Unisys ClearPath Forward® Systems
The Integrated Stack

By Peter Bye

White Paper
IT systems can be procured and delivered in a variety of ways; we can think of a spectrum of options. At one extreme, the various components making up the system – hardware, firmware and software – can be sourced from a variety of suppliers and integrated by the purchasing organisation. At the other extreme, the majority or all of the components can be obtained from a single source. We call these extremes the Multi-Source model and the Integrated Stack model.

This paper examines the characteristics of the two models, as well as two intermediate options, and argues that the Integrated Stack model, as it used for ClearPath Forward® systems, has many benefits in reliability, efficiency, security and support.

The paper was first published in 2014 with the title ‘Unisys ClearPath Systems: Integrated hardware/software stacks’. This revision reflects product name changes and the fact that all ClearPath Forward systems now use Intel hardware, with firmware providing the system architecture rather than hardware. ClearPath Forward Libra (ClearPath® MCP-based) and Dorado (ClearPath® OS 2200-based) systems use Unisys supplied Intel-based hardware. The ClearPath Software Series may run in hardware of the client’s choice.
# Table of Contents

Summary 4  
IT System Components 5  
The Multi-Source and Integrated Stack Models 6  
  The Multi-Source Model 6  
  The Integrated Stack Model 7  
ClearPath Forward Deployment Options 9  
Discussion 10  
Conclusions 13  
About the Author 14  
More Information 16
Summary

Obtaining efficient and cost/effective IT systems is a central concern for the CIO and other senior management. Two of the most common approaches, which represent opposite ends of a spectrum of possibilities, are:

1. The Multi-Source model, where the software and the platform(s) in which it runs are obtained from a number of suppliers, each of which is regarded as the best or most cost/effective option; the term ‘best of breed’ is often used when selecting products.

2. The Integrated Stack model, where most or all of the components are designed, developed, integrated and tested before delivery by a single organisation.

A note on terminology. All ClearPath Forward systems run on Intel hardware, with firmware (infrastructure software) providing the system architecture – instruction set, I/O connectivity and so on. For ClearPath Forward Libra (ClearPath MCP-based) and ClearPath Forward Dorado (ClearPath OS 2200-based), Unisys supplies the Intel-based hardware and the firmware. For the ClearPath Software Series, the client chooses the hardware; Unisys supplies the firmware and provides guidance on the required hardware configuration. This paper uses the word ‘machine’ to cover both of these options for ClearPath Forward systems. More generally, machine is used to mean either hardware that provides the system architecture, or a combination of hardware and firmware.

The Integrated Stack model has deep roots, typified originally by the way the mainframe was delivered. The Multi Source model is more recent, growing up during the late 1980s and 1990s. It dominated thinking for a long time because of the assumed benefits of the approach. In particular, a choice of software components such as database managers and application servers led to the view that costs would be lower without any significant sacrifice in performance.

That view has changed as the complications in deployment, integration and efficiency have become apparent. An illustration of a change of view in the industry during the past few years was Oracle’s adoption of the Integrated Stack model. This approach is of course exactly what Unisys has been following for decades. So has IBM, although with a higher proportion of third party products than Unisys.

This paper argues the case for the Integrated Stacks model in general and its significance for ClearPath Forward systems in particular. As will become clear, the Integrated Stack model offers considerable benefits:

- The machine and the software providing the operating environment are designed, developed, integrated and tested together before shipment, substantially reducing or even eliminating the client’s integration and test effort.
- The approach optimises performance, availability, reliability and security.
- The vendor can make changes to optimise the complete system. It allows the machine to evolve as capabilities may be located in different ways to take advantage of technology advances. The machine/software stack brings benefits to existing code, and enables new feature content as new capabilities are introduced. Examples include moving functions between software components and implementing special functions in the machine.
- There is a single support contact.
- ClearPath Forward users do not have to change applications with new system releases, reducing costs and simplifying deployment. The integrated stack does not guarantee no changes but it is a necessary condition. Unisys provides the extra benefit of requiring no application changes.

These benefits complement the open attributes of ClearPath Forward systems. The result is powerful and secure transaction processing platforms, able to participate in distributed environments. The ability to incorporate emerging technologies into ClearPath Forward systems enables them to stay up-to-date.

The remainder of the paper is organised as follows. The components of a stack are briefly explained. The two models are then discussed in more detail, followed by an analysis of their relative merits. The major findings of the analysis are then summarised, and some pointers to more information provided.
IT System Components

A system comprises an expanding array of software elements and a machine in which they run. The operating system – ClearPath MCP or ClearPath OS 2200, z/OS, Linux, UNIX and Windows among others – provides the runtime environment in which the applications execute, as shown schematically in figure 1. In the figure, the software components have been grouped into a small number of categories, which represent functional areas within a system. All the components, apart from the applications, constitute the stack.

As noted earlier, the machine implements the system architecture – instruction set and so on. It may be implemented in silicon in the hardware or in firmware running in some other hardware.

The components in the categories at the centre of the figure (inside the dashed box) manage the applications and their interactions with data stores and external systems:

- **File and database management** includes relational, network, hierarchical and other database models as well as simpler file systems such as the Common Internet File System (CIFS).
- **Transport and communications** manages the flow of traffic from networks into and out of the system. It would typically include TCP/Sockets and lower layers.
- **Middleware** is concerned with communication with other systems in a distributed environment, and includes message queuing, Web services and a variety of options for integrating different standards. In general, it uses the services of the transport and communications components.
- **Application environments** provide the facilities to support transaction, batch and timesharing operation.
- **Compilers and utilities** provide the tools for programming language processing and the variety of deployment and housekeeping functions any system needs.
Two other categories of component are shown at the sides; they provide services to optimise operation under normal and abnormal conditions. Their importance cannot be overstressed for enterprise-class systems.

- **Systems management and recovery** is concerned with system operation, including automation, and the preparation for and recovery from any failure.
- **Security** is responsible for protecting the system and its databases from external and internal threats, malfunctions (for example a rogue program), and for detecting and reporting any attempted violations.

All the components run under the operating system – MCP or OS 2200 in the case of ClearPath Forward systems.

The software runs in a machine, which provides key architectural functions such as the instruction set and I/O connectivity. The machine architecture may be supplied by the hardware platform or implemented in firmware, which provides the machine architecture within some other hardware platform. Operating systems may also run under a hypervisor such as VMware. Systems may also contain partitions or appliances supporting different environments (not shown in figure 1), housed within the host system and under its control, security and management. They should be regarded as an integral part of that system and not seen as separate systems.

### The Multi-Source and Integrated Stack Models

This section fleshes out the characteristics of the two models.

#### The Multi-Source Model

In the most extreme form of the Multi-Source model, the great majority of the software components and the machine in which they run come from different sources. Until the mid to late 1990s, this was very much a popular option. While there were some companies, notably Microsoft, which aimed to supply the whole software stack, there were many niche providers of products aimed at specific technical areas.

A combination of vendors extending their product ranges and consolidation by acquisition within the industry has resulted in more products coming from single suppliers. Each aims to provide a comprehensive package of products to its customers. An example is BEA Systems and middleware. BEA started with Tuxedo, expanded into Java with WebLogic, adding other products such as the AquaLogic Service Bus and WebLogic Integration, before finally being acquired by Oracle. Some companies have focused on specific technical areas, such as security, performance or business process management, developing a comprehensive range of products.

Even allowing for the degree of consolidation, many installations follow a ‘mix and match’ approach by combining products from a number of sources. Figure 2 and the following paragraphs discuss a typical example.

The system shown in figure 2 contains a Java application, which is to run within a wider environment belonging to an enterprise called ‘X’. The Java application’s primary function is to provide services to a number of other applications running in other platforms within X. It has in addition a number of internal users who access it through web browsers. Figure 2 shows the major components used; the list is not exhaustive.

- The selected machine is hardware in the form of a Dell PowerEdge Blade Server, or rather servers, because the configuration is intended to span more than one physical tier – a typical tiered configuration.
- X allows the use of open source products in a number of areas, assuming they can be adequately supported. Red Hat Enterprise Linux is chosen as the operating system. The transport and network components are supplied with Red Hat Enterprise Linux as part of the package.
- The application environment is Java EE for which the Red Hat JBoss Enterprise Application Platform (EAP) and JVM (Java Virtual Machine) are approved products for use in X.
- X has a corporate database policy of using Oracle with this kind of system, with the open source Hibernate as the persistence manager for use with Java.
Communication with other internal systems is based on message queuing, using IBM MQ, another product approved by corporate policy. The Apache Web Server supports the browser access.

Systems management uses BMC ProactiveNet for enterprise management and CA application performance management tools for managing the Java application's behaviour.

JBossSX and McAfee provide security management.

Recovery is provided within the Oracle environment.

As can be seen, there are products from a number of different companies, all of which have to function together to deliver the services. Entirely different sets of products could be put together to do the same thing, or individual components substituted, for example using Oracle WebLogic and associated products rather than JBoss. And of course products are available to support the many other technologies not used in the above example.

As can be seen, there are products from a number of different companies, all of which have to function together to deliver the services. Entirely different sets of products could be put together to do the same thing, or individual components substituted, for example using Oracle WebLogic and associated products rather than JBoss. And of course products are available to support the many other technologies not used in the above example.

Software vendors test and release components following their own schedules. They typically test with a wide variety of operating system instances and a few different types of other products. Each also provides its own support and, in the case of faults, expects the using enterprise to know which product has the fault.

Consider next the Integrated Stack model. All the software and the machine are designed, developed, integrated and tested by the same vendor, an approach long followed by Unisys for the two ClearPath system families. Unisys provides the software necessary to support all the standards likely to be required by its clients. Even if some part of the software originates from elsewhere, it is integrated and supplied by Unisys, effectively as if part of its own product suite. IBM MQ, for instance, is integrated by Unisys into the ClearPath Forward stack, including the management of transactional integrity processes.

Figure 3 shows the major components that could be used if X has ClearPath Forward systems, either ClearPath Forward Libra or Dorado (or both) systems, or ClearPath Software Series. To avoid over-complicating the picture, the systems management, recovery and security components are not shown, but they are discussed in the text following the figure.

Specifically, two-phase commit processes can be used for both database changes and messages on queues. If a transaction cannot commit, any database or message queue updates are rolled back.
In figure 3, the MCP components are shown on the left with the OS 2200 alternatives on the right.

- All ClearPath Forward systems run on Intel hardware. Unisys firmware provides the Libra (MCP) or Dorado (OS 2200) system architecture: the instruction set, I/O connectivity and so on. For ClearPath Forward Libra and Dorado systems, Unisys supplies the Intel-based hardware and Unisys firmware. For ClearPath Software Series, Unisys supplies the firmware; the client chooses the Intel hardware.

- The existing applications run under COMS (Transaction Server – MCP) or TIP and HVTIP (Transaction Processing – OS 2200). The databases are DMS II (Enterprise Database Server for ClearPath MCP) for MCP, or DMS and/or RDMS (Enterprise Relational Database Server for ClearPath OS 2200) for OS 2200.

- Compilers and utilities are supplied with the systems. The compilers are trusted and will not insert malware, for example. The compilers generate code tailored for MCP or OS 2200 environments. The code they produce will not execute in Windows or Linux, nor will Windows or Linux exe files execute in the Libra or Dorado environments.

- For middleware, IBM MQ is implemented using WebSphere MQ for ClearPath MCP and ClearPath OS 2200 QProcessor. The web browser and mobile device access is provided using a ClearPath ePortal appliance.

- Transport and communications use the network services components for MCP and CPCommOS (Communication Platform for Open Systems) for OS 2200.

- The systems management tool chosen for operation is Operations Sentinel from Unisys. The product interfaces with other systems management products, for example OpCon from SMA and BMC ProactiveNet, an enterprise-level manager. A variety of recovery and security features are incorporated into both MCP and OS 2200 platforms, providing fast, reliable recovery if needed and protection against data violation.

Figure 3 shows just a subset of the range of products available; X could have chosen others in some cases, for example for middleware. Unisys tests all of its products together in each release and releases them as a unit. Users can be sure that the products in a release work with each other. Unisys also provides support for the system as a whole and will be the single point of contact for any error even if it may be in a third-party hardware or software component.
ClearPath Forward Deployment Options

The stack for ClearPath Forward Libra and Dorado systems comprises Unisys Intel-based hardware and firmware, and all the software. The systems would typically be installed in an organisation’s own datacentres, in much the same way that earlier ClearPath systems would have been deployed. The Unisys-supplied hardware platforms for ClearPath Libra and Dorado systems include options for high availability by duplicating some of the Intel servers used within the systems. Figure 4 is a schematic of a ClearPath Forward Libra or Dorado system.

![ClearPath Forward Schematic](image)

Figure 4: Schematic of the architecture of ClearPath Forward Libra or Dorado systems. The integrated stack includes Unisys Intel-based hardware, firmware and software.

In figure 4, the light blue boxes labelled PMM (Processor Memory Module), ISM (I/O Service Module) and OPS represent separate Intel-based components, that is, servers, interconnected through a high speed infrastructure. The upper part of the diagram contains the ClearPath MCP or ClearPath OS 2200 operating system and associated components. The firmware in the PMM maintains application code compatibility while the ISM firmware maintains data format compatibility. More than one server may be used for the PMM and ISM for resilience purposes – the high availability (HA) feature. The lower part of the diagram shows optional appliances: the Unisys ClearPath OS 2200 QProcessor, which implements IBM MQ, is within the ClearPath system in an Intel-based server; and the ePortal is in a Unisys-supplied Intel server, connected to the client’s network.

The ClearPath Software Series is relatively new, with some MCP systems available at the time of writing this paper. The stack comprises Unisys firmware and the software, which may run on any suitably configured Intel hardware. The systems may be deployed on bare metal or under a hypervisor; VMware and Hyper-V are supported. Software Series systems may be installed in a user’s datacentre in much the same way as other ClearPath Forward systems, apart from the hardware requirements. The Software Series also provides the essential basis for cloud deployment, either in-house or externally in a cloud service provider’s environment. Unisys will be working with partners to develop the additional infrastructure required for private and public cloud deployment. Announcements will be made as and when appropriate.
Discussion

The Multi-Source model has been promoted as offering choice of product, allowing the client to select a combination best suited to the problem at hand – the ‘best-of-breed’ idea. Choice is also believed to be a way of getting the best value, as competition among providers is assumed to drive down price. There is, however, an implicit assumption that quality is not compromised. And while the price may initially appear to be lower, several serious issues have arisen with the Multi-Source Model, leading to greater costs than expected as well as other complications.

- The task of integration is significant, frequently more than may be anticipated. Getting multiple products from different vendors to work together efficiently has proved to be a non-trivial task, especially in the more demanding environments where high levels of performance, availability and security are required. The activities necessary to install, tune, test and deploy the various components require a lot of effort, risk of error and therefore expense. New releases of any of the components, or a change of underlying machine, will require at least a regression test and possibly more tuning before deployment.

- It can be difficult to deploy multiple applications on a single server. Applications may comprise significantly different sets of components, some of which may be common to a set of applications, and others unique to an application. It is not uncommon for a component unique to one application to have a negative impact on another application which does not require that component. For example, the memory or CPU needs for a component may increase in unexpected ways, which in turn have a negative effect on the resources available to an entirely different application. ClearPath Forward systems are able to operate at close to 100% utilisation or more, for extended periods of time. Being able to host multiple enterprise applications within a single environment has many benefits.

- Auditing and logging can be difficult due to the plethora of different mechanisms and event types. Operating systems normally provide a repository for event data; each component product can use it or provide its own. Because the various products have to work in a variety of environments, some will maintain their own audit and logging files as well as using the operating system’s repository. Pulling all this information together to get a clear picture of what is happening throughout the stack can be substantially harder than with an integrated stack.

- Because the products originate from different suppliers, staff with an understanding of each product will be required. The result may be a group of people each one of whom only knows a part of the environment; nobody understands it all. This seems to be a common phenomenon².

- The diversity of product origin can cause problems with suppliers. It can be difficult to pin down the source of a problem, resulting in suppliers blaming each other. This problem is compounded by the timing of releases from different vendors, and the problems arising from incompatibility among the different releases. This is a significant problem, often leading to extended and expensive test effort.

- Each product has to be able to operate in combination with a variety of other products, as part of a different stack. There is therefore little or no scope for optimisation or removal of bottlenecks by moving functions between different components to achieve the best results because the products come from different sources. Recovery/resilience, security and performance are among important areas which benefit from optimisation. Where, for instance, should recovery be implemented: in the database manager, the application server, the operating system? A combination of all of them may be the best solution, but is nearly impossible to achieve if the software components come from different sources. Similar questions can be asked about security and performance, both of which can take advantage of function relocation in software or even in the underlying machine. As will be seen, ClearPath Forward systems have fully exploited such options.

- Auditing the quality of suppliers is difficult when there are many of them.

² The present author has encountered this problem in a number of organisations. If a ClearPath system has a software problem, any one of a small team is often able to fix it. Fixing a problem in a multi-source environment requires many more people.
The Integrated Stack model solves these problems. However, there are other possible intermediate approaches which lie between the two extremes of Multi-Source and Integrated Stack models. Two of them are examined below.

The first will be referred to as the **Integrator model**. In this model, specialist companies integrate software products from a variety of sources, test them and then deliver the resulting stack as a product, perhaps including the machine in which they run as well. Some application vendors follow this approach, supplying the application, the supporting software required and even the machine as a complete package.

The Integrator model solves a number of the problems encountered with the Multi-Source model. The integration effort required by the ultimate client is reduced and there is a single point of contact for any support; the integrator has to manage the relationship with the original suppliers. However, there is still restricted scope for optimisation. Although the integrator can tune the performance of the combined stack during volume testing, the tools available are largely restricted to adjusting the configuration parameters and settings of the various products. There is still very limited scope for moving functions between component products.

The second of the two intermediate models to be considered is referred to as the **Single Software Stack model**. In this model, a single company delivers all the software, integrated and tested, but produces or owns all the software itself. Microsoft is the most well-known of the companies following this approach. As in the Integrator model, there are the advantages of reduced integration effort by the ultimate client and a single point of contact for support. However, the Single Software Stack model has a significant advantage: because the supplier owns all the software, it can optimise it in ways not possible in the Integrator model, for example relocating functions to the optimum components.

The Integrated Stack model combines the advantages of producing all the software with the additional benefits from owning the machine. These benefits are not trivial: being able to strike the right balance in implementation between machine and software brings significant advantages, especially in mission-critical systems, typified by the ClearPath Forward user base, where high levels of performance, availability and security are essential. The following are some of the benefits flowing from this approach in ClearPath Forward systems; they are in no particular order. Without control of both machine and software, the benefits would have been difficult or impossible to realise.

- The machine and software are designed, developed, integrated and tested together, before shipment, enabling optimised performance, availability and reliability. The testing includes specific peripherals, certified for use with the systems.
- Security in ClearPath Forward systems is assisted by the machine characteristics. Data cannot be executed and buffer overrun problems – a common method of attack – are eliminated.
- Protecting data in motion is essential for secure systems. Both ClearPath Forward families support all the major encryption standards. OS 2200 software includes CryptoLib, a software library, and Cipher API, a subsystem, both of which provide application programming interfaces (API) for encryption, decryption, and related services. Cipher API provides access to a core subset of the CryptoLib algorithms and supports additional data formats. ClearPath MCP also offers a full set of encryption and cryptography algorithms for both system services and user-written applications. The cryptographic library has been offloaded into special environments (either a hardened appliance or the system firmware depending on system type) which are optimised, hardened and secure.
- Functions or functional areas can be relocated to an alternative component to improve performance or time to market. The integrated specialty partitions include the QProcessor for WebSphere MQ with OS 2200.
• ClearPath Forward systems have been able to distribute functions between the operating system and other components to optimise performance, in recovery processes for instance. The options for high availability have been further increased through the eXtended Transaction Capacity (XTC) for OS 2200, and the Business Continuity Accelerator (BCA) for MCP systems.

• Special instructions can be introduced to improve performance. An example is the queue management instructions introduced to optimise the communications system (CPCommOS) performance in OS 2200.

• Pay-for-use pricing models have delivered substantial cost/performance benefits. Metering has allowed systems to reduce execution times and handle unexpected spikes in demand while at the same time reducing costs. This approach is difficult to follow without an integrated stack approach.

• ClearPath Forward users do not need to change their applications with each system release. This is not a necessary consequence of the Integrated Stack model. It would be possible to use the model but make changes in a new release of the stack requiring application modifications\(^3\). An example would be changing an API employed by users’ applications. The integrated stack makes it possible to ensure that no application changes are required. Unisys additionally ensures that no changes are made in a new release of the stack that would require application changes. The ClearPath Forward approach could be regarded as an ‘Integrated Stack Plus’.

---

\(^3\) Some vendors do release integrated stacks that require application changes. The risk is that users will prefer to remain on earlier releases rather than change the applications, which can lead to security problems among other things.
## Conclusions

The table below summarises the characteristics of the two models as well as the two intermediate approaches.

<table>
<thead>
<tr>
<th>Stack model</th>
<th>Description</th>
<th>Key characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multi-Source</strong></td>
<td>Software and machine from multiple sources. Integration performed locally, on site.</td>
<td>• Extensive testing is required to ensure that all the components work together as required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• At least regression testing is needed with each new release or change of software. Different release cycles from the various vendors can lead to additional effort and cost.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The scope for optimisation by moving functions between components is very limited.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Auditing and logging are difficult due to the number of different products, each with its own approach.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Multiple points of contact for support as there are many vendors.</td>
</tr>
<tr>
<td><strong>Integrator</strong></td>
<td>Software and possibly machine from multiple sources. Integration performed by integrator, who then releases the tested stack.</td>
<td>• The testing effort for the ultimate client is much reduced compared with the Multi-Source model.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Some testing will be required for changes to machine if the integrator has not included the hardware.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The integrator’s scope for optimisation by relocating functions is very limited.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Single point of contact for software support.</td>
</tr>
<tr>
<td><strong>Single Software Stack</strong></td>
<td>Software from a single source, designed, produced, integrated and tested before release as a product.</td>
<td>• The testing effort for the ultimate client is much reduced compared with the Multi-Source model.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The supplier has substantial scope for optimisation by moving functions between components, assuming the whole stack is being deployed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Single point of contact for software support.</td>
</tr>
<tr>
<td><strong>Integrated Stack</strong></td>
<td>Software and machine from a single source, designed, produced, integrated and tested before release as a product. Software releases may also be made without new hardware releases.</td>
<td>• The machine <em>and</em> software are designed, integrated and tested together before shipment, reducing or eliminating the integration and test effort by the client. The approach also optimises performance, availability, reliability and security.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The Integrated stack is necessary to allow ClearPath Forward users to install new system releases without changing applications, reducing costs and simplifying deployment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Owning both the machine and the software can enable changes to optimise the complete system. It allows the machine to evolve as capabilities may be located in different ways to take advantage of technology advances. The machine/software stack brings benefits to existing code, and enables new feature content as new capabilities are introduced. Examples include moving functions between software components and implementing special functions in the machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Flexible licencing models and single support contact point.</td>
</tr>
</tbody>
</table>
Controlling all stages of design and development of both the software and machine enables the supplier to deliver the most cost/effective systems. This is especially true in mission-critical environments, as is the case for many ClearPath Forward systems; performance, availability, security and fast recovery really do matter. The Integrated Stack model, the benefits of which have long been realised by Unisys and its clients, has been adopted by vendors outside of those most associated with mainframes. Oracle is the most high-profile example; others may follow.

A final observation is that the benefits flowing from the Integrated Stack model complement the open attributes of ClearPath Forward systems. The result is systems which are powerful and secure transaction processing platforms, and are able to participate in distributed environments, including service architecture implementations. These characteristics differentiate ClearPath Forward systems from other platforms commonly viewed as open.

**About the Author**

Now an independent consultant, Peter Bye was a systems architect in Unisys, based in London. His special area of interest is networked computing, including communications networking, middleware, and architectures. He has many years of experience in information technology, working as a programmer, analyst, team leader, project manager and consultant in large-scale projects in banking, transportation, telecommunications and government. He has also worked in software development centres, during which time he spent two years as a member of an international standards committee working on systems management.

He has worked for extended periods in Sweden, Denmark, Finland, Norway, the USA, France and Spain, as well as the UK. He has presented at a wide variety of conferences and other events, and is the author of a number of papers on networking, systems management and middleware. He is the co-author of a book on middleware and system integration – *IT Architectures and Middleware: Strategies for Building Large, Integrated Systems* (2nd Edition) – which was published by Addison-Wesley.
More Information

Want to know more? Contact your Unisys representative.

The following White Papers provide additional technical information. They are published on the Unisys Website; the URLs are provided.

ClearPath Forward Systems: An Introduction
Delivering Value: the Economics of ClearPath Forward Systems
ClearPath Forward as an Open System

There are many other White Papers and a variety of additional information on all aspects of ClearPath systems. They can be found at White Papers.