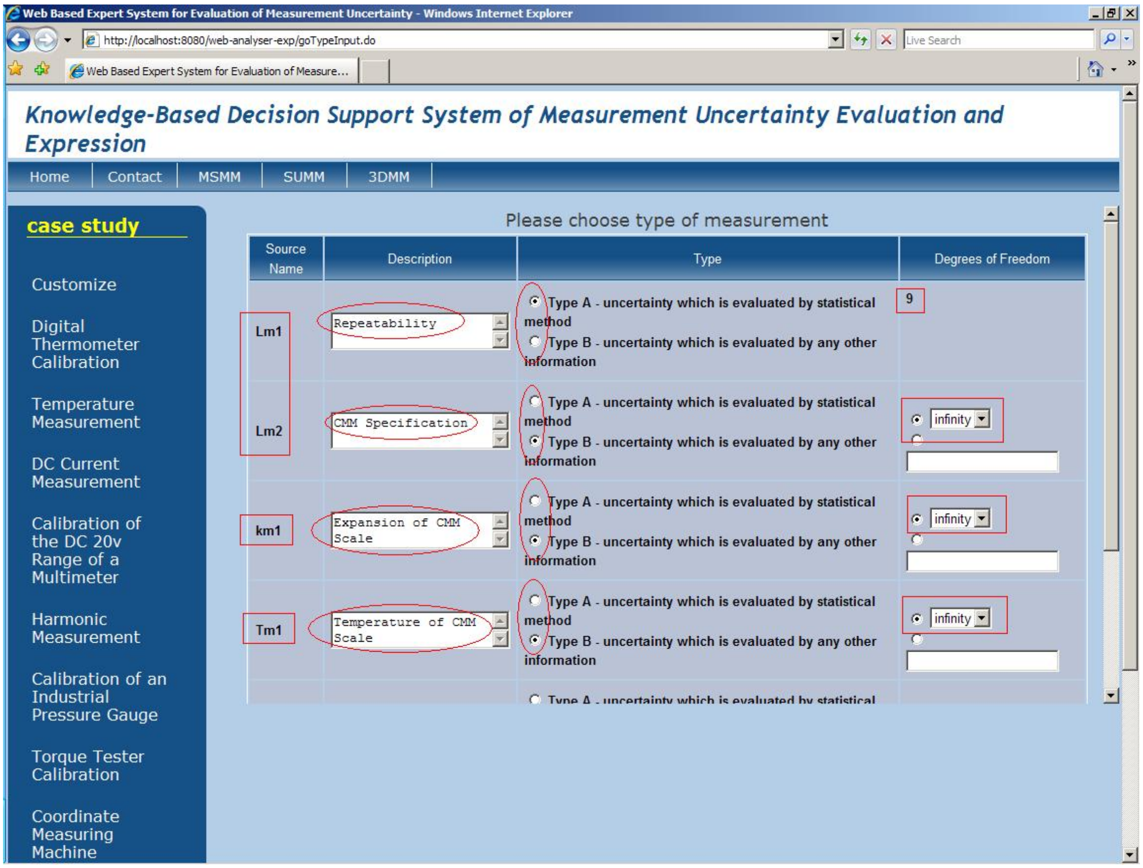
4) Obtain the result and the Uncertainty Budget Table.



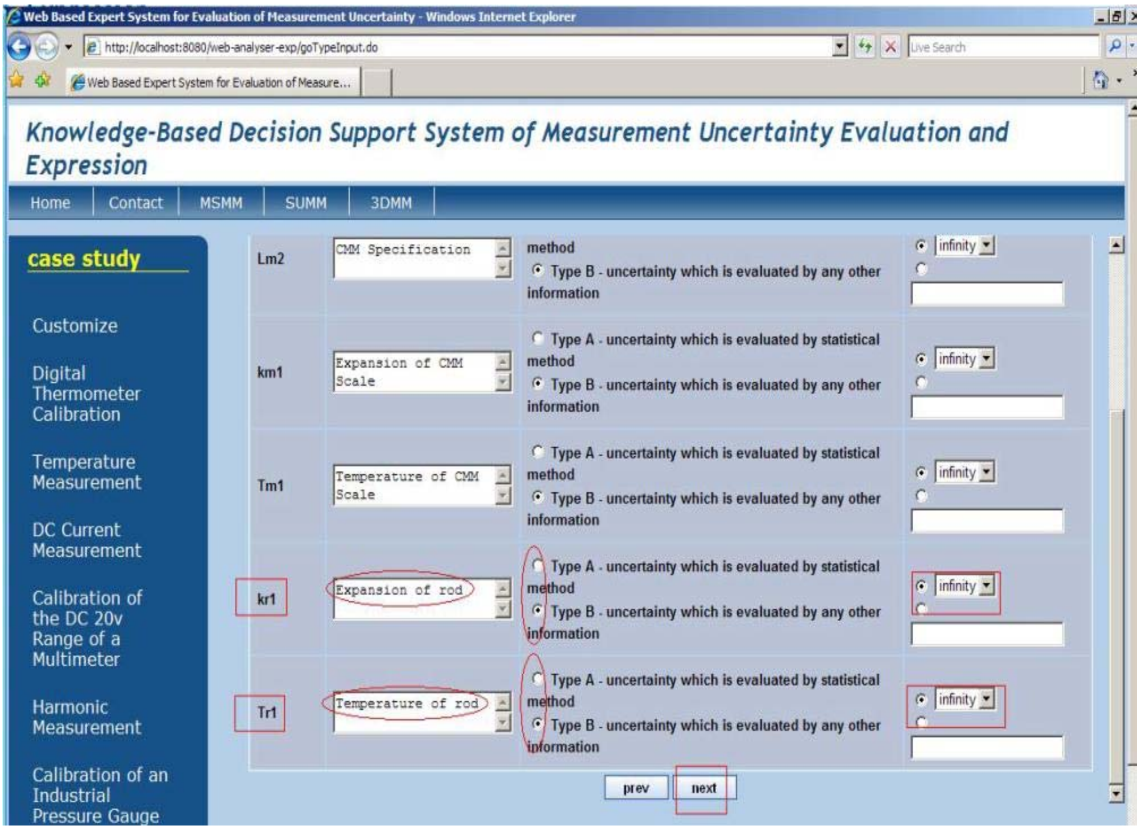
1)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
http://localhost:8080/web-analyser-exp/goQuantityInput.do
Home
Contact
MSMM
SUMM
3DMM
case study
Customize
Digital Thermometer Calibration
Temperature Measurement
DC Current Measurement
Calibration of the DC 20v Range of a Multimeter
Harmonic Measurement
Calibration of an Industrial Pressure Gauge
Torque Tester Calibration
Coordinate Measuring Machine Measurement
Tm
0.1
0.1
10^(0)
item being measured
measurement process
'Imported' uncertainties/calibration
Operator skill
Sampling issues
Environment
kr
0.0000115
0.0000115
measuring instrument
Tr
prev
next

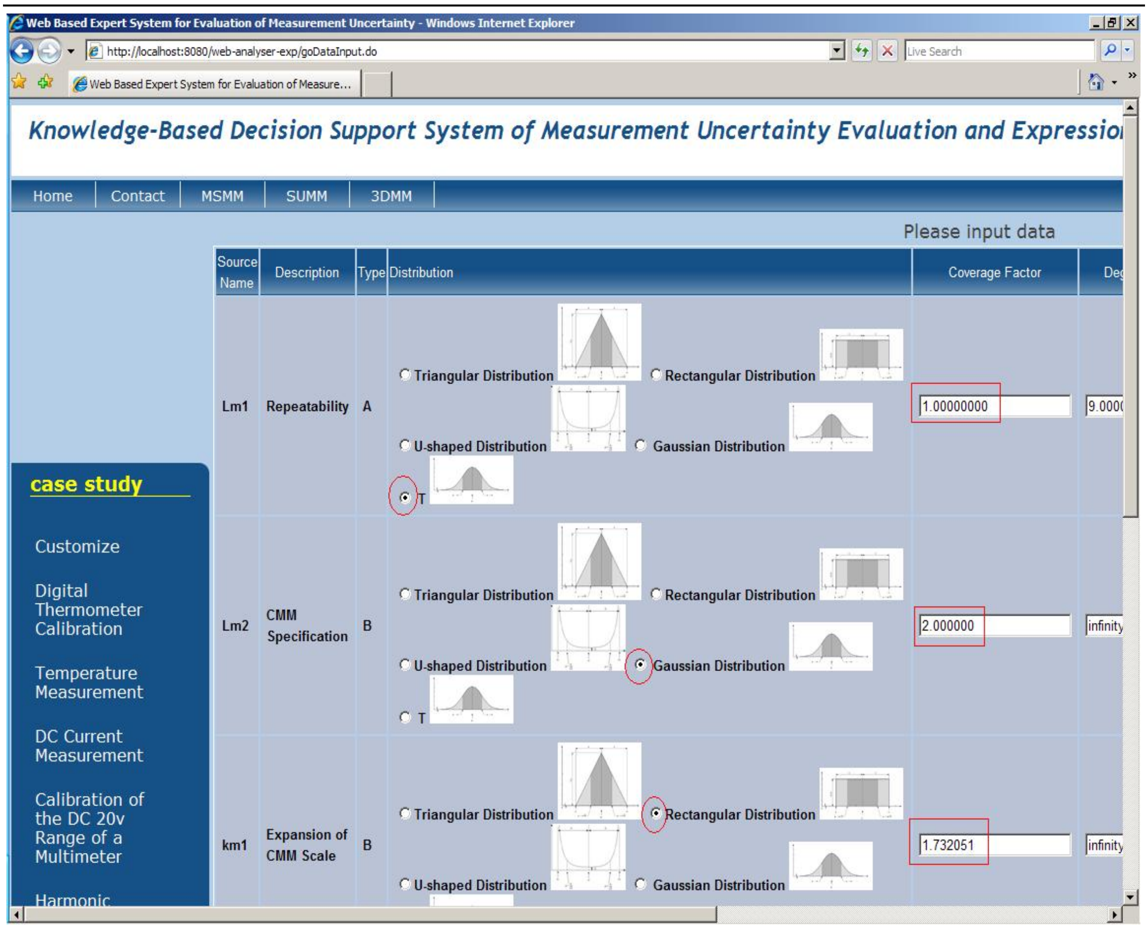
2)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
http://localhost:8080/web-analyser-exp/goTypeInput.do
Knowledge-Based Decision Support System of Measurement Uncertainty Evaluation and Expression
Please choose type of measurement
Source Name
Description
Type
Degrees of Freedom
Lm1
Repeatability
Type A - uncertainty which is evaluated by statistical method
Type B - uncertainty which is evaluated by any other information
9
Lm2
CMM Specification
infinity
km1
Expansion of CMM Scale
Tm1
Temperature of CMM Scale

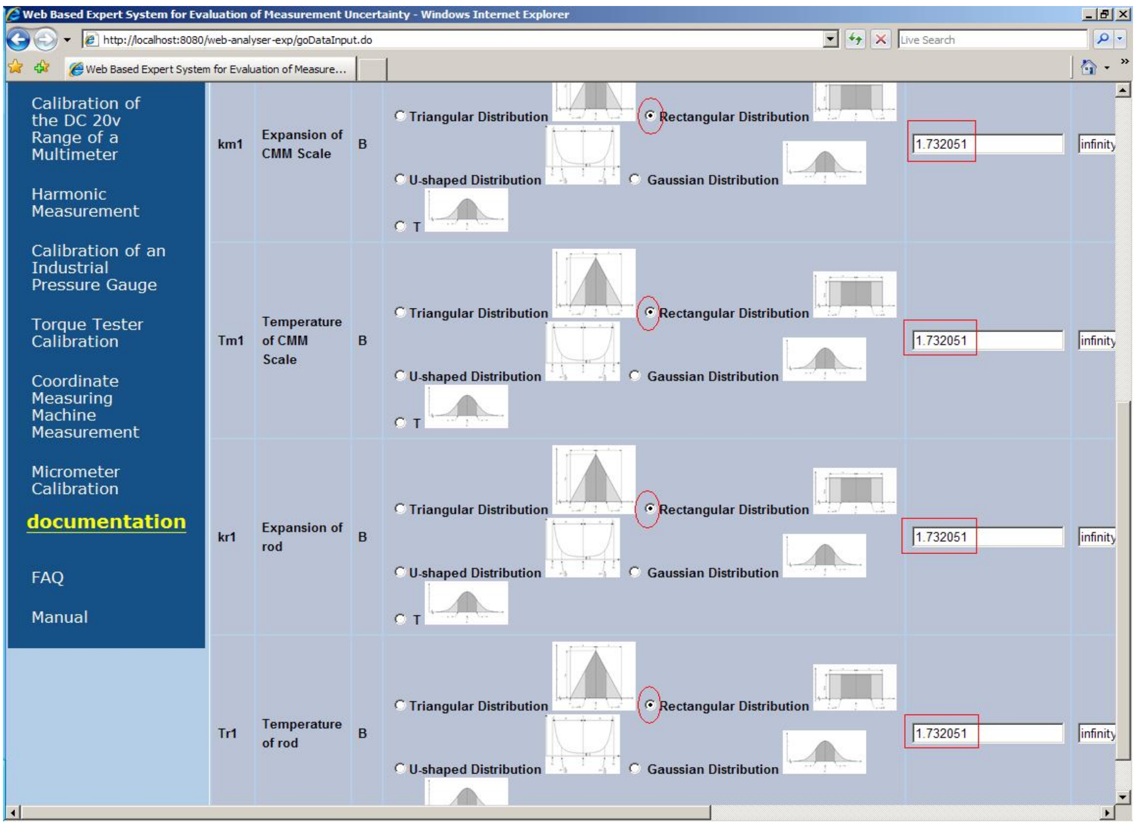
3)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
http://localhost:8080/web-analyser-exp/goTypeInput.do
Knowledge-Based Decision Support System of Measurement Uncertainty Evaluation and Expression
Home
Contact
MSMM
SUMM
3DMM
case study
Customize
Digital Thermometer Calibration
Temperature Measurement
DC Current Measurement
Calibration of the DC 20v Range of a Multimeter
Harmonic Measurement
Calibration of an Industrial Pressure Gauge
Lm2
CMM Specification
km1
Expansion of CMM Scale
Tm1
Temperature of CMM Scale
kr1
Expansion of rod
Tr1
Temperature of rod
Type A - uncertainty which is evaluated by statistical method
Type B - uncertainty which is evaluated by any other information
infinity
prev
next

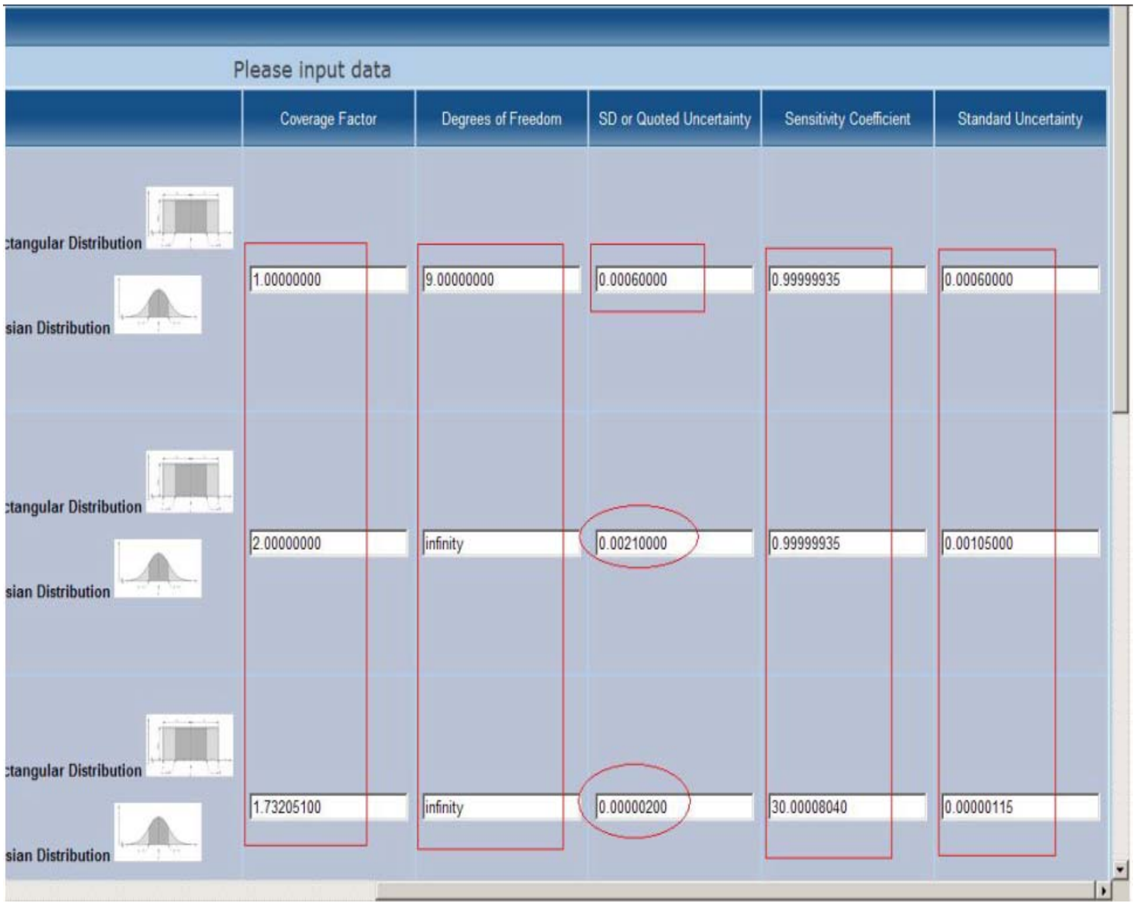
4)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
Knowledge-Based Decision Support System of Measurement Uncertainty Evaluation and Expressio
Home
Contact
MSMM
SUMM
3DMM
Please input data
Source Name
Description
Type
Distribution
Coverage Factor
case study
Customize
Digital Thermometer Calibration
Temperature Measurement
DC Current Measurement
Calibration of the DC 20v Range of a Multimeter
Lm1
Repeatability
A
Triangular Distribution
Rectangular Distribution
U-shaped Distribution
Gaussian Distribution
T
1.00000000
Lm2
CMM Specification
B
2.000000
km1
Expansion of CMM Scale
1.732051
infinity

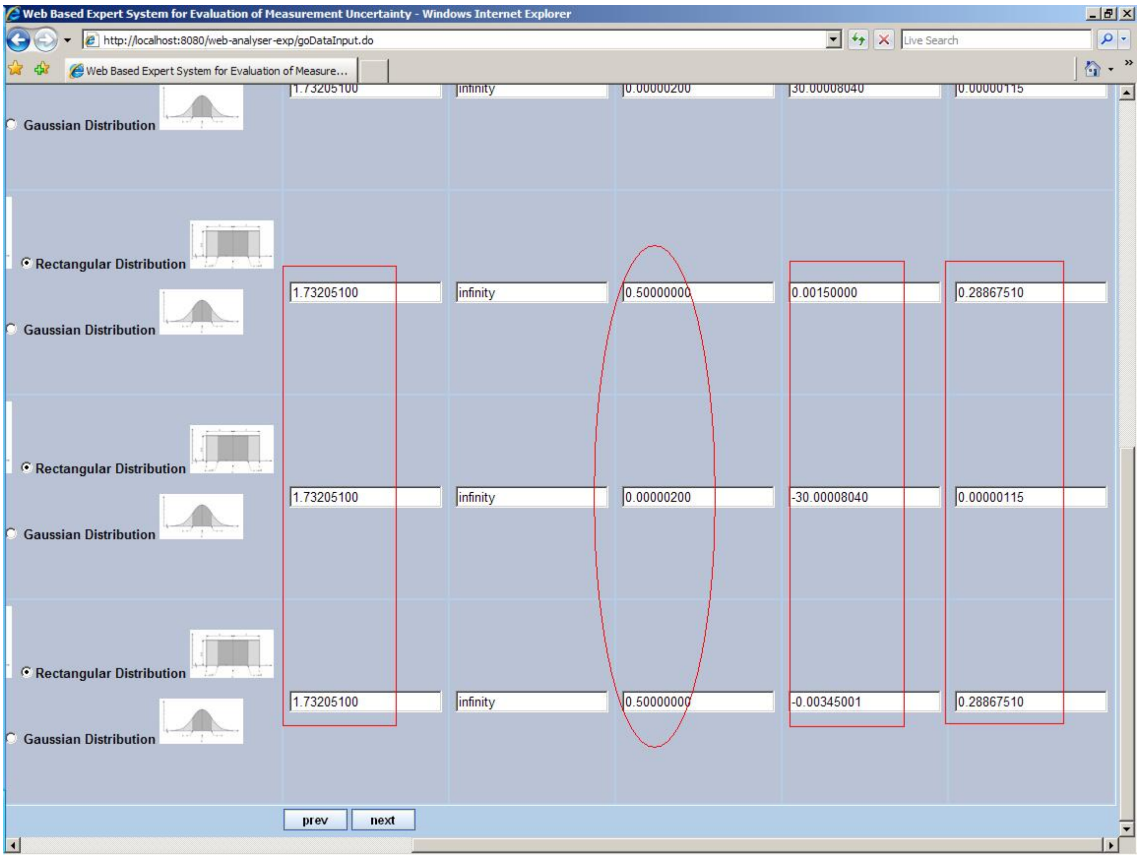
5)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
Calibration of the DC 20v Range of a Multimeter
Harmonic Measurement
Calibration of an Industrial Pressure Gauge
Torque Tester Calibration
Coordinate Measuring Machine Measurement
Micrometer Calibration
documentation
FAQ
Manual
km1
Expansion of CMM Scale
Tm1
Temperature of CMM Scale
kr1
Expansion of rod
Tr1
Temperature of rod
B
Triangular Distribution
Rectangular Distribution
U-shaped Distribution
Gaussian Distribution
T
1.732051
infinity

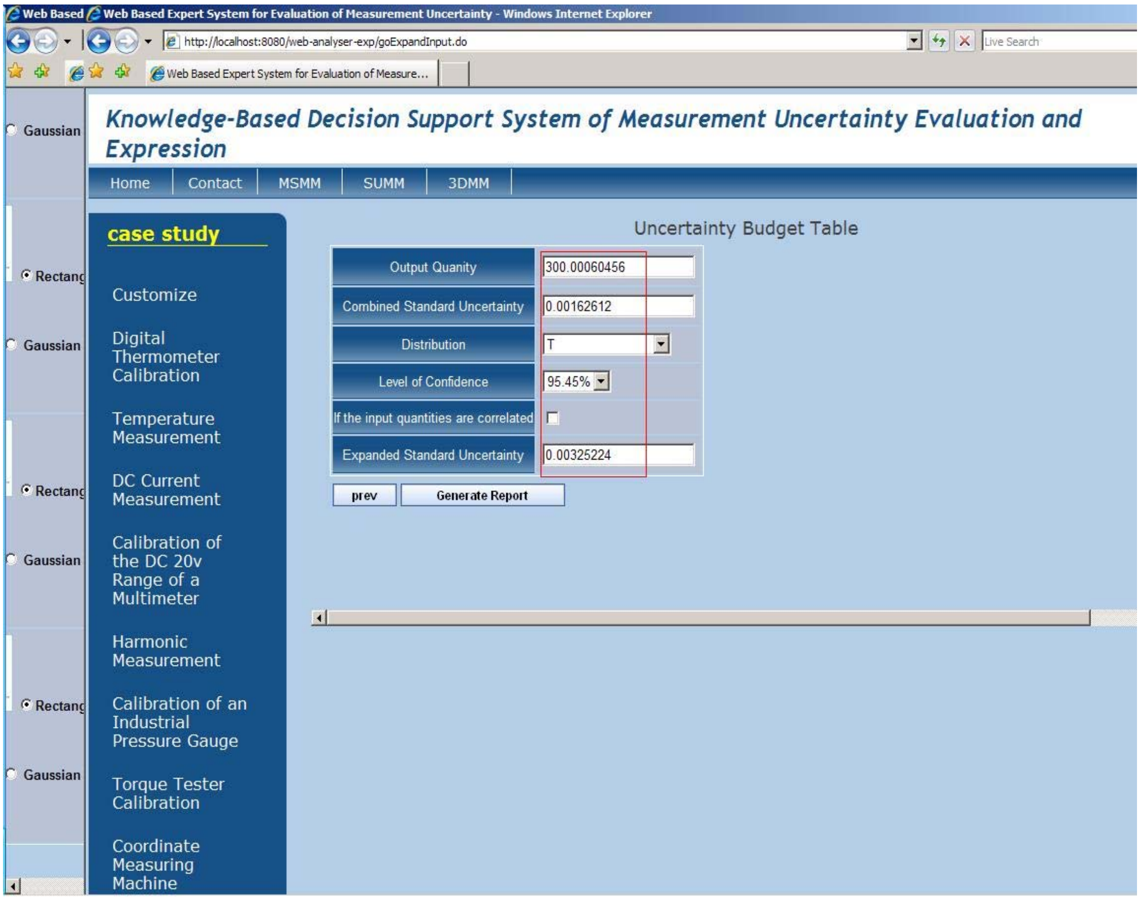
6)


Please input data
Coverage Factor
Degrees of Freedom
SD or Quoted Uncertainty
Sensitivity Coefficient
Standard Uncertainty
ctangular Distribution
sian Distribution
1.00000000
9.00000000
0.00060000
0.99999935
0.00060000
2.00000000
infinity
0.00210000
0.99999935
0.00105000
1.73205100
infinity
0.00000200
30.00008040
0.00000115

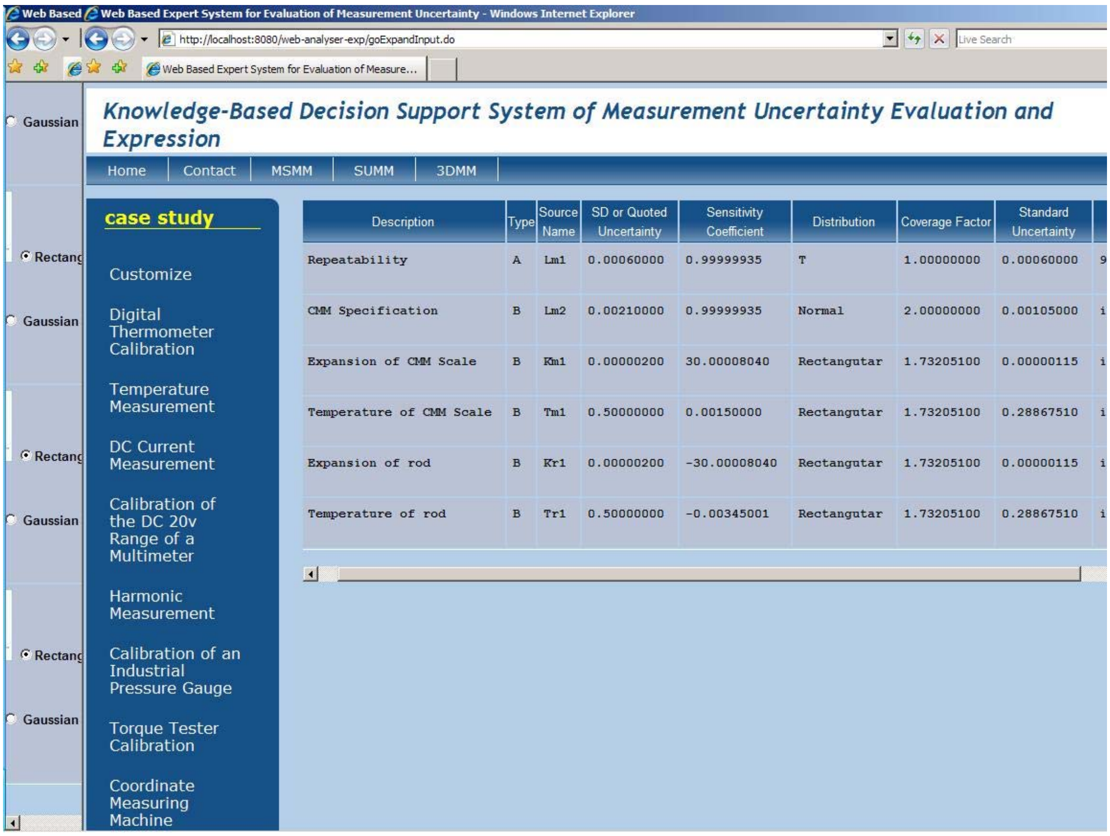
7)


Web Based Expert System for Evaluation of Measurement Uncertainty - Windows Internet Explorer
http://localhost:8080/web-analyser-exp/goDataInput.do
Live Search
Web Based Expert System for Evaluation of Measure...
Gaussian Distribution
Rectangular Distribution
1.73205100
infinity
0.50000000
0.00150000
0.28867510
1.73205100
infinity
0.00000200
-30.00008040
0.00000115
1.73205100
infinity
0.50000000
-0.00345001
0.28867510
prev
next

8)



9)



10)

Fig. 6.6 Procedure of Evaluating the GUM Framework

Through this case study, it is clearly seen that the testing result observed validates a high quality performance of the proposed system in the following sense:

1) The UIMM and the GUM Framework KB work effectively and the function of the proposed system is fully validated with no error observed during the testing;
2) With the user-friendly user interface, the user inputs the data directly into the table and any modification needed is made easy;
3) The proposed computation approach provides high efficiency;
4) The system provides good support to the user to simplify the measurement uncertainty evaluation. At stages where the decision by the user is needed, for example, the selection of the distribution and calculation approach, the user is provided with relevant information which assists the decision making activity.

6.3 Case Study II: Mass Calibration

This example is adopted from GUM SP1 (GUM SP1, 9.3). In Section 4.3.1 of this thesis, the proposed approach for establishing the mathematical model through the MSMM has been discussed. In this case study, we will use

$$\delta_m = (m_{R,c} + \delta_{m_{R,C}})[1 + (\rho_\alpha - \rho_{\alpha_0})(\frac{1}{\rho_W} - \frac{1}{\rho_R})] - m_{nom}$$

as the mathematical model, where δ_m is the mass of a small weight of density ρ_R added to R to balance it with W.

According to the description in GUM SP1 9.3, we can replace the mathematical expression with Y=(X1+X2) *((1+(X3-1.2)*(1/X4-1/X5))-100000. As shown in Fig. 6.7 below, the system automatically generates X1, X2, X3, X4, X5 accordingly.

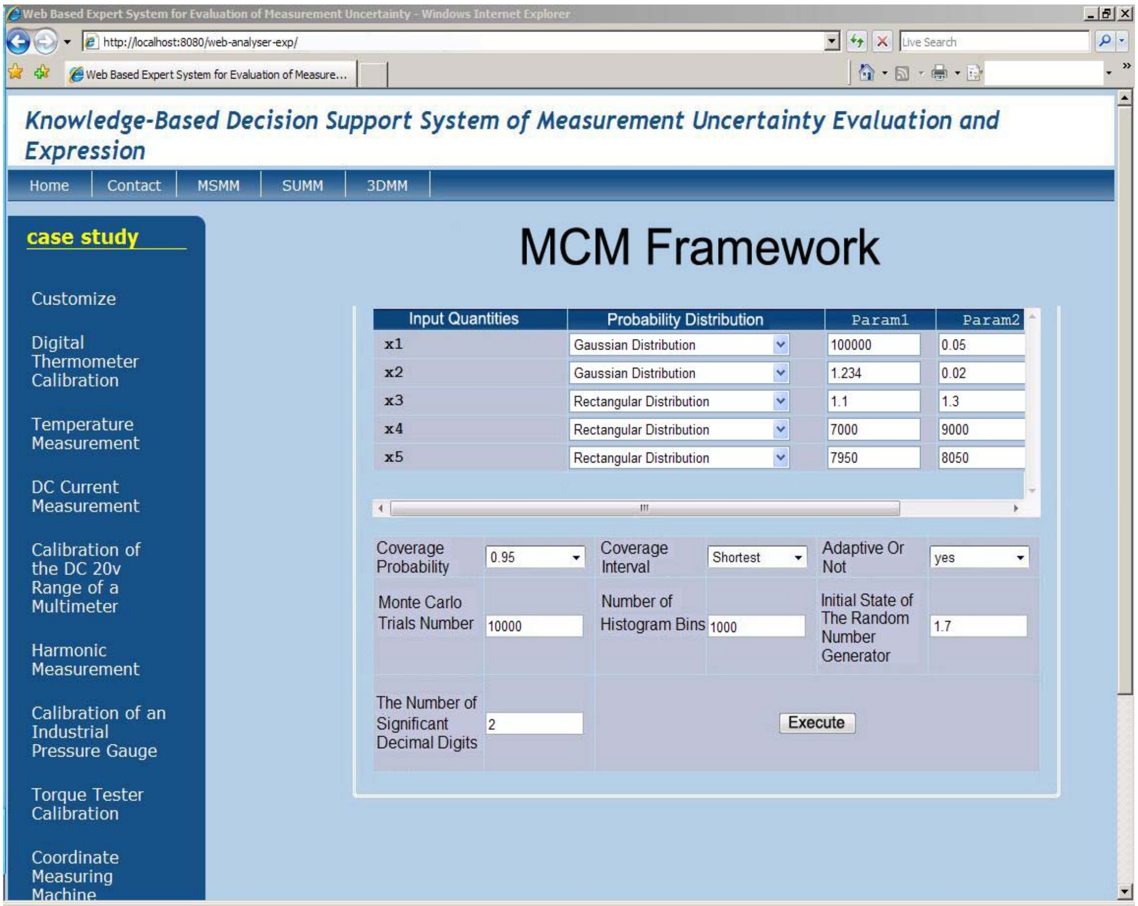
In the next step, the user inputs the data following the procedures of NPLUnc:

1) Assign the Probability Distribution to each Input quantity and define the elements related to the specific input quantity. For example, for Gaussian distribution, param1 (expectation mu) and param2 (standard deviation sigma) need to be defined;

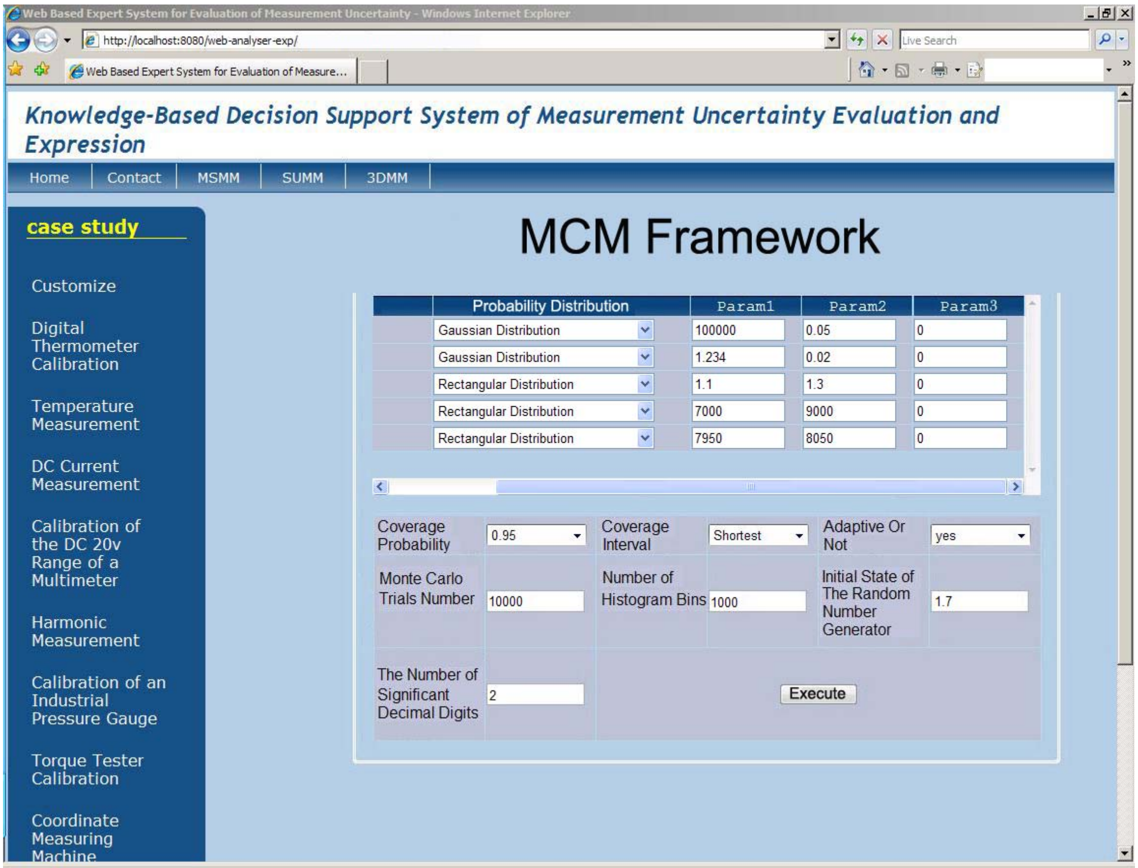
2) The user determines the Coverage Probability, Monte Carlo Trials Number and Coverage Interval etc;

3) Click the "Execute" button to get the results;

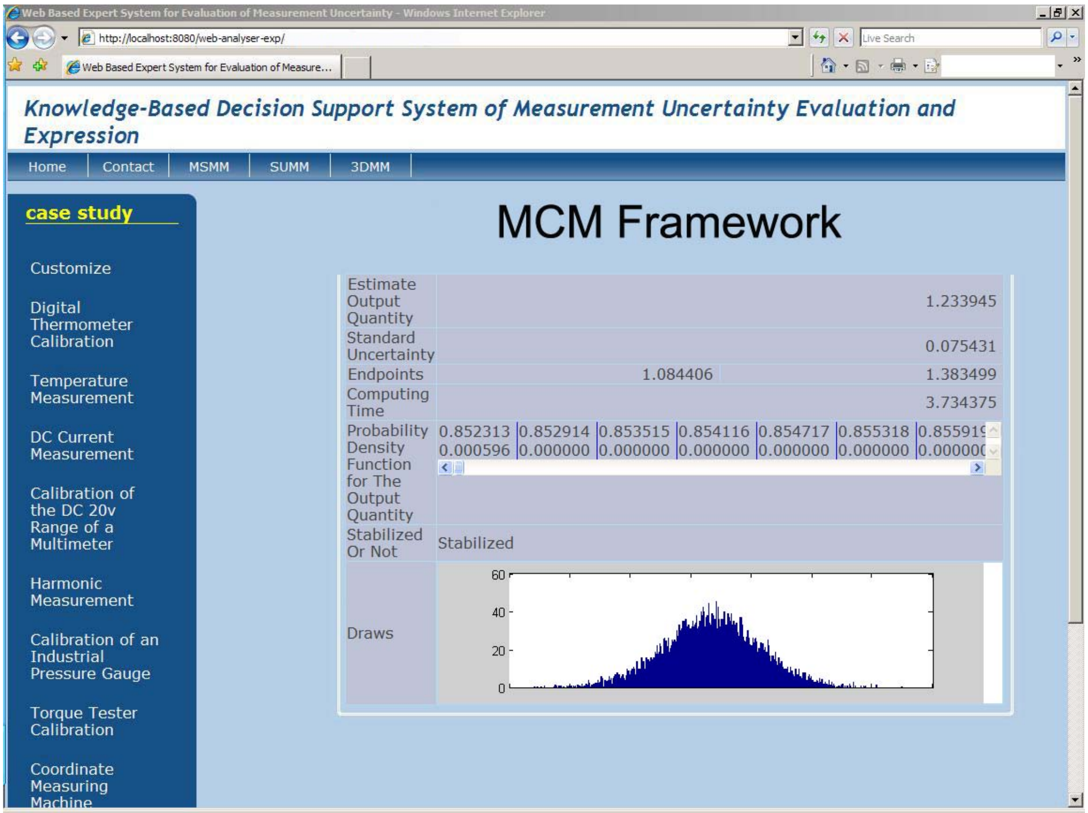
4) Results will be displayed on the JSP.



1) Assign the Probability Distribution of Each Input Quantities



2) Input Other Elements of Each Input Quantities



3) Results

Fig. 6.7 Procedure of Evaluating the MCM Framework

Through this case study and its results, the functionality of the proposed approach is validated, particularly in the following aspects:

1) The MSMM and the MCM Framework KB work successfully in an integrated environment and make the evaluation fully functional in different measurement applications. No error is observed during the testing;

2) It is proven again that the system offers a user-friendly user interface, which enables easy and efficient data input, update and modification;

3) The proposed computation approach is highly efficient. As shown in the result JSP, the computing time is 3.73 seconds.

6.4 Summary

In this chapter, the testing and evaluation for the proposed web-based KB-DSS is carried out through two case studies, adopting examples from documents based on GUM and GUM SP1, respectively. The testing results from the case studies have demonstrated that the proposed system works effectively with full functional realisation. The system proves to be user-friendly and provides substantial user support.

In particular, the application of the case studies shows that the proposed system is suitable for both the GUM Uncertainty Framework and the GUM SP1 MCM Framework, making it a general system showing great potential for wide applications in measurement.

Chapter 7 Conclusions and Recommendations for Future Work

In this chapter, the conclusions of the research are drawn, accompanied with the outline of the contributions to knowledge. Based on this thesis and the current development in the area of measurement uncertainty, recommendations for future research and work are also highlighted.

7.1 Conclusions

Based on the discussions in the previous chapters and the results from the evaluation of the proposed system, this thesis draws the conclusions as follows:

1) Following the rapid development of the computer technology especially the development of the Internet, web applications offers its users unprecedented advantages which the traditional desktop applications could not provide. In a web application, the user around the world can visit and run a program at a server via a web browser. A novel web-based evaluation system for measurement uncertainty is developed in this thesis. In order to achieve an optimised performance of the web application, a Model-View-Controller architecture is used for the proposed system. Java Transfer Object is utilised to optimise the data transfer of the system to ensure its stability, security, and efficiency.

2) Based on the investigation and discussions presented in the thesis, ES and DSS have been utilised for measurement uncertainty evaluation with the potential to greatly improve the quality and efficiency of the evaluation process. In particular, the integration of ES and DSS has been proposed to construct an intelligent Knowledge-based Decision Support System. In the system, the three knowledge bases, MMKB, GUM KB and MCM KB are all sub-systems. They are basically

independent systems, but they can also run separately when necessary.

3) The evaluation of the measurement uncertainty involves a large amount of calculation and computation. Based on the discussions in the previous chapters, many of the currently available systems based on the GUM Uncertainty Framework for measurement uncertainty evaluation uses numerical computation, which tends to limit the user-friendliness of the systems and consequently impedes the wide adoption of the GUM framework. Use of symbolic computation has been demonstrated for uncertainty evaluation based on GUM Framework, which overcomes the limitations of the pure numerical computation.

4) In order to assist the user of the proposed system in the modelling of the measurement mathematical model, an approach using relational database, rule-based ES and Three Dimensional Measurement System Oriented Modelling is used in the system.

5) In the design and implementation, the proposed system has integrated NPLUnc's MCM calculation routines based on Matlab. A number of modifications and enhancements have been made to the MCM method. The proposed system can be used for both GUM Uncertainty Framework and GUM SP1 MCM Framework.

7.2 Contributions to knowledge

Based on the discussions and investigations in this thesis and the proposed web-based KB-DSS for evaluation of measurement uncertainty, the following contributions to knowledge resulted from the thesis:

1) A novel and generic web-based application has been developed for evaluation of measurement uncertainty. As a web application it has the advantages of great applicability and accessibility over the Internet. As a generic system, it can

perform general, user-friendly and efficient evaluation of measurement uncertainty. The maintenance of the system is also made more efficient and more cost-effective.

2) An original intelligent KB-DSS system, which integrates ES and DSS to incorporate the advantages of both systems, has been developed for measurement uncertainty evaluation. The system provides friendly user interface and assistance in measurement modelling and computation method selection. This approach significantly improves the performance of the evaluation process for measurement uncertainty.

3) Symbolic computation has been applied for uncertainty evaluation based on GUM Framework. This overcomes the limitations of the conventional numerical computation, simplifies and enhances the evaluation process, thus facilitating their wider applications.

4) Four modelling modules have been proposed in the thesis, which can not only assist testing and calibration laboratories staff, but also help in specific measurement cases in other measurement fields. Each modelling module is an independent sub system and can be used alone or combined where measurement modelling is needed.

5) The proposed system has also integrated the standard MCM computation routines based upon Matlab with additional enhancements provided. The overall system is capable of general evaluation of measurement uncertainty based on both GUM Framework and GUM SP1 MCM framework.

7.3 Recommendations for future work

The area of measurement uncertainty continues to develop actively and there are areas that need further investigation and exploration. Specifically, this thesis recommends the following areas for future research and work:

1) In Chapter 4, a measurement system – oriented modelling module is proposed, the centre of which is a big and complex database. Further development and consolidation of this database is required to extend its coverage. A comprehensive database will be valuable to solve modelling problems in modern measurement system.

2) The user interface of the proposed system has room for further optimisation. Its interactivity with the users can be extended using the Web and multi-media technologies. For example, E-learning is one of the possible directions.

3) In the MCM Framework, more probability distributions can be added to offer the user more choices to solve practical problems in a broader range of measurements.

4) All web applications have a common problem of Internet security. More researches and investigation are needed to ensure the security of the data and information of the web-based system.

5) The BIPM is constantly updating the documents for measurement uncertainty. The system can be further updated based on the publication of new documents and guides. For example, GUM SP2, *Evaluation of measurement data – Supplement 2 to the "Guide to the expression of uncertainty in measurement" – Models with any number of output quantities* is scheduled to be made public in the near future. Also, the GUM SP3, *Evaluation of measurement data – Supplement 3 to the "Guide to the expression of uncertainty in measurement" – Modelling*, is currently at an early stage of preparation. These documents may bring useful information for further development of the system.

References

Alur, D., Crupi, J. and Malks, D., (2003), Core J2EE Patterns: Best Practices and Design Strategies, 2nd Edition, Prentice Hall / Sun Microsystems Press, ISBN:0131422464.

Arentze, T.A., Borgers, A.W.J. and Timmermans, H.J.P., (1995), "The integration of expert knowledge in decision support systems for facility location planning", Computers, Environment and Urban Systems, Volume 19, Issue 4, Pages 227-247.

Assaf, H. and Saadeh, M., (2008), "Assessing water quality management options in the Upper Litani Basin, Lebanon, using an integrated GIS-based decision support system", Environmental Modelling and Software, Volume 23, Issue 10-11, Pages 1327-1337.

Azadeh, A., Fam, I.M., Khoshnoud, M. and Nikafrouz, M., (2008), "Design and implementation of a fuzzy expert system for performance assessment of an integrated health, safety, environment (HSE) and ergonomics system: The case of a gas refinery", Information Sciences, Volume 178, Issue 22, Pages 4280-4300.

Bell, S., (1999), "A Beginner's Guide to Uncertainty of Measurement", Measurement Good Practice Guide No. 11, National Physical Laboratory, UK.

Bentley, J.P., (1988), Principles of Measurement Systems, 2nd ed. White Plains, NY: Longman.

Bich, W., (2006), "Uncertainty evaluation using a Monte Carlo method", APAT IUPAC International Workshop on Combining and reporting analytical results, Rome.

Bich, W., (2008), "Uncertainty Evaluation by Means of a Monte Carlo Approach", BIPM Workshop 2 on CCRI (II) Activity Uncertainties and Comparisons, National Institute of Metrological Research Italy.

Bich, W., (2007), "Recent Developments in Uncertainty Evaluation, Metrology and fundamental constants", Proceedings of the International School of Physics "Enrico Fermi", Volume 166, Page 81-94.

BIPM, IEC, IFCC, ISO, IUPAC, IUPAP and OIML, (1993-1995), "Guide to the Expression of Uncertainty in Measurement", GUM, Geneva, Switzerland, International Organization for Standardization, ISBN 92-67-10188-9.

BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML, (2008), "Evaluation of measurement data — Supplement 1 to the "Guide to the expression of uncertainty in measurement" — Propagation of distributions using a Monte Carlo method", GUM SP1, Joint Committee for Guides in Metrology.

Booty, W.G., Lam, D.C.L., Wong, I.W.S. and Siconolfi, P., (2001), "Design and implementation of an environmental decision support system", Environmental Modelling and Software, Volume 16, Issue 5, Pages 453-458.

Castrup, S., (2004), "A Comprehensive Comparison of Uncertainty Analysis Tools", Measurement Science Conference Anaheim, CA.

Celikoglu, H.B. and Cigizoglu, H.K., (2007), "Modelling public transport trips by radial basis function neural networks", Mathematical and Computer Modelling, Volume 45, Issues 3-4, Pages 480-489.

Chakraborty, S. and Dey, S., (2007), "QFD-based expert system for non-traditional machining processes selection", Expert Systems with Applications, Volume 32, Issue 4, Pages 1208-1217.

Cox, M.G., Dainton, M.P. and Harris, P.M., (2001), "Uncertainty and statistical modeling, Software Support for Metrology Best Practice Guide No. 6", National Physical Laboratory, UK.

Cox M.G. and Harris P.M., (2006), "Software specifications for uncertainty evaluation", NPL Report DEM-ES-010, Revised/Corrected November 2008, National Physical Laboratory, UK.

Cox, M.G., Dainton, M.P., Forbes, A.B. and Harris, P.M., (2001), "Use of Monte Carlo Simulation for Uncertainty Evaluation in Metrology", Advanced Mathematical & Computational Tools in Metrology V, pp 93 - 105.

Cox, M.G., Dainton, M.P. and Harris, P.M., (2001), "Software Specifications for Uncertainty Calculation and Associated Statistical Analysis", NPL Report CMSC 10/01, National Physical Laboratory, UK.

Cox, M.G., Harris, P.M. and Smith, I.M., (2008), "NPL Report MS 1 - Software for GUM Supplement 1: User Manual", National Physical Laboratory, UK.

Cox, M.G. and Siebert, B. R. L., (2006), "The use of a Monte Carlo method for evaluating uncertainty and expanded uncertainty", Metrologia, Volume 43, Issue 4, Pages S178-S188.

Crossland, M.D., Wynne, B.E. and Perkins, W.C., (1995), "Spatial decision support systems: An overview of technology and a test of efficacy", Decision Support Systems, Volume 14, Issue 3, Pages 219-235.

Dargam, F.C.C. and Perz, E.W., (1998), "A decision support system for power plant design", European Journal of Operational Research, 109 (2), pp. 310-320.

Davis, M., "Struts, an open-source MVC implementation, Manage complexity in large Web sites with this servlets and JSP framework", [online], Available at http://www.ibm.com/developerworks/java/library/j-struts/, [Accessed in January, 2009].

DiNucci, D., (1999). "Fragmented Future". *Print* 53 (4): 32.

Dokas, I. M., (2005), "Developing Web Sites For Web Based Expert Systems: A Web Engineering Approach", In the Proceedings of the Information Technologies in Environmental Engineering (ITEE'2005), pp. 202-217.

Duan, Y., Edwards, J.S., and Xu, M.X., (2005), "Web-based expert systems: benefits and challenges", Information & Management, Volume 42, Issue 6, Pages 799-811.

EA, (1999), "Expression of the Uncertainty of Measurement in Calibration", Document EA-4/02, European co-operation for Accreditation.

EA Laboratory Committee, (2003), "EA guidelines on the expression of uncertainty in quantitative testing", Document EA-4/16, European co-operation for Accreditation.

Ecker, K., Gupta, J.N.D. and Schmidt, G., (1997), "A framework for decision support systems for scheduling problems", European Journal of Operational Research, Volume 101, Issue 3, Pages 452-462.

Eckstein, R., (2007), "Java SE Application Design With MVC", [online], Available at http://java.sun.com/developer/technicalArticles/javase/mvc/, [Accessed in February, 2009].

Eich, B., (1998), "MAKING WEB PAGES COME ALIVE", [online], Available at http://web.archive.org/web/20080208124612/http://wp.netscape.com/comprod/columns/techvision/innovators_be.html, [Accessed in March, 2009].

Elster, C., Wöger, W. and Cox, M.G., (2007), "Draft GUM Supplement 1 and Bayesian analysis", Metrologia, 44, L31–L32.

Encyclopædia Britannica, (2009), "Information system", [online], Available at: http://www.britannica.com/EBchecked/topic/287895/information-system, [Accessed in January 2009].

Eurachem Working Group on Uncertainty in Chemical Measurement, (1995), "Quantifying Uncertainty in Analytical Measurement", First Edition, Eurachem.

Eurolab, (2006), "Guide to the evaluation of measurement uncertainty for quantitative test results", Eurolab technical report No 1/2006.

Eurolab, (2007), "Measurement uncertainty revisited: Alternative approaches to uncertainty evaluation", Eurolab Technical report No 1/2007.

Fernandez, M.S., Alegre Calderon, J.M. and Bravo Diez, P.M., (2009), "Implementation in MATLAB of the adaptive Monte Carlo method for the evaluation of measurement uncertainties", Accreditation and Quality Assurance, Volume 14, Issue 2, Pages 95-106.

Finlay, P. N., (1990), "Decision support systems and expert systems: A comparison of their components and design methodologies", Computers & Operations Research, Volume 17, Issue 6, Pages 535-543.

Flanagan, D., (2006), JavaScript: The Definitive Guide, 5th Edition, O'Reilly Media, 2006. [online], Available at http://my.safaribooksonline.com/0596101996, [Accessed in February, 2009].

Ford, F. N., (1985), "Decision support systems and expert systems: A comparison", Information & Management, Volume 8, Issue 1, Pages 21-26.

Frenkel, D., (2004), "Introduction to Monte Carlo Methods", Computational Soft Matter: From Synthetic Polymers to Proteins, NIC Series, Volume 23, pp. 29-60.

Garrett, J.J., (2005), "Ajax: A New Approach to Web Applications", [online], Available at http://www.adaptivepath.com/ideas/essays/archives/000385.php, [Accessed in March, 2009].

Gavin, J.P., (2006), Oracle Performance Tuning for 10gR2, Second Edition, Digital Press, ISBN: 978-1555583453.

Gosling, J., Joy, B., Steele G., and Bracha, G., (2005), The Java language specification, third edition. Addison-Wesley, 2005, ISBN: 978-0321246783.

Howarth, P. and Redgrave, F., (2008), "Metrology – in short", 3rd edition, EURAMET.

ILAC, (2002), "Introducing the Concept of Uncertainty of Measurement in Testing in Association with the Application of the Standard ISO/IEC 17025". ILAC-G17:2002, International Laboratory Accreditation Cooperation.

Iqbal, A., He, N., Li, L. and Dar, N.U., (2007), "A fuzzy expert system for optimizing parameters and predicting performance measures in hard-milling process", Expert

Systems with Applications, Volume 32, Issue 4, Pages 1020-1027.

Jang, S.Y., Parkt, J. and Park, N., (1996), “An integrated decision support system for FMS production planning and scheduling problems”, International Journal of Advanced Manufacturing Technology, Volume 11, Issue 2 , Pages 101-110.

Java BluePrints, (2002),“Model-View-Controller”, [online], Available at http://java.sun.com/blueprints/patterns/MVC-detailed.html, [Accessed in January, 2009].

JCGM, (2008), “International Vocabulary of Metrology – Basic and General Concepts and Associated Terms”, VIM, 3rd edition, JCGM 200:2008.

Jolly, R., (2009), “jscl-meditor - java symbolic computing library and mathematical editor”, [online], Available at http://jscl-meditor.sourceforge.net/, [Accessed in April, 2009].

Jurado, J.M. and Alcazar, A., (2005), “A software package comparison for uncertainty measurement estimation according to GUM”, Accreditation and Quality Assurance, 10, 373-381.

Kacker, R. N., (2006), “Bayesian alternative to the ISO-GUM's use of the Welch-Satterthwaite formula”, Metrologia, Volume 43, Issue 1, Pages 1-11.

Keen, P.G.W. and Morton, M.S.S., (1978), Decision Support Systems: An Organizational Perspective. Reading, MA: Addison-Wesley.

Kengpol, A. and O'Brien, C., (2001), “Development of a decision support tool for the selection of advanced technology to achieve rapid product development”, International Journal of Production Economics, Volume 69, Issue 2, Pages 177-191.

Klein, M.R. and Methlie, L.B., (1995), Knowledge-Based Decision Support Systems: With Applications in Business, 2nd edition, John Wiley & Sons, Inc. New York, NY, ISBN: 978-0471952954.

Liao, S.H., (2005), “Expert system methodologies and applications—a decade review from 1995 to 2004”, Expert Systems with Applications, Volume 28, Issue 1, Pages 93-103.

Lira, I., (2002), Evaluating the Uncertainty of Measurement. Fundamentals and Practical Guidance. CRC Press, ISBN9780750308403.

Lira, I. and Wöger, W., (2001), “Bayesian evaluation of the standard uncertainty and coverage probability in a simple measurement model”, Measurement Science and Technology, 12 (8), pp. 1172-1179.

Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A., and Lederberg, J., (1993), “DENDRAL: A case study of the first expert system for scientific hypothesis formation”, Artificial Intelligence, Volume 61, Issue 2, Pages 209-261.

Matthies, M., Giupponi, C. and Ostendorf, B., (2007), “Environmental decision support systems: Current issues, methods and tools”, Environmental Modelling and Software, Volume 22, Issue 2, Pages 123-127.

McPherson, S., (2000), “JavaServer Pages: A Developer's Perspective”, [online], Available at http://java.sun.com/developer/technicalArticles/Programming/jsp/, [Accessed in March, 2009].

Moradkhani, H., Hsu, K.L., Gupta, H.V. and Sorooshian S., (2004), “Improved streamflow forecasting using self-organizing radial basis function artificial neural networks”, Journal of Hydrology, Volume 295, Issues 1-4, Pages 246-262.

NCSL, (1995), “Determining and Reporting Measurement Uncertainties, Recommended Practice RP-12”, National Conference of Standards Laboratories, USA.

Nordtest, (1995), “Calibration, Traceability and Uncertainty”, NT Techn Report 305, Nordtest, Finland.

NPL, (1994), “NIS 80 - Guide to the expression of uncertainties in testing”, Edition 1, NAMAS Executive, National Physical Laboratory, UK.

O'Reilly, T., (2005), “What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software”, [online], Available at http://oreilly.com/web2/archive/what-is-web-20.html, [Accessed in January, 2009].

Power, D.J., (1997), “What is a DSS?”, The On-Line Executive Journal for Data-Intensive Decision Support, Vol.1, No. 3.

Power, D.J., (2007), “A Brief History of Decision Support Systems”, [online], Available at: http://DSSResources.COM/history/dsshistory.html, [Accessed in February, 2009].

Power, D.J., (2002), Decision Support Systems: Concepts and Resources for Managers, Westport, CT: Greenwood/Quorum Books, ISBN: 156720497X.

Power, D.J., (2006), "What are the advantages and disadvantages of computerized decision support?", DSS News, Vol. 7, No. 24.

Power, D.J., (2000a), "Web-based and model-driven decision support systems: concepts and issues", Proceedings of AMCIS 2000, Americas Conference on Information Systems, Long Beach, CA.

Power, D.J., (2000b), "Decision Support Systems Hyperbook", [online], Available at: http://dssresources.com/subscriber/password/dssbookhypertext, [Accessed in February, 2009].

Power, D.J. and Kaparthi, S., (1998), "The Changing Technological Context of Decision Support Systems", Context-Sensitive Decision Support Systems, London: Chapman and Hall, Pages 41 – 54.

Ramachandra, T.V., Krishna, S.V. and Shruthi, B.V., (2005), "Decision support system for regional domestic energy planning", Journal of Scientific and Industrial Research, Volume 64, Issue 3, pp 163-174 .

Rasmussen, S.N., (1999), "Tools for the Test Laboratory to Implement Measurement Uncertainty Budgets", NT Techn Report 430, Nordtest, Finland.

Reenskaug, T., (2003), The Model-View-Controller (MVC) Its Past and Present, [online], Available at: http://folk.uio.no/trygver/2003/javazone-jaoo/MVC_pattern.pdf, [Accessed in January, 2009].

Robert, C. P. and Casella, G., (2004), Monte Carlo Statistical Methods, 2nd edition, New York, Springer, ISBN 0387212396.

Rubinstein, R. Y. and Kroese, D. P., (2008), Simulation and the Monte Carlo Method, 2nd edition. New York, John Wiley & Sons, ISBN 9780470177938.

Sage. A.P., (1991), Decision Support Systems Engineering, John Wiley & Sons, Inc., New York, ISBN: 978-0471530008.

Shang, Y., (2005), "Expert Systems", The Electrical Engineering Handbook, Pages 367-377.

Sprague, R. H., Jr. and Carlson, E.D.,(1982), Building Effective Decision Support Systems. Englewood Cliffs, N.J.: Prentice-Hall.

Steele, O., (2004), "Web MVC", [online], Available at http://osteele.com/archives/2004/08/web-mvc, [Accessed in January, 2009].

Steffen, B., Kallio, H. and Þorsteinsson, F., (1994), "Traceable Calibration and Uncertainty of Measurements and Tests", NT Techn Report 251, Nordtest, Finland.

Sun Microsystems, (1999), JavaServer Page Technology White Paper, [online], Available at http://java.sun.com/products/jsp/pdf/jspguide-wp.pdf, [Accessed in February, 2009].

Taylor, B.N. and Kuyatt, C.E., (1994), "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results", NIST Technical Note 1297, National Institute of Standards and Technology, Gaithersburg, USA.

UKAS, (1997), "The Expression of Uncertainty and Confidence in Measurement", M 3003, United Kingdom Accreditation Service, UK.

Wang, Q., (2007), "Artificial neural networks as cost engineering methods in a collaborative manufacturing environment", International Journal of Production Economics, Volume 109, Issues 1-2, Pages 53-64.

Wei, P., Yang, Q.P., Salleh, M.R. and Jones, B.E., (2007), "Symbolic Computation for

Evaluation of Measurement Uncertainty," Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE, Warsaw, Poland, pp.1 - 4.

Weise, K. and Zhang, H., (1997), "Uncertainty Treatment in Monte Carlo Simulation", Journal of Physics A: Mathematical and General, 30 (17), pp. 5971-5980.

Weise, K. and Wöger, W., (1992), "A Bayesian theory of measurement uncertainty", Measurement Science and Technology, 3, pp1–11.

Xie, S.Q., (2006), "A decision support system for rapid one-of-a-kind product development", International Journal of Advanced Manufacturing Technology, Volume 28, Issue 7-8, Pages 643-652.

Yang Q. and Butler C. (1998), "Object-oriented model of measurement systems," IEEE Transactions on Instrumentation and Measurement, Volume 47, Pages 104–107.

Yang Q., (2005), "A Novel Representation of Energy and Signal Transformation in Measurement Systems", IEEE Transactions on Instrumentation and Measurement, Volume. 54, Pages 432-435.

Zemel'man, M. A.,(1985) "Classifying measurement errors", Measurement Techniques, Volume 28, Pages 471-475.

Zhou, B.H., Wang, S.J. and Xi, L.F., (2008), "Agent-based decision support system for dynamic scheduling of a flexible manufacturing system", International Journal of Computer Applications in Technology, Volume 32, Issue 1, Pages 47-62.

Zopounidis, C., Doumpos, M. and Matsatsinis, N.F., (1997), "On the use of knowledge-based decision support systems in financial management: A survey", Decision Support Systems, Volume 20, Issue 3, Pages 259-277.

Appendices

Appendix I

Publications Resulting from the Research

1. Wei, P., Yang, Q.P., Salleh, M.R., Jones, B.E., (2007), "Symbolic computation for evaluation of measurement uncertainty", IEEE Instrumentation and Measurement Technology Conference, May 2007, Warsaw, pp50.

2. Wei, P., Yang, Q.P., Hu, Y., Li,Y., (2009), "Web-Based Evaluation of Measurement Uncertainty Using Symbolic Computation", IEEE Instrumentation and Measurement Technology Conference, May 2009, Singapore.

3. Wei, P., (2009), "Web-based Knowledge-Based Decision Support System for Evaluation and Expression for Measurement Uncertainty", Proceedings of the IEEE (In press).

4. Salleh, M.R., Yang, Q.P., Wei, P., Jones, B.E., (2007), "Experimental validation of FEA modelling of touch trigger probes". IEEE Instrumentation and Measurement Technology Conference, May 2007, Warsaw, pp36.

5. Hu, Y, Yang, Q.P., Wei, P., (2009), "Development of a novel virtual coordinate measuring machine", I2MTC '09. IEEE Intrumentation and Measurement Technology Conference. Singapore, 5-7 May 2009. pp. 230 - 233.

Symbolic Computation for Evaluation of Measurement Uncertainty

Peng Wei, Qing Ping Yang, Barry Jones
School of Engineering and Design, Brunel University, United Kingdom
Tel: 44 1895 265854, Fax: 44 1895 269763, Email: empgppw@brunel.ac.uk

Abstract: In recent year, with the rapid development of symbolic computation, the integration of symbolic and numeric methods is applied in the industry world increasingly widely. [1] The objective of this paper is to discuss the evaluation of measurement uncertainty using symbolic computation, which is easy to understand and manipulate, and the application of symbolic computation proved feasible and rewarding. This approach also may give new directions to evaluation of measurement uncertainty toward network calculation, mobile remote calculation and local hardware calculation.

Keywords: Uncertainty of Measurement, Evaluation of Measurement Uncertainty, Symbolic Computation, Numeric Computation.

1. INTRODUCTION

Since the publication of the ISO Guide to the Expression of Uncertainty in Measurement (GUM) in 1993 [2], the importance of measurement uncertainty and its evaluation has been increasingly recognized. According to the GUM, the mathematical model for the measurand concerned plays a key role in the evaluation of measurement uncertainty. In deed, the definition of sensitivity coefficient of each uncertainty source is based upon partial derivative, which generally implies the use of mathematical models.

To facilitate the evaluation and automate the computation, several software tools have become commercially available to assist practical evaluations [3]. Most of these software tools are based on pure numerical methods using traditional programming languages (e.g. C/C++) or spreadsheets (e.g. Microsoft Excel) [3-4]. Despite its overwhelming popularity for scientific computing, pure numerical methods have some considerable limitations in uncertainty evaluation.

The major limitations of pure numeric computation in uncertainty evaluation include the following:

1) The mathematical model is the starting point and the key to the uncertainty evaluation. It is highly desirable to allow the use of any common symbols or user-defined symbols directly in the representation of the mathematical model. Uncertainty evaluation using pure numeric computation often limits the use of symbols or even only allows the symbols predefined by the software.

2) This also applies to the intermediate calculations of individual standard uncertainty. The use of user defined symbols for intermediate calculations is not allowed or limited in the most of the existing uncertainty evaluation tools.

3) The numerical computation of sensitivity coefficients usually calculates the partial derivatives according to the definition, and the approximate result is thus sensitive to the algorithms

and rounding errors. Further it requires that the values of all the variables are already known, the computation will have to be repeated if any of the values changes.

4) The results of the calculation can be directly affected by errors in the previous steps, which can trigger a chain reaction for the following steps [5].

5) Numeric computation also makes it less obvious to see the influences and contribution of various factors to the final measurement uncertainty.

The above mentioned limitations of pure numeric computation tend to make uncertainty evaluation appear more difficult and less user friendly. The direct evidence of these limitations can be easily seen from the limited capacity for expressing and processing the mathematical models in the existing software tools. They are also likely to impede the wide application of uncertainty evaluation.

The main difficulty lies in that the mathematical model for the measurement varies in each case, and that it has to be defined by the user at the run time and can not, in general, be predefined in the software.

2. PROPOSED APPROACH

This paper proposes and discusses the integration of symbolic computation and numeric computation in uncertainty evaluation.

Symbolic computation as a completely different analytical method has been increasingly applied as an alternative computing tool [6]. Compared with numeric computation, symbolic computation represents and manipulates information in symbolic form, i.e. it directly processes the symbols rather than the values of all the variables in the calculation.

Symbolic computation aims at the automation of mathematical calculation and is able to overcome most of the above limitations associated with numeric computation with the following advantages:

1) It simplifies mathematical calculations and makes it natural to use mathematical expressions and models.

2) It puts more emphasis on the analytical process than pure numerical methods, making the logic easily understood. [5]

3) This computation approach requires as little input data as possible.

More specifically, the authors proposed the use of symbolic computation for the measurement mathematical model and its subsequent processing. The symbolic representation of the mathematical model is crucial since both the user and the software can readily understand. Common mathematical functions and expressions can be directly entered and understood by the symbolic computation engine.

The calculation of sensitivity coefficients is a very important step in measurement uncertainty evaluation. They are defined as partial derivatives with respect to individual input variables. Their derivation from mathematical models may not be very straightforward for many users. Symbolic computing can generate the sensitivity coefficients automatically, thus significantly simplify the evaluation process.

The uncertainty evaluation will have to generate numerical values, but symbolic computation can minimize the use of numeric calculations, usually only in the final step. The intermediate steps can all be calculated by symbolic computation, which could eliminate possible calculation errors in the intermediate steps.

Ultimately, the use of symbolic computation will make the uncertainty evaluation easier and more user-friendly, facilitating its use in more applications.

3. RESULTS

Symbolic computation was implemented using the mathematical software, Maple 10. The feasibility of performing evaluation of measurement uncertainty using symbolic computation has been studied through examples, initially created as Maple worksheets.

The worksheet is composed of three main sections, as illustrated in Fig. 1, which shows an example concerned with DC current measurement [7].

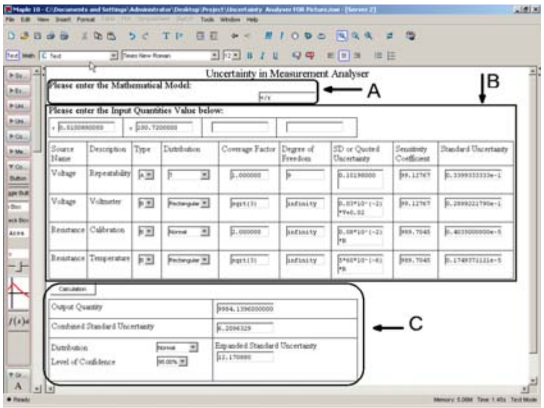


Fig. 1 Worksheet Main Interface

Section A

This step is used for inputting the mathematical model. The user is required to enter the mathematical model in the 'Mathematical Expression Component' according to the measurement process concerned. Variables can be represented by any symbol, either commonly defined or user defined. Symbols for standard operations can be directly used.

In this DC current measurement case, the current is function of the voltage and resistance. According to Ohm's law,

$$I = f(x_1, x_2) = f(V, R) = \frac{V}{R} \qquad (1)$$

I is the current, V is the voltage, and R is the resistance. Although the user is normally required to type in $\frac{V}{R}$ into the Fig. 2 , thanks to the symbolic computation, the user can define the symbols of the formula, such as $\frac{A}{B}$, $\frac{x}{y}$ as the user prefer.

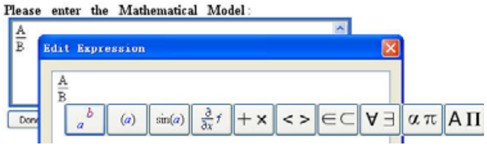


Fig. 2 Establish Mathematical Model

Maple 10 provides three commands which can be programmed to extract the symbols [8] :

- indets - find indeterminates of an expression
- op - extract operands from an expression
- nops - number of operands of an expression

Using the commands above, input quantities are extracted from the expression input by the user. If the user inputs the expression $\frac{A}{B}$, i.e. $f(A,B)=\frac{A}{B}$, input quantities A and B are extracted by using 'indets' command. In the meantime, in the background process, 'Diff' command is used to compute the partial derivative of the function $\frac{A}{B}$ with respect to A.

$$diff\,(f\,(A,B),A)\rightarrow\frac{\partial}{\partial A}f\,(A,B)\rightarrow\frac{\partial}{\partial A}(\frac{A}{B})\rightarrow\frac{1}{B} \qquad (2)$$

Sensitivity coefficient related to input quantity A is determined. In the same way, sensitivity coefficient related to input quantity B is determined.

$$diff\,(f\,(A,B),B)\rightarrow\frac{\partial}{\partial B}f\,(A,B)\rightarrow\frac{\partial}{\partial B}(\frac{A}{B})\rightarrow-\frac{A}{B^2} \qquad (3)$$

In the environment provided by Maple 10, in accordance with the basic principle of symbolic computation, the symbols are directly involved in the calculation to realize the analysis of all the variables and constants in the user defined formula and their mathematical relations. Thus, $\frac{1}{B}$ and $-\frac{A}{B^2}$ are involved in the following calculation in forms of symbols.

In more complex mathematical models, like Torque Tester Calibration [7], the mathematical model is

$$T_m = M\cdot\left(1-\frac{\rho_a}{\rho_m}\right)\cdot g\cdot d\ \cdot(1+\alpha\cdot\theta)+e \qquad (4)$$

The user can use simple symbols which is easier to input, as demonstrated below

$$A\cdot\left(1-\frac{B}{C}\right)\cdot D\cdot E\cdot(1+F\cdot G)+H \qquad (5)$$

Through symbolic computation, all the partial derivatives of the function (5) with respect to each symbol are determined. For example, the sensitivity coefficient related to input quantity M is

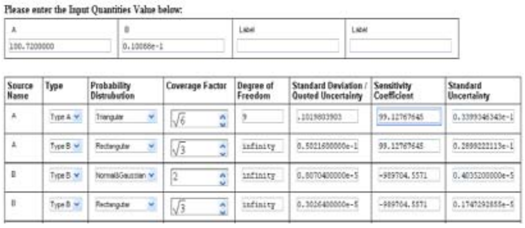
$$
\begin{aligned}
& diff\,(Tm, M) \\
& \rightarrow \frac{\partial T_m}{\partial M} \\
& \rightarrow \frac{\partial}{\partial M}(M \cdot \left(1 - \frac{\rho_a}{\rho_m}\right) \cdot g \cdot d \cdot (1 + \alpha \cdot \theta) + e) \\
& \rightarrow \left(1 - \frac{\rho_a}{\rho_m}\right) \cdot g \cdot d \cdot (1 + \alpha \cdot \theta)
\end{aligned}
\tag{6}
$$

All the processes above based on the principles of symbolic computation are completed automatically by the programme. This unburdens the user of deducing the mathematical formulas, and in the meantime minimizes the involvement of numeric computation in the early stages of the calculation, which is a disadvantage of numeric computation.

Section B

This step collects the information about each individual source of uncertainty, including type of evaluation, probability distribution, coverage factor, degree of freedom, standard deviation and quoted uncertainty. The user can choose different options and enter numeric values or expressions using symbols. The Sensitivity Coefficients and standard uncertainty in the last column can be calculated separately for each uncertainty source or together with the final evaluation results in section C.

In this DC current measurement case, Type in the measurement data into the table of Fig. 3:



Please enter the Input Quantities Value below:

A	B	Label	Label
100.7200000	0.10068e-1		

Source Name	Type	Probability Distrubution	Coverage Factor	Degree of Freedom	Standard Deviation / Quoted Uncertainty	Sensitivity Coefficient	Standard Uncertainty
A	Type A	Triangular	$\sqrt{6}$	9	.1019803903	99.12767645	0.3399346343e-1
A	Type B	Rectangular	$\sqrt{3}$	infinity	0.5021600000e-1	99.12767645	0.2899222113e-1
B	Type B	Normal&Gaussian	2	infinity	0.8070400000e-5	-989704.5571	0.4035200000e-5
B	Type B	Rectangular	$\sqrt{3}$	infinity	0.3026400000e-5	-989704.5571	0.1747292855e-5

Fig. 3 Input Quantities

A). Enter the value of the input quantities. As for multiple input values , e.g. {100.68, 100.83, 100.79, 100.64, 100.63, 100.94, 100.60, 100.68, 100.76, 100.65} which are the measurement results of A, the mean {100.72} will display in box A. Moreover, the Standard Deviation of data will be automatically calculated by command ‘stats[describe, standarddeviation]’;

B). According to the property of each source, choose the Uncertainty Types and Probability Distrubution, relevant Coverage Factor and Degree of Freedom will be calculated;

C). Type ‘0.03%*A+0.02’ into the quoted uncertainty blank of source Voltage-Voltmeter. Type ‘0.08%*B’ into the quoted uncertainty blank of source Resistance-Calibration. Type ‘60*10^(-6)*5*B’ into the quoted uncertainty blank of source Resistance-Temperature.

The difference between the numeric computing and the symbolic computing is clearly seen in this step. When using numeric computing, the user needs to calculate the value of A and B according to the measurement record and then execute the expression involved. When symbolic computing is applied, the value of A and B will display after entering the measurement results. Both A and B are the symbol that can be considered as the conveyer, the numeric represented by which will be calculated in the final step formula.

Through symbolic computing, the use of symbols and related mathematical formulas is maximized and the procedure is simplified and the automatization of computer calculation is realized, which enables the whole process to be more significant in terms of experiment and mathematics.

Section C

This step displays the results of the evaluation, including the mean value of the output quantity or the measurand, combined standard uncertainty and the expanded uncertainty. The user can also choose the level of confidence and distribution for the combined standard uncertainty. The results can be updated by clicking the Calculation button.

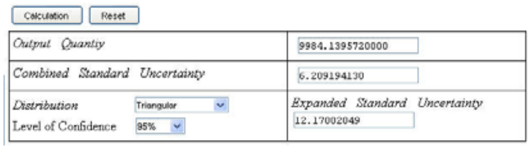


Fig.4 Final Report

In this DC current measurement case, according to the definition of the Combined Standard Uncertainty which is related to the uncorrelated input quantities(7),

$$u_C{}^2(y) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j) \quad (7)$$

the background process will calculate the combined standard uncertainty, and all symbols in Section A and section B are involved in this final formula to calculate the combined standard uncertainty which is:

$$u_c^2\left(\frac{A}{B}\right) = \left(\frac{\partial}{\partial A}\left(\frac{A}{B}\right)\right)^2 u_1^2(A) + \left(\frac{\partial}{\partial A}\left(\frac{A}{B}\right)\right)^2 u_2^2(A) + \left(\frac{\partial}{\partial B}\left(\frac{A}{B}\right)\right)^2 u_3^2(B) + \left(\frac{\partial}{\partial B}\left(\frac{A}{B}\right)\right)^2 u_4^2(B) \quad (8)$$

$$= \left(\frac{1}{B}\right)^2 u_1^2(A) + \left(\frac{1}{B}\right)^2 u_2^2(A) + \left(-\frac{A}{B^2}\right)^2 u_3^2(B) + \left(-\frac{A}{B^2}\right)^2 u_4^2(B)$$

The results based upon several examples have shown the great power of symbolic computation. Expressing the mathematical model symbolically, it is also capable of processing the model (e.g. calculating the sensitivity coefficients) both symbolically and numerically. Its great potential has been demonstrated for the evaluation of measurement uncertainty.

4. CONCLUSION

Through the worksheet created in Maple, it is clear that symbolic computation, together with the conventional numeric computation, makes evaluation of measurement uncertainty easier for general users, in terms of the deducing of the formulas and the numeric calculation. Just a few simple steps are required to get the final report. The user-friendly nature of this analytical method will highly improve the use and promotion of uncertainty evaluation.

This paper discusses symbolic computation in evaluation of measurement uncertainty through Maple's powerful mathematical computing engine. If JAVA, which is capable of symbolic computing through programme and is highly compatible of network extension, hardware and software, is used as the platform of uncertainty evaluation [9], an outlet of uncertainty evaluation to the network can be realised. It would not be just imagination that a report of evaluation of measurement uncertainty can be sent to users' mobile phones. Even though for mobile phones which are not compatible of JAVA extension, a report may be available through a SMS massage.

Our further research direction is to extend the using of symbolic computation for evaluation of measurement uncertainty to other platform, such as Java.

REFERENCES

[1] Palancz B., "Symbolic computation in engineering", Proceedings of the Conference on Mechanical Engineering, Vol. 2, pp. 862-866, 1998.

[2] ISO, "Guide to the Expression of Uncertainty in Measurement (GUM)", Geneva, Switzerland, 1993. (Reprinted 1995).

[3] S. N. Rasmussen, Danish Technological Institute, "Software tools for the expression of uncertainty in measurement", Metro Trade Workshop on Traceability and Measurement Uncertainty in Testing. Berlin, January 30-31, 2003.

[4] B. D. Hall, "Uncertainty calculations using object-oriented computer modelling", Meas. Sci. Technol. **10** 380-386 doi:10.1088/0957-0233/10/5/307, 1999.

[5] I. Munteanu, D. Ioan, "Symbolic Computation with Maple V for Undergraduate Electromagnetics", IEEE Transactions on Education, Vol. 44, No. 2, pp. 3, 2001.

[6] A. Juozapavièius, "Symbolic Computation: Systems and Applications", Nonlinear Analysis: Modelling and Control, Vilnius, IMI, Vol 3, 1998.

[7] Singapore Institute of Standards and Industrial Research, "Guidelines on the Evaluation and Expression of the Measurement Uncertainty", pp. 45-54, 1995.

[8] L. Bernardin, B. Char and E. Kaltofen, "Symbolic Computation in Java: an Appraisement". Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation, pp. 237-44, 1999.

Web-Based Evaluation of Measurement Uncertainty Using Symbolic Computation

Peng Wei, Qing Ping Yang, Yang Hu, Li Yi
School of Engineering and Design
Brunel University
London, United Kingdom
empgppw@brunel.ac.uk

Abstract—This paper proposes a web-based approach for evaluation of measurement uncertainty over the GUM uncertainty framework, focusing on the developing process of symbolic computation. The architecture of the web-based approach and procedure will be presented. The great potential and powerful features of this approach for measurement uncertainty evaluation is demonstrated through a measurement example.

Keywords- Uncertainty of Measurement; Evaluation of Measurement Uncertainty; Web-based Application; Symbolic Computation; JSP

1. Introduction

The importance of measurement uncertainty and its evaluation has been increasingly recognized since the publication of the ISO Guide to the Expression of Uncertainty in Measurement (GUM) in 1993 [1]. It has since become a key document and guide for evaluation of measurement uncertainty.

Several software tools have become commercially available to facilitate the evaluation and automate the computation, e.g. Uncertainty Calculator, DFM-GUM and Timeko Uncertainty [2]. In the GUM framework, the mathematical model of a measurement process is of critical importance in uncertainty evaluation. Most of the existing software tools have rather limited ability to handle mathematical models, although some of them (e.g. Timeko Uncertainty) have enhanced the ability to enter the measurement model.

Secondly, all the existing software tools rely upon numerical computation. The authors have discussed the limitations of numeric computation in evaluation of measurement uncertainty in [3]. For those users who are familiar with the measurement model, it is of critical significance if they can establish the mathematical expression in the software tool without too many restrictions, regardless of complexity of the model or how the model is entered.

A worksheet using symbolic computation (Maple) has been developed by the authors to demonstrate its use for uncertainty evaluation, which overcomes most of the limitations associated with numeric computation [3]. The GUM users can directly type any valid mathematical expression into the worksheet, using standard symbols based on

Math Markup Language (MathML) 2.0 [4]. However, some Maple functions and components (e.g. Form, Display format) need to be predefined by the programmer as well. Thus, dynamic worksheets can not be created and the interactions between the user and the worksheet are limited in Maple environment.

Thirdly, the existing software tools for uncertainty evaluation are traditional desktop applications with limitations in cross-platform compatibility (i.e., Windows, Mac, Linux etc.), accessability, cost-effectiveness and ease of system maintenance. This paper proposes and discusses a web-based approach that is able to overcome the above limitations and has symbolic computation capability for evaluation of measurement uncertainty.

2. PROPOSED APPROACH

The proposed methodology combines the JSP web content technologies based on the Apache Struts [5], an open source web application framework that implements the Model-View-Controller (MVC) design pattern, with the free and open source software (FOSS) Java Symbolic Computing Library and Mathematical Editor (JSCL-Meditor) [6].

The Internet and web technologies have been growing rapidly ever since its creation. The Internet is not only a media platform for information sharing, but also offers a ubiquitous platform where complex software tools are networked. A number of web-based applications have been reported in the literature [7] and the web based approach are becoming increasingly popular because they offer a large number of advantages over traditional desktop software [8, 9]:

Web-based applications only rely on a common web browser to render the application executable. The cross-platform compatibility is therefore extended.

Due to the ubiquity of web browser and the convenience of using a web browser as a thin client [10], multiple concurrent users worldwide can access and execute the applications immediately and universally through the Internet.

Requirements on the thin client's side are minimum because the application only needs to be installed on the server. System maintenance and update can all be performed on the server without distributing and installing software on potentially thousands of client computers. The cost is also been reduced dramatically.

The relevant content and system can be easily updated to suit users' requirements to provide dynamic information, it thus creates a more interactive experience. In the mean time, online storage and sharing of data are also available for advanced data management, project management and computer aided design.

Java Server Pages (JSP) has been used to create a dynamically generated web site to evaluate the measurement uncertainty and perform its subsequent processing. In order to achieve a modular design involving database codes, page design codes and navigational codes, a free open-source Apache Struts framework [5] (Fig.1) has been adopted, which was designed to help developers create Java web applications that utilize a Model-View-Controller (MVC) architecture[11].

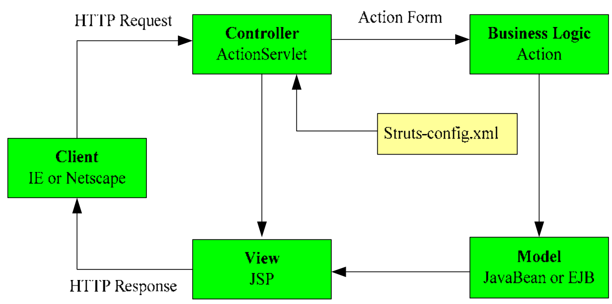


Fig.1 Apache Struts Architecture Pattern

In the mean time, JSCL-Meditor (Java symbolic computing library and a mathematical editor) has been selected for symbolic evaluation of measurement uncertainty, due to its excellent portability across different computer platforms,

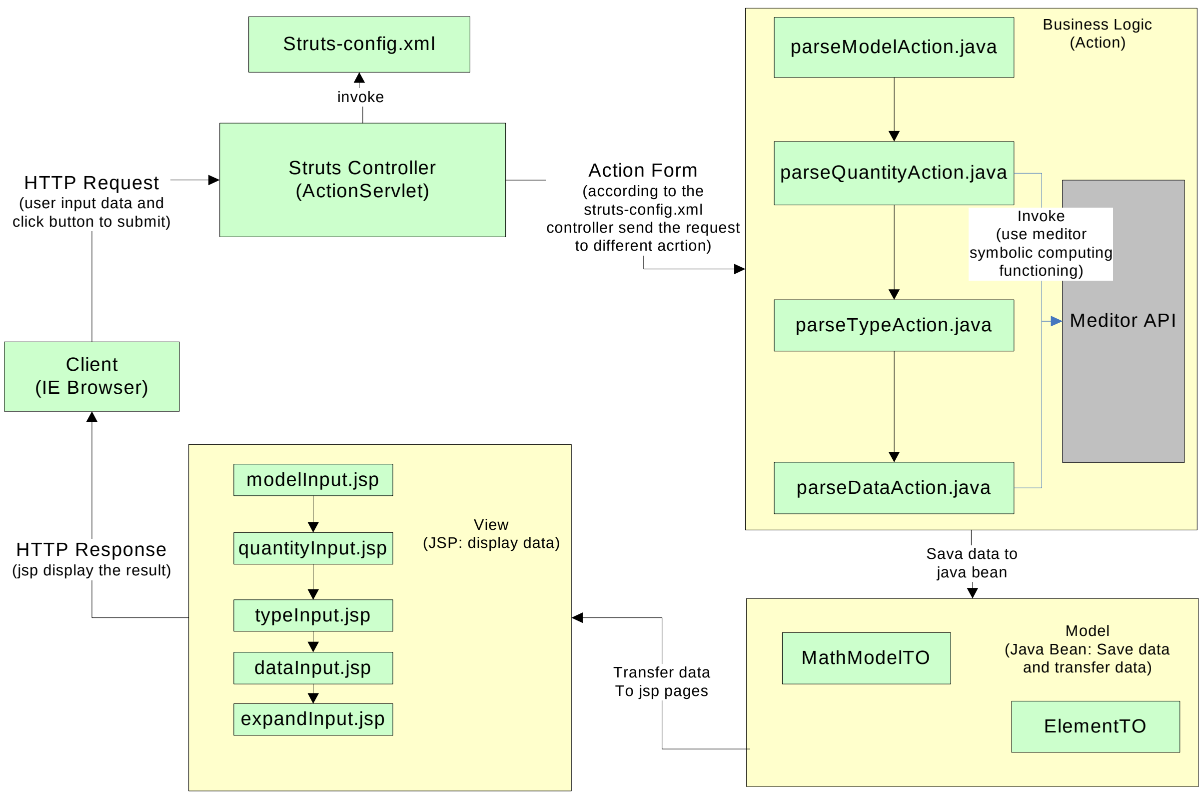


Fig.2 Application Architecture

symbolic capability and the readability [6, 12]. JSCL-Meditor is the most critical computation engine in Java web application and the free open source symbolic package ever written for Java. Once JSCL-Meditor is embedded as the business logic in this web-based application, common mathematical functions and expressions directly entered on the web view can be understood by the symbolic computing library model. The library model can automatically update and/or generate additional views and results, e.g. the sensitivity coefficients, thus significantly simplifying the evaluation process.

3. IMPLEMENTION

The Struts was downloaded from the Internet and then extracted to the Webapp folder of Tomcat5.0 [13], where system related files and folders were also generated. The user could then modify the definitions of the Action Mapping and the Path in the struts configuration. These procedures should be followed to complete the framework of the web application.

Using Apache Struts, the business logic is based on the GUM framework, the view is the JSP/Presentation (also know as User Interface/UI) and the controller is the code that gathers dynamic data, as shown in (Fig. 2).

The JSP page is mainly used to show the value returning from the business logic. QuantityInput.jsp shows the page where the user inputs the source quanitites; typeInput.jsp shows the page where the user chooses the measurement type; dataInput.jsp shows the page where the user chooses the distribution and inputs the quoted uncetainty; expandInput.jsp shows the page of standard uncertainty and combined uncertainty.

All the HTTP requests from the user are submitted to the DoServiceAction Class to be processed. DoServiceAction Class encapsulates the user requests in the StepTo Class and calls the DoServiceCmd Class to submit the content to the rule engine for processing.

In the system, once the user completes inputting the quantities and presses the 'next' button, the system will run ParseQuantityAction and ParseQuantityCmd to conduct the derivative computation of the input formulas and run the input values in the formulas.

Designs by using Transfer Object (TO) data structure(Fig.3), 'JSP', 'Action', 'Command' are separated and the most efficient algorithm is realized during the implementation. Via the jsp page, the user submits the date to Action, which processes the logic using command. Command stores the processed data in the transfer object(TO) and returns that to Action, which evaluates whether the processing is successful, and then goes to the next jsp page according to the struts config.xml.

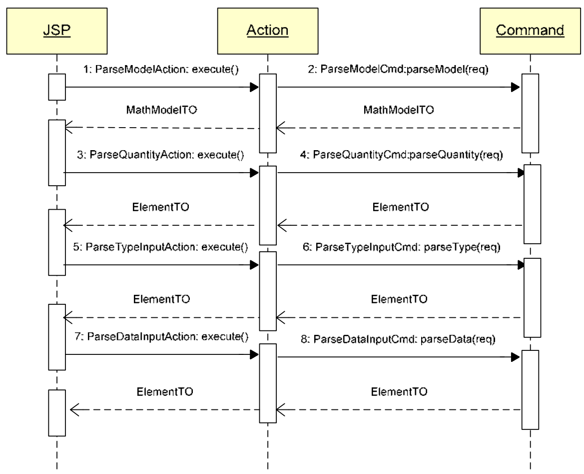
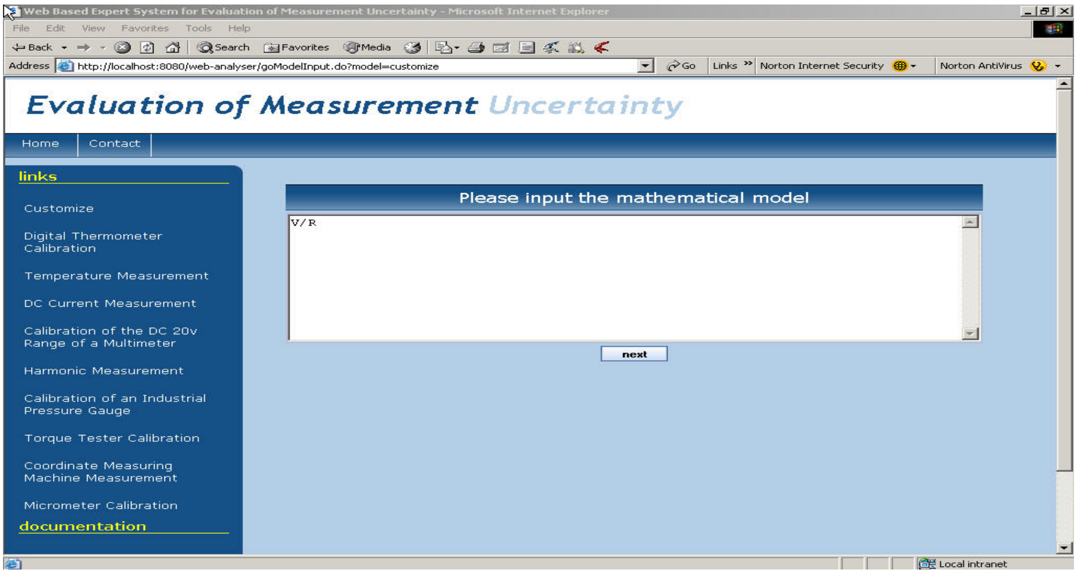


Fig.3 Transfer Object Data Structure

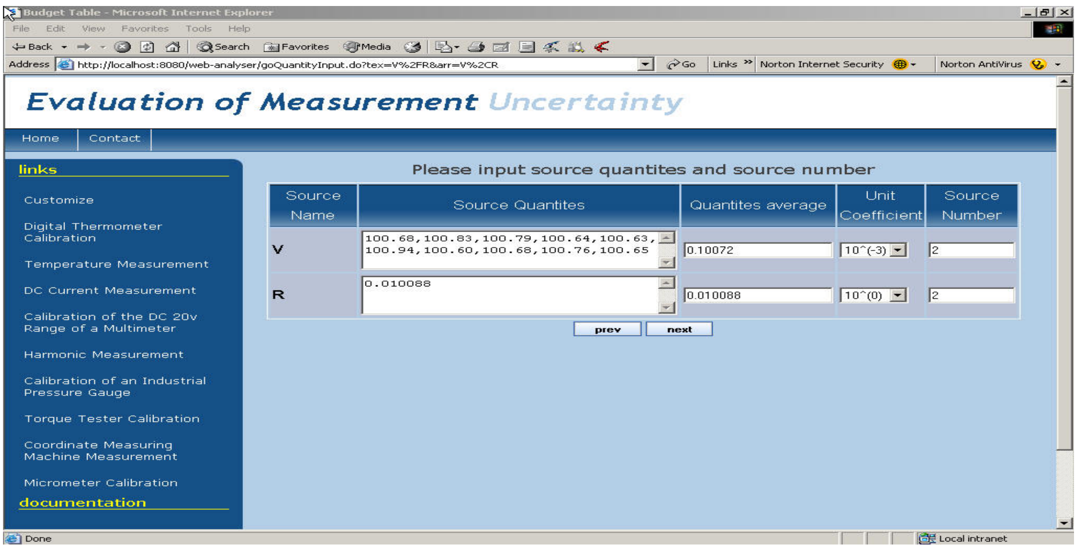
The GUM users interact with the User Interface to input the relevant measurement model and data. The controller handles the input event from the user interface and sends the request to the different Actions; the business logic will classify and deal with the different Actions and save the data to the JavaBean.

4. Result

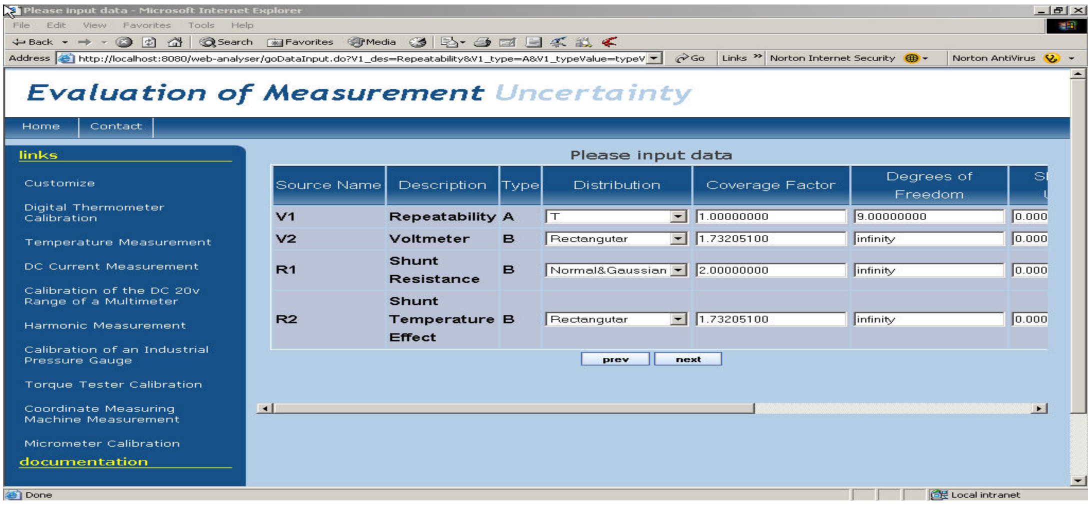
The JSP web-based evaluation of measurement uncertainty using symbolic computation was implemented with the servlet container Apache Tomcat 5.0 [13]. This web-based application is composed of several web pages, as illustrated in Fig.4, which shows an evaluation example concerning DC current measurement [14].



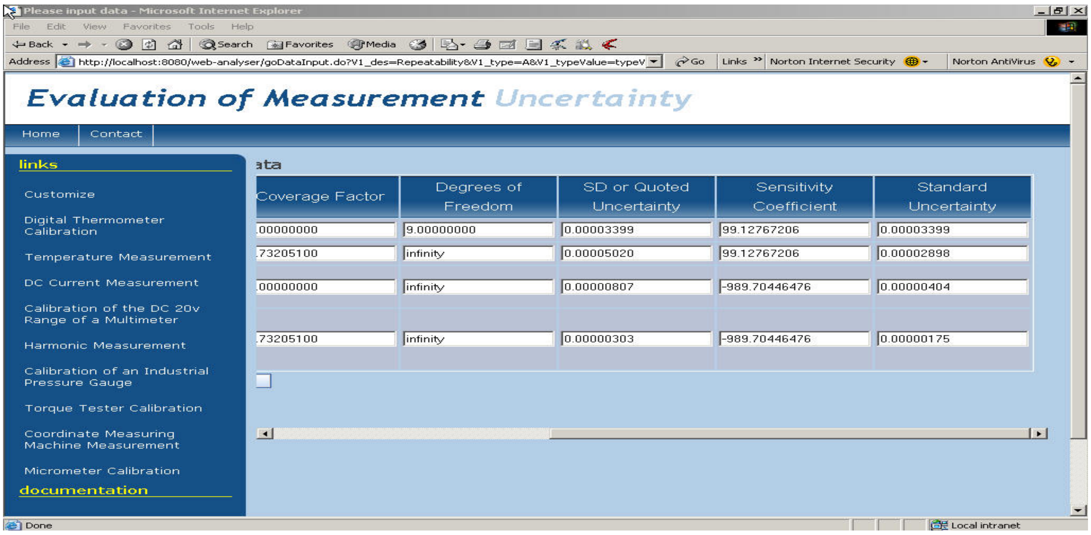
A)



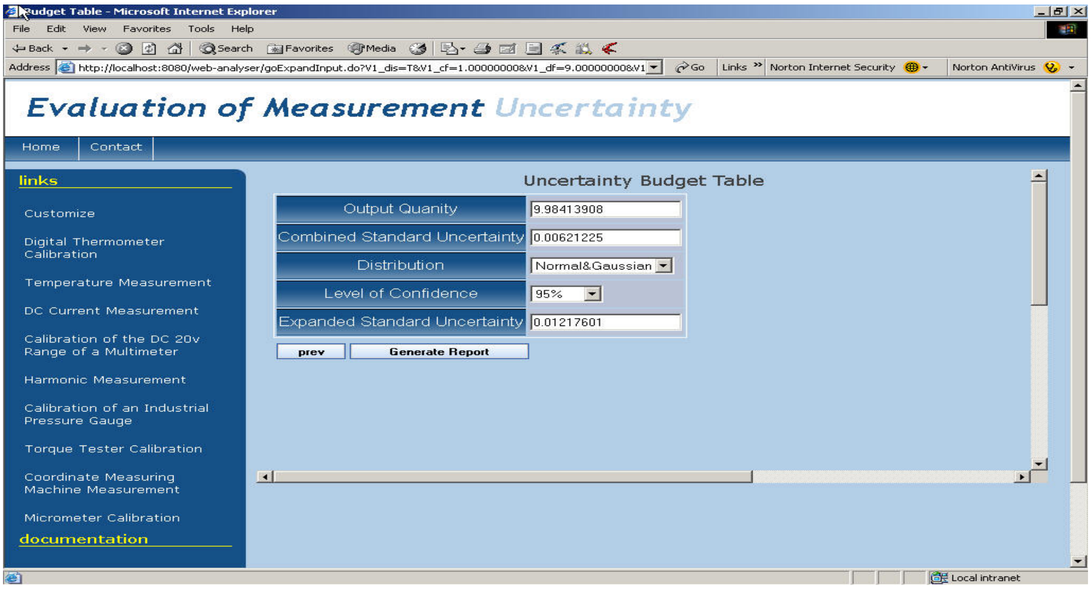
B)



C)



D)



E)

Fig.4 JSP WEB PAGES

Page A) is used for selecting or inputting the mathematical model. The end user can select the measurement case from some built-in mathematical models listed in left panel. The contents of the detailed cases can be continuously updated and completed. If the case is not listed, the end user is required to enter the mathematic expression in the text box according to the measurement process concerned. In the DC current measurement case, the measured current I is calculated according to Ohm's law,

$$I = f(V, R) = \frac{V}{R} \tag{1}$$

where V is the voltage, and R is the resistance. Variables can be represented by any symbol, either commonly defined or user defined. Symbols for standard operations can be directly used. Thus the user can directly type any valid mathematical expression, using standard symbols, e.g. $\frac{V}{R}$ or user defined symbols e.g. $\frac{X}{Y}$. Once the user click the 'NEXT' button in the JSP, JavaScript based on the thin client will parse the expression by separating the variable symbols and the math operator symbols and will submit them to the Server.

Pages B, C and D collect the information about each individual source of uncertainty, including type of evaluation, distribution, coverage factor, degree of freedom and quoted uncertainty. The user can choose different options and enter numeric values or expressions using symbols. The Sever will keep these values in a Transfer Object to achieve the transferring in the follow steps (Fig.5).

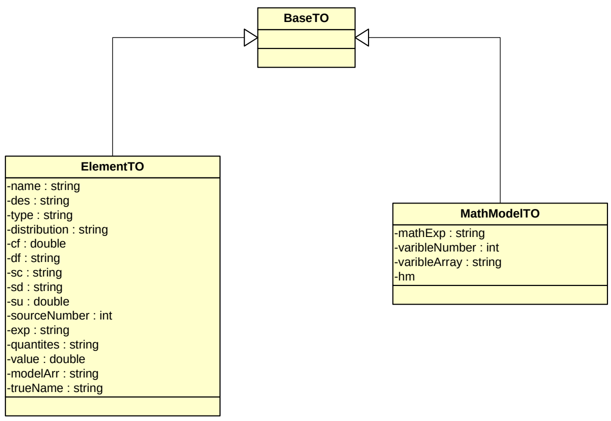


Fig.5 Transfer Object Relationship

In the mean time, JSCL-Meditor API [6] is called to compute the derivatives of the expression with the Java programming.

Beside this, JSCL-Meditor has also been used to compute other related math computation in the web application. Both the sensitivity coefficient and the standard uncertainty in the last column can be calculated separately for each uncertainty source.

Page E display the results of the evaluation, including the mean value of the output quantity or the measurand, combined standard uncertainty and the expanded uncertainty. The user can also choose the level of confidence and distribution for the combined standard uncertainty. The full uncertainty budget table can be subsequently produced.

5. Conclusions

The results based upon several examples have demonstrated the capability and potential of the web-based approach for evaluation of measurement uncertainty utilizing JSP and JSCL-Meditor. The end user is able to evaluate the measurement uncertainty via Internet anytime and anywhere with just a Web browser. It offers great performance in accessibility, usability, compatibility and interoperability.

Moreover, by using Java symbolic computing library, it is capable of processing the model symbolically, e.g. calculating the sensitivity coefficients. Its great potential has been demonstrated for the evaluation of measurement uncertainty.

References

[1] ISO, "Guide to the Expression of Uncertainty in Measurement (GUM)," Phil. Geneva, Switzerland, 1993. (Reprinted 1995)

[2] S. N. Rasmussen, "Software tools for the expression of uncertainty in measurement," In: Metro Trade Workshop on Traceability and Measurement Uncertainty in Testing, Berlin, 30/31 Jan 2003.

[3] P. Wei, Q. P. Yang, M. R. Salleh, B. E. Jones, "Symbolic Computation for Evaluation of Measurement Uncertainty," In: Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE, 1-3 May 2007, Warsaw, Poland, pp.1 – 4.

[4] MathML. 2001. [online]. [Accessed 2nd Mar 2009]. Available from World Wide Web < http://www.w3.org/Math/>.

[5] Doray, Beginning Apache Struts: From Novice to Professional, Apress, 2006, pp.37-p52.

[6] Jscl-Meditor: Java Symbolic Computing Library and Mathematical Editor. 2003. [online]. [Accessed 2nd Mar 2009]. Available from World Wide Web <http://jscl-meditor.sourceforge.net>.

[7] Ginige, S. Murugesan, "Guest Editors' Introduction: Web Engineering An Introduction," IEEE MultiMedia Volume 8, Issue 1. IEEE Computer Society Press. Los Alamitos, pp.14 – 18, 2001.

[8] P. Graham, Hackers & Painters: Big Ideas from the Computer Age, Oreilly & Associates, 2004, Inc, pp.56-86.

[9] S. L. Fowler, V. R. Stanwick, Web Application Design Handbook: Best Practices for Web-based Software. Morgan Kaufmann, 2004.

[10] J.T. Sinclair, M. S. Merkow, Thin Clients Clearly Explained, Morgan Kaufmann, 1999.

[11] C. S. Horstmann, G. Cornell, Core Java 2, Volume 1- Fundamentals，7th Edition. Prentice Hall PTR, 2004.

[12] L. Bernardin, B. Char, E. Kaltofen, “Symbolic Computation in Java: an Appraisement,” In: Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation, 1999, pp. 237-244.

[13] V. Chopra, A. Bakore, J. Eaves, B.Galbraith, C. Wiggers, S. Li, Professional Apache Tomcat 5. John Wiley and Sons, 2004, pp.29-43.

[14] Singapore Institute of Standards and Industrial Research, "Guidelines on the Evaluation and Expression of the Measurement Uncertainty", pp. 45-54, 1995.

Appendix II

Part of Programmes for GUM Knowledge Base

- quantityInput.jsp

```
<%for(int i=0;i<len;i++){%>
 <tr>
	<td class="field_value"><%=hm.get(x+i)%></td>
	<td class="field_value"><textarea name="<%=hm.get(x+i)%>" rows="3" cols="35" onblur="avg(this)"/></textarea></td>
	<td class="field_value"><input type="text" name="<%=hm.get(x+i)%>_avrg" readonly/></td>
	<td class="field_value">
	<select name="<%=hm.get(x+i)%>_av" onchange="checkUnit(this,'<%=hm.get(x+i)%>')">
	<option value="0.001">10^(-3)</option>
	<option value="0.01">10^(-2)</option>
	<option value="0.1">10^(-1)</option>
	<option value="1" selected>10^(0)</option>
	<option value="10">10^(1)</option>
	<option value="100">10^(2)</option>
	<option value="1000">10^(3)</option>
	</select>
   </td>
  <td class="field_value">
<input type="text" name="<%=hm.get(x+i)%>_s" onblur="checkNumber(this)" size="10" maxlength="2"/> </td>
<input type="hidden" name="<%=hm.get(x+i)%>_qua">
<input type="hidden" name="<%=hm.get(x+i)%>_snm">
<input type="hidden" name="<%=hm.get(x+i)%>_avr">
<input type="hidden" name="<%=hm.get(x+i)%>_qu">
</tr>
<%};%>
```

- typeInput.jsp

```
<%for(int i=0;i<vt.size();i++)
   {
     ElementTO to=(ElementTO)vt.get(i);%>
	<tr>
		<td class="field_value"><%=to.getName()%></td>
		<td class="field_value"><textarea name="<%=to.getName()%>_des" rows="4" cols="40"></textarea></td>
		<td class="field_value">
		<SELECT NAME="<%=to.getName()%>_type" id="<%=to.getName()%>_type_id" onchange="changeType(this,'<%=to.getName()%>')">
	<script>
	changeList.push(document.getElementById('<%=to.getName()%>_type_id'));
					nameList.push('<%=to.getName()%>');
	</script>
	<OPTION VALUE="A">A</option> <OPTION VALUE="B">B</option>
	</SELECT>
```

```
	</td>
	<td class="field_value">
	<div id="<%=to.getName()%>_A"><%=to.getDf()%>
	</div>
	<div id="<%=to.getName()%>" style="visibility:hidden;overflow:auto;z-index:6;">
	<input type="radio" name="<%=to.getName()%>_typeValue"
id="<%=to.getName()%>_typeValue" value="typeValue1"
onclick="checkRadio(this,'<%=to.getName()%>')" checked/>
	<SELECT NAME="<%=to.getName()%>_typeValue1">
		<OPTION VALUE="infinity" selected>infinity</option>
		<OPTION VALUE="50">50</option>
		<OPTION VALUE="12">12</option>
		<OPTION VALUE="8">8</option>
		<OPTION VALUE="6">6</option>
		<OPTION VALUE="3">3</option>
		<OPTION VALUE="2">2</option>
	</SELECT>
<br/>
<input type="radio" name="<%=to.getName()%>_typeValue"
id="<%=to.getName()%>_typeValue" value="typeValue2"
onclick="checkRadio(this,'<%=to.getName()%>')"/>
<input type="text" name="<%=to.getName()%>_typeValue2" onblur="checkNum(this)"/>
</div>
<script>
radioList.push(document.getElementById('<%=to.getName()%>_typeValue'));
</script>
	</td>
</tr>
<%}%>
```

- dataInput.jsp

```
<%for(int i=0;i<vt.size();i++)
   {
     ElementTO to=(ElementTO)vt.get(i);%>
   <tr>
<td
class="field_value"><%=to.getName()%>       &
nbsp;        </td>
<td class="field_value"><%=to.getDes()%></td>
<td class="field_value"><%=to.getType()%></td>
<td class="field_value">
<select name="<%=to.getName()%>_dis"
onchange="changeDis(this,'<%=to.getName()%>')">
		<option value="Triangular" selected>Triangular</option>
		<option value="T">T</option>
		<option value="Rectangutar">Rectangutar</option>
```

```
            <option value="U-Shape">U-Shape</option>
            <option value="Normal">Normal&Gaussian</option>
  </select>
</td>
<td class="field_value"><input type="text" name="<%=to.getName()%>_cf"
value="2.449489" readonly/></td>
<% if(to.getType().equals("A")){%>
    <td class="field_value"><input type="text" name="<%=to.getName()%>_df"
value="<%=to.getDf()%>" readonly/></td>
    <td class="field_value"><input type="text" name="<%=to.getName()%>_sd"
value="<%=to.getSd()%>"
onblur="repFunc('<%=to.getName()%>','<%=to.getValue()%>')"/></td>
     <td class="field_value"><input type="text" name="<%=to.getName()%>_sc"
value="<%=to.getSc()%>" readonly/></td>
     <td class="field_value"><input type="text" name="<%=to.getName()%>_su"
value="<%=to.getSu()%>" readonly/></td><%}else if(to.getType().equals("B")){%>
      <td class="field_value"><input type="text" name="<%=to.getName()%>_df"
value="<%=to.getDf()%>" readonly/></td>
      <td class="field_value"><input type="text" name="<%=to.getName()%>_sd" value=""
onblur="repFunc('<%=to.getName()%>','<%=to.getValue()%>')"/></td>
        <td class="field_value"><input type="text" name="<%=to.getName()%>_sc"
value="<%=to.getSc()%>" readonly/></td>
        <td class="field_value"><input type="text" name="<%=to.getName()%>_su"
value="" readonly/></td><%}%>
     </tr>
  <%}%>
```

- expandInput.jsp

```
<%for(int i=0;i<vt.size();i++)
{
   ElementTO to=(ElementTO)vt.get(i);
   String msd = new String(to.getSd());
   String msc = new String(to.getSc());%>
<tr>
   <td class="field_value"><PRE><%=to.getDes()%></PRE></td>
   <td class="field_value"><PRE><%=to.getType()%></PRE></td>
   <td class="field_value"><PRE><%=to.getName()%></PRE></td>
   <td class="field_value"><PRE><%=msd%></PRE></td>
    <td class="field_value"><PRE><%=msc%></PRE></td>
    <td class="field_value"><PRE><%=to.getDistribution()%></PRE></td>
    <td class="field_value"><PRE><%=to.getCf()%></PRE></td>
    <td class="field_value"><PRE><%=to.getSu()%></PRE></td>
    <td class="field_value"><PRE><%=to.getDf()%></PRE></td>
</tr>
<%}%>
```

```
// Calculate Sensitivity Coefficient
try{
          ..........................................
Generic ge = Expression.valueOf(modelExp);
Generic rt = ge.expand();
rt = rt.simplify();
c = rt.toString();
          .......................................
}
// Calculate standard uncertainty
public double getSU(String sd,String df){
    return Double.parseDouble(sd)/Math.sqrt(Double.parseDouble(df)+1);
}
// Calculate degrees of freedom
public String getDof(String quantity){
String arr[] = quantity.split(",");
    if(0==arr.length-1){
        return String.valueOf(1);
    }else{
        return String.valueOf(arr.length-1);
    }
}
//Calculate SD
public String getSD(String quantity){
    String arr[] = quantity.split(",");
    int len = arr.length;
    if(len-1 == 0){
        len = 2;
    }
    double sum=0;
    double avg = getAverage(quantity);
    for(int i=0;i<arr.length;i++){
sum = sum +
(Double.parseDouble(arr[i])-avg)*(Double.parseDouble(arr[i])-avg);
    }
    double sd = Math.sqrt(sum/(len-1));
    sd = sd/Math.sqrt(len);
    return String.valueOf(sd);
}
    ..........................................................
//Difine transfer object list
Vector vt = new Vector();
int sn = Integer.parseInt(sourceNumberStr);
for(int j=1;j<=sn;j++){
    String vName = arr[i]+j;
    String trueName = arr[i];
```

```
//Difine transfer object.
    ElementTO eto = new ElementTO();
    eto.setName(vName);
    eto.setModelArr(modelArr);
    eto.setSourceNumber(Integer.parseInt(sourceNumberStr));
    eto.setQuantites(quantity);
    eto.setDf(getDof(quantity));
    eto.setSd(getSD(quantity));
    eto.setSc(c);
    eto.setSu(getSU(getSD(quantity),getDof(quantity)));
    eto.setValue(avg);
    eto.setTrueName(trueName);
    vt.add(eto);
```

Appendix III

NPLUnc MCM Calculation Code

Function implementing a Monte Carlo method

The inputs to the function are a specification **model** of the measurement model, a specification **pdfin** of the probability distributions assigned to the input quantities in the measurement model, a coverage probability **p**, the **type** of coverage interval, and controls **controls** for running the Monte Carlo calculation. The function returns an estimate **yMCM** of the output quantity, the standard uncertainty **uyMCM** associated with the estimate, the endpoints **IyMCM** of the coverage interval for the output quantity of the given type corresponding to coverage probability **p**, an approximation **pdfMCM** to the probability density function for the output quantity, and an indication **conv** of whether the Monte Carlo calculation has stabilized within the number of trials undertaken.

model is a string containing the name of a function for evaluating the measurement model for values of the input quantities. For an example of such a function, see model_additive.

pdfin is a matrix of **N** rows and **4** columns, the ith row of which defines the probability distribution for the ith input quantity. For a measurement model with a single input quantity, **pdfin** can take the following forms:

pdfin = [1, mu, sigma, inf] defines a Gaussian distribution with expectation **mu** and standard deviation **sigma** (the fourth element of **pdfin** is not used);

pdfin = [2, mu, sigma, nu] defines a t-distribution with shift parameter **mu**, scale parameter **sigma** and degrees of freedom **nu**;

pdfin = [3, a, b, inf] defines a rectangular distribution with lower limit **a** and upper limit **b** (the fourth element of **pdfin** is not used);

pdfin = [4, a, b, r] defines a curvilinear trapezoidal distribution with lower limit **a**, upper limit **b** and (fractional) reliability **r** for the semi-width;

pdfin = [5, a, b, inf] defines a U-shaped distribution with lower limit **a** and upper limit **b** (the fourth element of **pdfin** is not used).

p is the coverage probability, which can be 0.90, 0.95 or 0.99, and **type** defines the type of coverage interval, which can be 'Shortest' or 'Symmetric'.

controls is a row vector of five elements and contains controls for the Monte Carlo calculation as follows:

controls(1) determines whether the calculation is adaptive (1) or not (0);

controls(2) is the maximum number of Monte Carlo trials to be undertaken (as a multiple of 10 000);

controls(3) is the number of histogram bins for defining an approximation to the probability density function for the output quantity;

controls(4) is the initial state of the random number generator used in generating random draws from the probability distributions for the input quantities;

controls(5) is the number of significant decimal digits regarded as meaningful in the value of the standard uncertainty when testing for stabilization of the results.

```
function [yMCM, uyMCM, IyMCM, pdfMCM, conv] = ...
    MCM_calculation(model, pdfin, p, type, controls)

Input

model      string  Function defining the measurement model
pdfin      N x 4   Probability density functions for the input quantities
                   in the measurement model
p                  Coverage probability (0.90, 0.95 or 0.99)
type       string  Coverage type ('Shortest' or 'Symmetric')
controls   1 x 5   Controls for running a Monte Carlo calculation:
                     adap    the Monte Carlo calculation is adaptive
                             (1) or not (0)
                     M       the maximum number of Monte Carlo trials
                             (a multiple of 10 000)
                     Nb      the number of histogram bins for defining an
                             approximation to the probability density
                             function for the output quantity
                     state   the initial state for the random number
                             generator
                     ndig    the number of significant decimal digits
                             for testing for stabilization

Output

yMCM               Estimate of the output quantity
uyMCM              Standard uncertainty associated with the estimate of
                   the output quantity
IyMCM      1 x 2   Lower and upper endpoints of a coverage interval for
                   the output quantity
pdfMCM     2 x Nb  Approximation to the probability density function for
                   the output quantity
conv               Indicates whether the results of the Monte Carlo
                   calculation have stabilized (1) or not (0)
```

Step 1: Set controls for the Monte Carlo calculation

```
    adap   = controls(1);
    M      = controls(2);
    Nb     = controls(3);
    state  = controls(4);
    ndig   = controls(5);
```

Step 2: Set the state for the random number generator

```
rand('twister', state)
```

Step 3: Initialize the Monte Carlo calculation

```
yk     = zeros(M,1);
uyk    = zeros(M,1);
Iyk    = zeros(M,2);
m      = 10000;
conv = 0;
k      = 1;
```

Step 4.0: Start of loop

```
while ~conv && k < M+1
```

Step 4.1: Form values of the output quantity

```
xr = sampledata(pdfin, m);
yr = eval([model, '(xr)']);
```

Step 4.2: Form results for the complete set of values for the output quantity

```
if k == 1
    yMCM       = mean(yr);
    uyMCM      = std(yr);
    ymin       = min(yr);
    ymax       = max(yr);
    yleft      = [];
    yrght      = [];
    [fy, fx] = hist(yr, Nb);
else
    [yMCM, uyMCM, yleft, yrght, fx, fy] = ...
        MCM_update_results(ymin, ymax, yr, (k-1)*m, ...
                           yMCM, uyMCM, yleft, yrght, fx, fy);
end
```

Step 4.3: Form results for a subset of values for the output quantity

```
[yk(k), uyk(k), Iyk(k,:)] = MCM_results(yr, p, type);
```

Step 4.4: Test for stabilization of the results

```
        if k > 1
            if uyMCM == 0
                tol = 0;
            else
                r    = -(floor(log10(uyMCM)) - (ndig - 1));
                tol = 0.5*10^(-r);
            end
            stab = 2*std([yk(1:k) uyk(1:k) Iyk(1:k,1) Iyk(1:k,2)])/sqrt(k);
            if adap
                if max(stab) <= tol; conv = 1; end
            else
                if k == M
                    if max(stab) <= tol; conv = 1; end
                end
            end
        end
```

Step 4.5: End of loop

```
        k = k + 1;
    end
```

Step 5: Form approximations to the probability distribution for the output quantity

```
    if uyMCM == 0
        Gx = [yMCM, yMCM];
        Gy = [    0,       1];
        gx = yMCM;
        gy = 1;
    else
        [Gx, Gy, gx, gy] = MCM_distributions(yleft, yrght, fx, fy);
    end
    pdfMCM = [gx; gy];
```

Step 6: Evaluate the coverage interval for the output quantity

```
    IyMCM = coverage_interval(Gx, Gy, p, type);
```

Function for making random draws from various probability distributions

function xr = sampledata(pdf, M)

Input

pdf	N x 4	Probability density functions for the input quantities in the measurement model
M		Number of Monte Carlo trials

Output

xr	N x M	Random draws made from the probability distributions for the input quantities

For each input quantity, the calculation depends on the nature of the probability distribution assigned to the quantity.

```
N    = size(pdf,1);
xr = zeros(N,M);
for k = 1:N
    switch pdf(k,1)
```

Gaussian distribution.

```
        case {1}
            x    = pdf(k,2);
            ux = pdf(k,3);
            mn = 0;
            ndist = zeros(1,M);
            while mn < M
                v1 = rand;
                v2 = rand;
                if mn < M
                    mn = mn + 1;
                    ndist(mn) = sqrt(-2*log(v1))*cos(2*pi*v2);
                end
                if mn < M
                    mn = mn + 1;
                    ndist(mn) = sqrt(-2*log(v1))*sin(2*pi*v2);
                end
            end
            xr(k,:) = x + ux*ndist;
```

t-distribution.

```
        case {2}
            x    = pdf(k,2);
            ux = pdf(k,3);
            nu = pdf(k,4);
```

```
            mt = 0;
            tdist = zeros(1,M);
            while mt < M
                v1 = rand;
                v2 = rand;
                if v1 < 0.5
                    t = 1/(4*v1 - 1);
                    v = v2/t^2;
                else
                    t = 4*v1    - 3;
                    v = v2;
                end
                if v < 1 - abs(t)/2 || v < (1 + t^2/nu)^(-(nu+1)/2)
                    mt = mt + 1;
                    tdist(mt) = t;
                end
            end
            xr(k,:) = x + ux*tdist;
```

Rectangular distribution.

```
        case {3}
            a = pdf(k,2);
            b = pdf(k,3);
            xr(k,:) = a + (b - a)*rand(1,M);
```

Curvilinear trapezoidal distribution.

```
        case {4}
            a   = pdf(k,2);
            b   = pdf(k,3);
            r   = pdf(k,4);
            d   = r*(b - a)/2;
            am = a - d;
            ap = a + d;
            ak = am + (ap - am)*rand(1,M);
            bk = (a + b) - ak;
            xr(k,:) = ak + (bk - ak).*rand(1,M);
```

U-shaped distribution.

```
        case {5}
            a = pdf(k,2);
            b = pdf(k,3);
            xr(k,:) = (a + b)/2 + ((b - a)/2)*cos(pi*rand(1,M));
        end
    end
```

Function for updating the results of a Monte Carlo calculation based on a new set of values of the output quantity

function [yMCM, uyMCM, yleft, yrght, fx, fy] = ...
    MCM_update_results(ymin, ymax, yr, M0, yMCM, uyMCM, yleft, yrght, fx, fy)

Input

ymin		Left-hand end of the first bin
ymax		Right-hand end of the last bin
yr	1 x m	New values of the output quantity
M0		Number of Monte Carlo trials undertaken to obtain the current results (below)

Input/output

yMCM	Estimate of the output quantity
uyMCM	Standard uncertainty associated with the estimate of the output quantity
yleft	Values of the output quantity less than ymin
yright	Values of the output quantity greater than ymax
fx, fy	Frequency distribution for the output quantity between ymin and ymax

```
m      = length(yr);
d      = sum(yr - yMCM)/(M0 + m);
mu     = yMCM + d;
s2     = ((M0 - 1)*uyMCM^2 + M0*d^2 + sum((yr - mu).^2))/(M0 + m - 1);
yMCM   = mu;
uyMCM = sqrt(s2);
ileft = find(yr < ymin);
yleft = [yleft, yr(ileft)];
irght = find(yr > ymax);
yrght = [yrght, yr(irght)];
yr([ileft, irght]) = [];
fy = fy + hist(yr, fx);
```

Function for evaluating the results from a Monte Carlo calculation based on a set of values of the output quantity

function [yMCM, uyMCM, IyMCM] = MCM_results(yr, p, type)

Input

yr	1 x M	Values of the output quantity
p		Coverage probability (0.90, 0.95 or 0.99)
type	string	Coverage type ('Shortest' or 'Symmetric')

Output

yMCM Estimate of the output quantity
uyMCM Standard uncertainty associated with the estimate of the output quantity
IyMCM 1 x 2 Lower and upper endpoints of a coverage interval for the output quantity

```
yMCM   = mean(yr);
uyMCM = std(yr);
M   = length(yr);
Gx = sort(yr);
Gy = ((1:M) - 1/2)/M;
IyMCM = coverage_interval(Gx, Gy, p, type);
```

Function for evaluating approximations to the probability distribution for the output quantity

function [Gx, Gy, gx, gy] = MCM_distributions(yleft, yrght, fx, fy)

Input

yleft Values of the output quantity less than ymin
yright Values of the output quantity greater than ymax
fx, fy Frequency distribution for the output quantity between ymin and ymax

Output

Gx, Gy Approximation to the distribution function for the output quantity
gx, gy Approximation to the probability density function for the output quantity

```
ml = length(yleft);
mr = length(yrght);
Nb = length(fx);
M   = ml + sum(fy) + mr;
bw = fx(2) - fx(1);
Gx(1) = fx(1) - bw/2;
Gy(1) = ml/M;
for k = 1:Nb
    Gx(k+1) = Gx(k) + bw;
    Gy(k+1) = Gy(k) + fy(k)/M;
end
```

Otherwise, make allowance for any values in **yleft** and **yrght**.

```
if ml > 0
    Gx = [sort(yleft), Gx];
    Gy = [((1:ml) - 1/2)/M, Gy];
end
if mr > 0
```

```
        Gx = [Gx, sort(yrght)];
        Gy = [Gy, ((M-mr+1:M) - 1/2)/M];
    end
    iz = find(diff(Gy) == 0);
    if ~isempty(iz)
        for k = 1:length(iz)
            Gy(iz+1) = Gy(iz) + 10*eps;
        end
    end
    if ml > 0
        Nbl = fix((fx(1) - bw/2 - min(yleft))/bw) + 1;
        fxl = linspace(fx(1)-Nbl*bw, fx(1)-bw, Nbl);
        if Nbl == 1
            fyl = ml;
        else
            fyl = hist(yleft, fxl);
        end
        fx   = [fxl, fx];
        fy   = [fyl, fy];
    end
    if mr > 0
        Nbr = fix((max(yrght) - fx(end) - bw/2)/bw) + 1;
        fxr = linspace(fx(end)+bw, fx(end)+Nbr*bw, Nbr);
        if Nbr == 1
            fyr = mr;
        else
            fyr = hist(yrght,fxr);
        end
        fx   = [fx, fxr];
        fy   = [fy, fyr];
    end
    gx = fx;
    gy = fy/(M*bw);
```

Function for evaluating a coverage interval for the output quantity

function IyMCM = coverage_interval(Gx, Gy, p, type)

Input

Gx, Gy		Approximation to the distribution function for the output quantity
p		Coverage probability (0.90, 0.95 or 0.99)
type	string	Coverage type ('Shortest' or 'Symmetric')

Output

IyMCM	1 x 2	Lower and upper endpoints of a coverage interval for the output quantity

```
    if strcmp(type, 'Shortest')
```

```
        M       = length(Gy);
        pcov = linspace(Gy(1), Gy(M)-p, 101)';
        ylow = interp1(Gy, Gx, pcov);
        yhgh = interp1(Gy, Gx, pcov+p);
        lcov = yhgh - ylow;
        imin    = find(lcov == min(lcov), 1, 'first');
        IyMCM = [ylow(imin), yhgh(imin)];

  end
    if strcmp(type, 'Symmetric')
        ylow    = interp1(Gy, Gx, (1-p)/2);
        yhgh    = interp1(Gy, Gx, (1+p)/2);
        IyMCM = [ylow, yhgh];

    end
```