E-WISDOM: Enterprise-WIde Software Development, Operation and Management

Current as on 23rd March 2004

I Introduction

Today, the requirement for secure, trusted, fast, reliable and distributed computing is ubiquitous. For a successful and modern Enterprise, it is imperative that its "digital cogs, wheels and gears" are in top condition not merely to fulfill the current objectives of the Enterprise, but also to meet its future challenges. These "cogs, wheels and gears", which form the core software and hardware of the Enterprise, must be bought either externally, or in the absence of a reliable external source, must be sourced internally.

In an Enterprise where Security, Reliability and Continuity of Progress are paramount, this is equal to an "everything sourced internally or everything outsourced" kind of situation. A mix-and-match approach will compromise Security and Reliability. In situations where external sourcing of software is either unreliable (e.g., the source may "dry" up due to economical, political or Intellectual Property Rights [IPR] related issues) or prone to risks (such as lack of security, poor breakable encryption, possibility of network - wired or wireless - eavesdropping, etc.), it will be prudent on the part of the Enterprise to fully source the software internally.

This research note explores the various possibilities in such a situation ("everything sourced internally") and analyses the probable avenues of redress, assuming that sourcing externally is **not** an option. The issues discussed in this research note will **not** be applicable in a situation where a trusted, reliable, agile and approved external software source is available for the Enterprise.

This entire research note can be divided into three broad regimes viz., Development, Operation and Management. Development relates to the actual coding (or, if available, the customisation of existing code) of the Software that can be deployed and shared across the Enterprise. Operation is the use of the developed (or customised) code in the whole Enterprise, to meet the objectives of the Enterprise. Management is the constant upgradation, refinement and fine-tuning of this Software. These three regimes can be viewed from different Perspectives, which are discussed below briefly.

The most important of these Perspectives are:

- The Hardware (HW) Perspective
- The Operating Systems Perspective
- The Application Software Perspective
- The Scientific Software Perspective
- The Interface (HW-SW-HW) Perspective (e.g., driver software)
- The Networking Software Perspective
- The Security & Encryption SW Perspective
- Use of Open Standards from end-to-end
- Distribution and Documentation of these SW solutions
- The User Support for HW and SW

- Building a EcoSystem around these SW solutions
- Updating, Maintenance and Refining the SW solutions

II The Different Perspectives

Though these Perspectives have been segmented for greater clarity, they are so interdependent on each other that not a single Perspective can be accorded less importance.

The Hardware Perspective

One of the most important aspects of this exercise is that the developed software must be compatible with every piece of hardware that is being used in the whole of the Enterprise. This perspective is also closely related to the Interface Perspective described below. Any Enterprise will have collected a multitude of hardware in the form of Servers, Desktops, Peripherals, Laptops, Notebooks, Smart Hand-held Devices and Mobile devices. The software development process must take into account the diversity of available hardware. This is not to say that the developed software must be one-size-fits-all piece of code; rather, the entire collection of software must be so robust that it takes care of this hardware spectrum.

The second aspect of this perspective is the expected growth of hardware in the years to come. The number and types of devices that might get connected to the Network in general (which includes Intranet, Extranet, Internet and other forms of wired and wireless networks) would increase exponentially. These devices must be fully compatible with the developed software.

The Operating System Perspective

This is the core of this development exercise. The Operating System itself comprises of many modules, including the Core System, File System, Input-Output System, Network System, Encryption, Security, Communications, Auditing, Logging, Installation, Updating, and Management and so on. Depending upon the target hardware, one or a few of these modules may be absent / not configured. These modules must be developed in such a way that (a) they function among themselves seamlessly and efficiently, (b) they drive the hardware spectrum discussed above, and (c) they operate the various application software discussed below.

The Application Software Perspective

Hardware and the Operating System together are not sufficient for an Enterprise. It requires a host of application software, which is aligned with the business objectives of the Enterprise. Most popular among these are the office applications, (editing and presenting documents, spreadsheets, etc.), communication applications, and simple database applications. However, for an enterprise-wide operation a number of application (non-scientific) software is essential. Prominent among them are data storage

software (in the form of a relational or hierarchical database), data access software (to view, edit, modify and delete data records, either locally or across a network), data mining software, enterprise resource planning software, computer-aided design, manufacturing and engineering software, etc.

These need to be developed so as to co-exist with the operating system and the host of hardware described above.

The Scientific Software Perspective

Depending upon the business objectives of the Enterprise, a number of Scientific, particularly *data processing*, software will be required. The term "data" is used here in the most generic form, which includes both structured and unstructured numeric and symbolic data. Typical data could be in the form of signals, images, audios, videos, column vectors, matrices, or any raw unstructured data.

Prominent sub-fields of scientific software are simply too many to be described here, but a quick list would give a flavor of what the task is: complete set of generalized mathematical operations, signal and image processing and analysis, cluster analysis, pattern recognition, database systems, knowledge-based systems, business intelligence systems (or decision support systems), virtual reality systems, visualizations, modeling software, and so on.

The Interface (HW-SW-HW) Perspective

Discerning readers would have noticed that this research note talks only about developing software (SW) internally, not about manufacturing hardware (HW) internally. That said, it is obvious that the Enterprise will be full of different types of hardware sourced from different vendors around the world.

In order to make these hardware work synchronously not only among themselves but also with the operating system (and other software), the HW-SW interface (prominent among them being device drivers) must be developed and tested.

This is not an easy task, particularly so if the Enterprise wishes to discard the OEM supplied device drivers and substitute them with drivers developed in-house, for reasons of security. There could be situations where this is not practicable; in which case, the source code of the OEM supplied device driver must be subjected to greater scrutiny.

The Networking Software Perspective

No hardware piece in future will operate in isolation; if it does, in all probability, that piece is obsolete! So, networking various hardware and software pieces will form a core task of this Enterprise.

The networking perspective is closely linked to the hardware and operating system issues discussed above, in addition to the Security / Encryption issues discussed below. It

would be natural to use the standard TCP/IP protocol as the basis with IPv6. More on the use of standards in the Section titled "Use of Open Standards from End-to-End."

The Security & Encryption SW Perspective

This is by far the most important aspect of this whole exercise. Apart from physical security, in a fully connected Enterprise data and software security commands the most attention and importance. The issues involved are:

- Secure Authentication
- Secure Authorization
- Secure Identity Flow
- Secure File Systems
- Secure Encryption and Decryption (128-bit or above)
- Obfuscation
- Code Security
- Prevention of Disassembling of Code
- Secure Point-to-Point Communication
- Secure Auditing and Logging
- Secure Inter-Process Communication
- Secure Protocols (e.g., https)
- Secure Certificates and Certifying Authorities
- Secure Signatures / Code Signing
- Error Correction
- Buffer Overflow Detection and Neutralization
- Type Checking

And so on.

Each and every aspect these issues must be addressed in the software that is developed and deployed. Constant vigil is required to check software malfunction; and these must be corrected using updates / patches.

Use of Open Standards from end-to-end

Fortunately, a number of open and evolving Standards have been identified for both software and data. Interoperability among these Standards is being established, so that the whole process becomes easier, to some extent.

Some obvious suggestions to follow:

- TCP/IP – Network Transport
- HTML – Data Markup Language
- HTTP – Hypertext Transport
- HTTPS – Secure Hypertext Transport
- XML – eXtensible Markup Language for Data Representation

- RDF – Resource Description Framework for Data Description
- UML – Unified Modeling Language for Software Specification and Development
- SOAP – Simple Object Access Protocol for transmitting data, particularly from one process to another
- UDDI – Universal Discovery, Description and Integration for processes to advertise themselves across a network or Internet

Most or all of these Standards can be used as they are, reducing the development time and efforts.

Distribution and Documentation of these SW Solutions

Documentation of the development efforts and processes, and the distribution of the software themselves, to the Users at the edge of the Enterprise, are equally important.

Though distribution (particularly, web-based distribution) is particularly easy, documentation could be overwhelming. Documentation should include version information, apart from benchmark results.

While software development can be streamlined using UML, documentation can be made in XML throughout, with appropriate XML Schemas defined. This would enable the reproduction of documentation in a device/browser agnostic fashion.

Documentation can also be located as a web service, which is accessible to a software processes that require them.

Enterprise-wide distribution can be achieved easily through multiple, mirrored, geographically-separated yet synchronized, high-speed servers that authorize software downloading after appropriate authentication.

The User Support for HW and SW

In spite of the best of documentation and descriptive, step-by-step procedures, users still need to point of support. Though web-based support is acceptable, depending upon the type of business the Enterprise is in, a human being manned support Center is advisable.

Within the intranet, this can be made accessible via a number of streams including Email and Voice.

If the set of software and their deployment is standardized, and if the Intranet has broadband connectivity, an Interactive Voice Response System (IVRS) can be commissioned entirely on a Voice-over-IP (VoIP) channel.

If the size of the Enterprise is too large, spanning several time zones, then, a 24x7 Call Center would be appropriate.

As User satisfaction and feedback are crucial for the success of these ventures, the importance of Support Desks / Centers cannot be over emphasized.

Building a EcoSystem around these SW Solutions

In order to allow measurable and sustained growth of the software usage, it is essential that a good ecological system consisting of Developer, Administrators, Managers, Users and Customers (of the Enterprise) is developed over a period of time.

Such an ecosystem would go a long way in getting precious feedback on the performance of the developed software from different viewpoints, which would otherwise be difficult to get.

Typical components of such an ecosystem would be:

- Newsletters
- Bulletin Boards
- Electronic Groups
- Printed Newsletters
- News Alerts on new software development
- Dissemination of Software Update information
- Publication of Whitepapers

Etc.

Third party benchmarking of the developed software against existing and commercially available similar software would be a great eye-opener, provided the security concerns of the Enterprise are addresses appropriately.

Updating, Maintenance and Refining the SW Solutions

It is very important for the Enterprise to keep its eyes are ears open to anticipate, or at least observe the winds of change that blow across the jungle of software development. One needs to be particularly adept in identifying concepts that are bound to change the way software evolution takes place. Web Services, XML, SOAP, TCP/IP, RDF, Server-side computing, n-tiered or layered computational approach, relational database management systems (RDBMS) are all such ground-breaking concepts. And there are many more being conceptualized everyday.

Conceptual changes are easy to identify, yet require constant attention to the changing landscape of software development. The furious speed with which hardware changes are taking place (particularly in increasing the speed, power and utility, and still reducing the size and cost) is bound to affect even a well-established and streamlined software development process.

In this scenario, keeping developed software updated (with respect to both identified bugs and emerging hardware), and refining the solutions (and algorithms), is mandatory. Otherwise the tide of software and hardware evolution would sweep these software solutions away in a matter of months.

III Future Scenario, Summary and Conclusions

The future will witness an explosive growth in (a) mobile devices, (b) data, (c) knowledge, (d) types of hardware and (e) network traffic. The emphasis will be to store everything, search everything and to control everything. Decision making will be fault-tolerant and distributed. The Enterprise must be aware of these trends in order to develop software solutions that pass the test of this emerging diversity in data, methods and devices.

This research note may sound ambitious; yet, it is immensely achievable. It is achievable if the Enterprise is clear of its objectives and if it has the courage to stay focused when initial failures and cost and time overruns threaten the nascent development processes.

Still, this is not a one-shot solution that Version 1.0 of the developed software will stay and serve for eternity. To stay in the same level of competence, the Enterprise must keep running; it must innovate, redefine its goals, refine its processes and always stay ahead of that technological inflection point.

Otherwise the whole exercise will be a waste of effort and time.