Internet-Enabled Calibration: An Analysis of Different Topologies and a Comparison of Two Different Approaches

Åsmund Sand, Michael Stevens, and Graeme Parkin

***Abstract*—This paper discusses different network topologies used in Internet-enabled metrology and calibration and explores and compares two different remote calibration systems used by the National Metrology Institutes in England and Norway: the National Physical Laboratory (NPL) and the Justervesenet (JV). The two systems are iGen (NPL) and iMet (JV). The systems both deal with remote calibration of electrical equipment but have substantial architecture differences. In iGen, calibration procedures are downloaded from a server and then locally run at the instrument client, where the operator sits. The client is generic in such a way that it is not dependent on the structure of the measurement procedures. In iMet, two clients can communicate via a public server, and the calibration process may be remotely controlled and monitored. That is, the instruments and the operator may be separated by the Internet.**

***Index Terms*—Calibration, distributed measuring systems, Internet, Internetworking, measurement.**

I. Introduction

THE introduction of the Internet in the 1990s enabled people to digitally communicate over long physical distances. This communication can be done in a fast and secure way, and today, a lot of services are available to the public. These include text chat, video conferencing, IP telephony, document collaboration, and online gaming.

In recent years, the international metrology community has started to explore the possibilities of remotely monitoring and controlling measurements and calibrations via the Internet [1]–[4]. When performing remote calibrations, a calibrated transfer standard is first transported to the device-under-test (DUT). Both the standard and the DUT are then connected to a PC, which again connects to the Internet. The calibration process may be controlled or monitored in many ways. The operator may either be colocalized with the instruments or may be separated from them by a network. The many advantages of running calibrations in such a way are as follows.

1) The DUT is calibrated in its working environment, thus lowering the total calibration uncertainty.
2) The DUT is out of operation for a much shorter period than for the traditional calibrations, which is often critical for the owner of the DUT.
3) The effects of transporting the transfer standard are often much better understood than the effects of transporting the DUT.
4) If a suitable transfer standard is found, traceability is obtained, and at the same time, the calibration chain is cut down to only one link.
5) If a suitable transfer standard is found, a much better cost-to-accuracy ratio is obtained.
6) The DUT owners usually get more involved in the calibration process, which may be a benefit for their normal laboratory work.

II. System Architectures

There are normally three different system architectures that can be used for instrument operation and remote calibration.

1) The operator and the instruments are located at the same local-area-network (LAN) computer connected to the Internet. Measurements are downloaded in the form of software routines from a public server.
2) The operator uses a LAN computer, while the instruments are connected to a public web server or a computer with easy access to a public web server. This architecture is quite useful when setting up online laboratories.
3) The operator and the instruments are located at different LAN computers separated by the Internet. Instrument control commands and measurements may be sent from the operator to the instruments via a public relay server.

The architectures mentioned have different advantages and disadvantages.

The type 1) architecture is quite robust to unstable network connections since the instrument communication is locally done. The instrument communication is also very fast. However, this architecture needs a stable connection when performing a multistep calibration. If the entire calibration routine could be downloaded all at once, the connection to the server would need to be open only while downloading the routine (and while uploading the calibration results). There is also the need for the operator to travel to the instruments to control them.

The type 2) architecture promotes the use of very thin clients (often, a regular web browser is enough on the client side); thus, little needs to be installed on the client computer before using the system. Since the instruments need to be connected to a public web server, this architecture is not suitable for general instrument operation (most instruments are connected

Manuscript received March 10, 2005; revised January 15, 2007.

Å. Sand is with the Justervesenet, the Norwegian Metrology Service, 2007 Kjeller, Norway (e-mail: aasmund.sand@justervesenet.no).

M. Stevens and G. Parkin are with the National Physical Laboratory (NPL), TW11 0LW Teddington, U.K. (e-mail: Mike.Stevens@npl.co.uk; Graeme.Parkin@npl.co.uk).

Digital Object Identifier 10.1109/TIM.2007.903628

to the regular LAN computers). For direct instrument control, the communication is dependent on the available bandwidth.

The type 3) architecture enables the users to remotely operate the instruments anywhere, as long as the instruments are connected to a computer connected to the Internet. In addition, there is no need for the operator to travel to the instruments to control them; thus, third-party experts could easily take part in the control process. However, this architecture can be sensitive to unstable network traffic and network congestions, since persistent connections are often needed when doing direct instrument control. If a calibration process can be run all at once, the sensitivity is the same as for type 1). Direct instrument control is dependent on the available bandwidth. The occurrence of long delay times may distract the operator.

Of the three architecture types presented, only types 1) and 3) seem suitable for Internet-enabled calibration, since type 2) requires the setting up and configuration of a public server for each instrument to be calibrated.

III. Preparations

Before developing systems used for the Internet-enabled calibration, several issues need to be addressed by the National Metrology Institute (NMI) or calibration laboratory.

A. Operator

A decision needs to be made about who should be the operator: in-house experts, third-party experts, or maybe the customer.

B. Operator Localization

This means choosing between one of the three architecture types mentioned. For the Internet-enabled calibration, this usually means type 1) or 3).

C. Degree of System Autonomy

Should the operator always have total control of the process, or should the system sometimes decide which actions to perform? Maybe the system could be completely automatic?

D. Data Handling

Should data produced in the measurements be analyzed right away, or should it just be stored for future analysis? Should the analysis be done manually or automatically?

E. Security

The system owner must have a clear understanding of who should be authorized to use the system. If the data generated by the system are sensitive, they should be encrypted. To obtain data integrity, digital signatures should be used to sign measurement procedures and data.

F. Availability

Should the system always be up and running, or is it alright to have downtimes? The system functionality often depends on the stability of the network conditions. Does the system work on all LANs, or is interference from network administrators necessary? On which type of software platform should the system be able to run (e.g., Windows, Linux or Mac)?

G. Scalability

Is it desirable to add new calibration routines while the system is running, or will the system be shut down? A decision needs to be made about which hardware interfaces should be supported and what equipment will be connected. Will it be necessary to add new hardware interfaces or equipment?

H. Development

Will it be easy to develop and test new calibration routines, or are software experts needed? Should the system be static, or will it be necessary to add extra functionality to the system?

I. Configuration

Often, addresses on the Internet change, and sometimes, new communication protocols are needed. Will the system be developed with some kind of configuration possibilities? Can the system be configured using text files, or will the configuration happen in assembly (need to recompile for every change made)?

J. Usability

Should anyone be able to use the system, or will user training be needed? Should it be a system for experts?

IV. Choice of Architecture

In this paper, we look at the National Physical Laboratory's (NPL) iGen system, based on iOTDR [5], which has been used for optical fiber measurements with optical time domain reflectometers (OTDRs), and Justervesenet's (JV) iMet system [6], which will be used in calibrating electrical instruments via the Internet.

NPL has done much work in the Internet-enabled calibration area, and over the past few years, several systems, like iPIMMS, iOTDR, iVR, and iCOLOUR [7], have been developed. These systems all belong to the type 1) architecture. The iGen system was built on the iOTDR system and, thus, inherited its type 1) architecture.

JV had done little work in the area before 2002 and had to choose one of the three architectures. The type 3) architecture seemed the better choice, since it was decided that the calibration process should be controlled by personnel at JV without the need to travel to the DUT. The present iMet system is, therefore, of type 3).

V. Systems Overview

The two systems are described in detail as follows. Both the composition and the functionality of the systems are discussed.

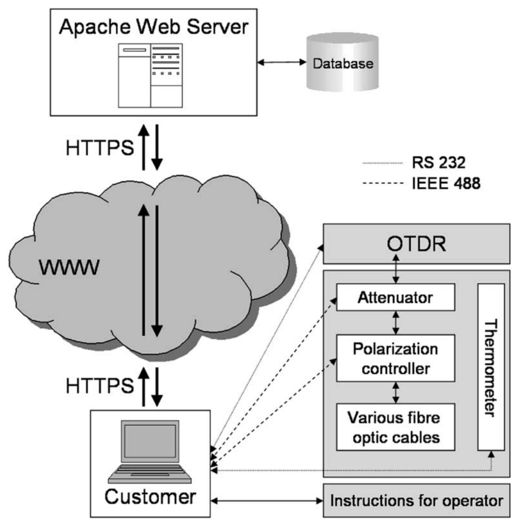

Fig. 1. iGen overview. The customer sets up a secure connection to the server, downloads calibration scripts from the server's database, and locally runs the calibration. When finished, the results are sent back to the server for further processing. The figure shows the instrument setup used in the calibration of linearity of an OTDR. This part of the calibration of an OTDR requires the use of two RS232 and two IEEE488 devices.

A. iGen

1) System Structure: The system architecture is shown in Fig. 1. The software is composed of a client, written in Microsoft Visual Basic [8], which runs on a computer located at the customer's site, and a server program, written in PHP [9], which runs on a server computer at the calibration laboratory.

The iGen server uses a MySQL database [10] for storage, which contains instrument information, calibration history data, and measurement procedures.

2) System Functionality: In the client, a Visual Basic control called INET [12] is used in handling communications with the server. The control calls a function "execute" to send messages to the server. This function takes the address of the server and the message to be sent, which is in Extensible Markup Language (XML) format, as parameters. The protocol in use, which is XMP-RPC [11], works over Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS) and is thus quite firewall friendly. The XML-RPC protocol enables structurally complex and binary data to be exchanged.

When performing a calibration, the server controls the calibration process, while the client, which is using instructions received from the server (in the form of VB Script [13]), controls the equipment and interacts with the operator. Data collected from the instruments are transmitted back to the server for processing, and the results are stored in a database.

Although the server software dictates how the calibration is conducted, the client software is in control throughout the process and initiates all communication. The client sends a message to the server and waits for a reply. Upon receipt of each message, the server acts and then issues a reply.

A calibration is performed by executing a sequence of procedures on the server. Effectively, the server is state driven. Each message sent from the client includes the name of the function that the server is to execute; this name has been sent from the server to the client in the previous message. The first message from the client names the "loginpage" function that the server uses to log the user on to the system. The server returns an HTML page to the client, enabling the user to enter the login name and the password. It also returns the name of the function that must be called next to start the calibration sequence. The client responds with a new communication that reactivates the server. The message includes the login name and the password, and the name of the function to be executed.

The client software has the capability of talking up to four RS232 ports and as many IEEE488 [14] addresses as the National Instruments card can handle. A Visual Basic control is included in running VBScript. The VBScript is a subset of Visual Basic, and it can be sent down from the server as an ASCII string to be run on the client. The VBScript can include calls to the Visual Basic functions on the client that read data from or write instructions to the hardware. These function calls can be embedded in loops.

3) Demonstration: The complete calibration of an OTDR consists of four different calibrations: the wavelength, the distance scale, linearity, and loss scale factor. The iGen system has been used in performing two of these calibrations over the Internet, namely, the distance scale and the linearity. The customer is instructed through each of these calibrations over the Internet, including the setup of the instruments. The reference standard used is a calibrated fiber. Fig. 1 shows the instrument setup for the calibration of linearity of the OTDR. Once the instruments are correctly set up, the calibration is completely and automatically done (using the VBScript), which includes the following: checks to see whether the instruments are correctly set up, automatic adjustment of the instruments, using loops to wait for instruments to complete their measurements, and monitoring key environmental factors like temperature.

B. iMet

1) System Structure: The system architecture is shown in Fig. 2. The system consists of a LAN computer with some connected instruments, an operator's LAN computer, and a public Microsoft IIS Web Server. The operator could be colocated with the instruments, and then, only one LAN computer is needed. The instrument computer runs a specific .NET Remoting [15] client application, which handles communication with the server and the locally connected instruments. The operator's computer also runs a .NET Remoting client application, which handles communication with the server. The last application is built to control the former, so that the operator may access the remote instruments as if they were locally connected. The client applications communicate via a software object hosted on the server. Both the clients and the object are written in Microsoft C# [16].

The server utilizes the data stored in an SQL Server database [17], which contains instrument information, calibration history data, and measurement procedures.

2) System Functionality: Both clients set up a full-duplex HTTPS channel to the server before logging on. This enables the data to be sent back and forth without inefficient client polling. The solution is based on the code provided by

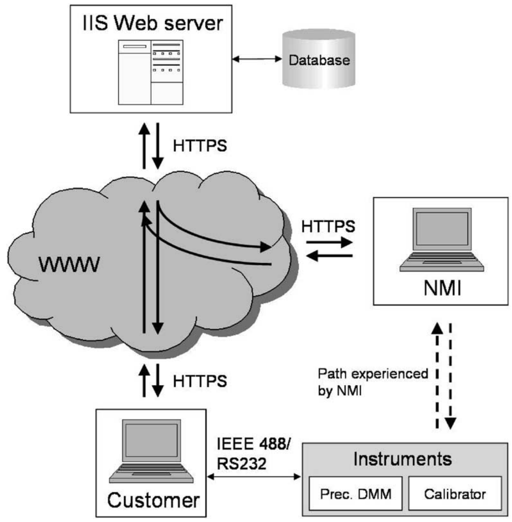

Fig. 2. iMet overview. The controlling NMI authority operates the customer's instruments via secure connections to the web server. To the authority, the instruments seem to be locally connected. The calibration procedures are stored as text in the server's database. When needed, the procedures can be sent to the customer and locally compiled and run. The figure shows the instrument setup used in the calibration of an electrical calibrator using a precision DMM as a reference. There is no limitation as to which instruments may be connected.

GenuineChannels [18]. The channel basically consists of two regular HTTPS connections, where one is in a pending HTTP request mode, enabling the server to instantly contact a client.

After logging in, the clients may communicate seamlessly via the server without configuring local firewalls, proxy servers, or Network Address Translators (NATs). This is due to the firewall-friendly nature of the HTTPS. To the clients, the server is transparent, and it seems to them that the interclient communication directly happens.

A calibration may be done in two ways. The operator can tell the server to send the calibration routine to the instrument computer to be locally run. This is useful when the routine contains frequent instrument calls. Alternatively, the calibration routine may be downloaded from the server and directly run on the operator's computer. Then, only the individual instrument calls are sent via the Internet to the instrument computer. This way, the operator has more insight into the calibration process, but the process is sensitive to the changing network conditions.

There is no limitation as to where the operator may sit, as long as there is an Internet connection available.

The system may also be used for remote instrument control, where an operator sends individual instrument commands to the remote instruments. To the operator, the remote instruments seem locally connected, except for possible network delays. Tests have shown typical delay times of up to 2 s. It is hard to say if the delay times of this order are acceptable or not, but there is obviously a tradeoff between the functionality and the delay times. In order to allow the third-party experts to remotely operate an instrument (e.g., to try to find the source of odd behavior), one has to accept delay times some degree.

The .NET application at the instrument computer communicates with the connected instruments via the Virtual Instrumentation Software Architecture (VISA) interface [19]. The VISA interface makes it possible to communicate over several hardware interfaces, e.g., RS232 or IEEE488, without the need to change the control application.

3) Demonstration: JV tested the system in a realistic setup with two connected client computers (an authority at JV and a customer), with firewalls at both locations, and a public web server. The instruments involved were two electrical precision digital multimeters (DMMs), used as reference, and an electrical calibrator, which is the DUT. Two multimeters were used to investigate the effects of transport. The customer connected the multimeters and the calibrator to his computer, logged in to the public server, and registered the two instruments. The authority also connected to the server and obtained a reference to the customer and to the two connected instruments. The correct calibration procedure was then picked from a list. The list was automatically generated on the server, on the basis of which instruments were connected. The procedure source code was then sent to the client and compiled and run. Three different calibrations were performed: ac/dc current, ac/dc voltage, and dc resistance. All results were returned to the authority, displayed in a graph, and then stored for further processing. A later analysis showed that the two multimeters had slightly changed after transportation, and more work is therefore needed to find suitable transport standards.

VI. Comparison

In this section, the two systems will be compared against the checklist under Section III (see Tables I–X).

TABLE I

The operator	
iGen	The operator is meant to be the customer
iMet	The operator is meant to be an NMI expert
Conclusion	If operated by experts, the error rate would be minimized. This difference may affect how easy it would be to get the service accredited. The customer could be trained to become an expert user.

TABLE II

Operator localization	
iGen	The operator needs to be with the instruments
iMet	The operator may sit anywhere as long as an Internet-connection is available
Conclusion	iMet has more options as to who may operate the system, without the need to travel to the instruments. This is directly correlated with the operator role of the systems.

TABLE III

Degree of system autonomy	
iGen	The operator initiates each step of the calibration process.
iMet	Same as for iGen.
Conclusion	Both systems are somewhat non-autonomous. If making the system more autonomous, the error rate could potentially be lowered.

TABLE IV

Data handling	
iGen	The data is shown to the operator using a built-in web browser. It is then uploaded to the database for further analysis.
iMet	The data is shown to the operator using a custom graph area. It is then uploaded to the database for further analysis.
Conclusion	iGen has a greater potential for displaying the data to the operator in a dynamic way, due to its HTML-capabilities.

TABLE V

Security	
iGen	All data traffic between the client and the server is encrypted according to the HTTPS protocol. The system uses an X.509 server certificate. No data is signed. All users sign in with a username and password.
iMet	Same as for iGen. In addition, each client also uses a server-signed X.509 client certificate in the authentication process.
Conclusion	Both systems support data encryption. To reduce the risk of server attacks, the iGen system should add support for X.509 client certificates in the authentication of connecting clients. They should both add support for signing the calibration routines (by the NMI) and the calibration data (by the instrument computer), for additional integrity.

TABLE VI

Availability	
iGen	The system could be set to run always. The system is quite robust with regard to changing network traffic, since the instrument communication happens locally. If behind a non-transparent proxy server, the client application at the customer cannot access the server. The system is dependent on the Windows operating system.
iMet	The system could be set to run always. The system is sensitive to changing network conditions when operating instruments directly from remote, since this communication requires keep-alive connections between the client computer and the server. The system also works from behind non-transparent proxy servers. The system is dependent on a Windows operating system. There exist potential solutions **Error! Reference source not found.** to run the client software on Linux, though these have not been tested.
Conclusion	Both systems are quite available, even though iMet is somewhat more sensitive to changing network conditions when operating instruments from remote. The iGen system should add support for non-transparent proxy servers, so that more customers may use the system. Open source solutions should be sought to replace the present Windows-dependent systems.

TABLE VII

Scalability	
iGen	New calibration routines may be added as source code to a database, while the system is running. The system currently supports IEEE 488 and RS232, and potentially all equipment communicating via these hardware interfaces.
iMet	Same as iGen. In addition, the system works with all hardware interfaces supported by the VISA standard. In previous versions of the system, new calibration routines would have to be incorporated into the client application before shipping to the customer.
Conclusion	Both systems are quite scalable, and it is easy to add new calibration routines and equipment (if these communicate via GPIB or RS232). It is easier for the iMet system to add support for new hardware interfaces, if these are supported by the VISA standard.

TABLE VIII

Development	
iGen	The system is developed using Visual Basic, VB Script, PHP and SQL, but lacks a good development platform. This, with the heterogeneity of the software, makes debugging difficult, and error messages (if they appear) are often quite general.
iMet	The system is developed using C# and SQL, and supports a good development platform (MS Visual Studio). Due to C#'s object-oriented nature and its exception handling capabilities, it is easier to develop more complex systems.
Conclusion	The iMet system is easier to develop, because of the chosen programming language and its good development platform.

TABLE IX

Configuration	
iGen	The system uses XML-based configuration files, where the client can change different settings, e.g. the application server address, the port number, and the protocol in use (HTTP or HTTPS).
iMet	Same as for iGen.
Conclusion	Both systems are quite easy to configure, which make the clients very flexible without the need for recompilation.

TABLE X

Usability	
iGen	The operator gets guidance from the server-generated HTML pages. Most often user interaction consists of picking values from dropdown lists and pushing buttons.
iMet	The guidance of the operator is not as dynamic as for the iGen system, and more training is needed. Most often user interaction consists of picking values from dropdown lists and pushing buttons.
Conclusion	The iGen system is somewhat easier to use for novice users, due to the flexibility of HTML. This is positive since the customer is supposed to be the operator for this system.

VII. CONCLUSION

An analysis of different network topologies in Internet-enabled calibration has been performed, and two different approaches to Internet-enabled calibration have been analyzed and compared in this context, which are the NPL's iGen system and the JV's iMet system. There are pros and cons associated with each of them.

The most important difference is the system structure. For the iGen, the customer is supposed to be the operator, and the system is constructed so that the operator and the equipment are placed together. For the iMet, an NMI person is supposed to be the operator, the system is constructed so that the operator and the equipment may be separated by the Internet, and the operator may remotely control the calibration process without needing to go to the equipment under test.

The difference in operator role also reflects the usability of the system. The iGen system is believed to be easier to use for new personnel, because detailed instructions are given to the operator by server-generated web pages.

The development environment for iGen is much more difficult than the .NET facilities available to iMet.

REFERENCES

[1] F. Pianegiani, D. Macii, and P. Carbone, "An open distributed measurement system based on an abstract client–server architecture," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 3, pp. 686–692, Jun. 2003.
[2] M. Bertocco, F. Ferraris, C. Offelli, and M. Parvis, "A client–server architecture for distributed measurement systems," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 5, pp. 1143–1148, Oct. 1998.
[3] W. Winiecki and M. Karkowski, "A new Java-based software environment for distributed measuring systems design," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 6, pp. 1340–1346, Dec. 2002.
[4] M. Bertocco and M. Parvis, "Platform independent architecture for distributed measurement systems," in *Proc. IMTC*, May 2000, vol. 2, pp. 648–651.
[5] *Measuring Optical Time-Domain Reflectometers Over the Internet*. [Online]. Available: http://www.npl.co.uk/scientific_software/case_studies/software_dev/iotdr/
[6] Å. Sand and H. Slinde, "IMET—A secure and flexible approach to Internet-enabled calibration at Justervesenet," in *Advanced Mathematical and Computational Tools in Metrology*, vol. 72. Singapore: World Scientific, 2006, pp. 229–236.
[7] *Survey of International Activities in Internet-Enabled Metrology*. [Online]. Available: http://www.npl.co.uk/ssfm/download/documents/cmsc21_03.pdf
[8] *Microsoft Visual Basic Developer Center*. [Online]. Available: http://msdn.microsoft.com/vbasic/
[9] *PHP: Hypertext Preprocessor*. [Online]. Available: http://www.php.net
[10] *MySQL*. [Online]. Available: http://www.mysql.com/
[11] *XML-RPC*. [Online]. Available: http://www.xmlrpc.com/
[12] *Internet Transfer Control*. [Online]. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon98/html/vbconusinginternettransfercontrol.asp
[13] *VB Script*. [Online]. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/0a8270d7-7d8f-4368-b2a7-065acb52fc54.asp
[14] *IEEE 488*. [Online]. Available: http://www.ni.com/gpib/
[15] *Microsoft.NET Remoting*. [Online]. Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/hawkremoting.asp
[16] *Microsoft C#*. [Online]. Available: http://msdn.microsoft.com/vcsharp/
[17] *Microsoft SQL Server*. [Online]. Available: http://www.microsoft.com/sql/default.mspx
[18] *GenuineChannels*. [Online]. Available: http://www.genuinechannels.com
[19] *Virtual Instrumentation Software Architecture*. [Online]. Available: http://www.ni.com/visa/
[20] *The Mono Project*. [Online]. Available: http://www.go-mono.com

Åsmund Sand received the M.Sc. degree in physics from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2000. He received the Ph.D. degree in Internet-enabled metrology from NTNU's University Graduate Center (UniK), Kjeller, Norway, in 2006.

He is currently working with issues concerning the application of the Internet to electrical metrology and calibration with the Justervesenet, the Norwegian Metrology Service, Kjeller.

Michael Stevens received the B.Sc. degree in computer science from Kingston Polytechnic (now Kingston University) in 1974.

He has been with the National Physical Laboratory, Teddington, U.K., since 1974. He started working on an information retrieval system and writing Braille translation software. He moved on to handwritten character recognition analysis and developing electronic paper software. He then went on to develop benchmarks for parallel processors, particularly transputers. More recently, he has been developing a generic Internet calibration service.

Graeme Parkin received the M.Sc. degree in mathematics in 1975.

He has been with the National Physical Laboratory, Teddington, U.K., since 1978. He started working in the area of data security, where he developed a software protection device using cryptographic techniques. From there, he became involved in software engineering. He was involved in the application of formal methods in specifying various security standards and prototyping these in various languages. He was also involved in standardizing VDM-SL. He then went on to apply a model checking tool, which is known as Prover, in the railway and nuclear industries. Recently, he has been looking at ways to ensure that the software is fit for the purpose and testing of compilers.