

The 64-bit .NET e-portal applications for e-Learning and e-Science

Victor Chang
School of Electronics and Computer Science
University of Southampton

1. Abstract

This is one of the first comprehensive 64-bit .NET research to be completed in a British University. The objective is to investigate (1) the techniques to develop 64-bit applications; and (2) how 64-bit computing can be used for e-Learning and e-Science communities.

This paper is focused on an emerging technology for e-Learning and e-Science communities -- a 64-bit .NET e-portal application, and its functionality, development and technical implementation are described. Currently, additional troubleshooting is required to make .NET Framework 1.1 working in stable ways on 64-bit Windows operating systems. After stabilizing .NET environment, the next step is the application development, which is a dynamic repository with functions such as registration, download, login-logout, product submissions, database storage and statistical reports. The technology is based on Visual Studio .NET 2003, .NET 1.1 Framework with Service Pack 1, SQL Server 2000 with Service Pack 4 and IIS Server 6.0 on the Windows Server 2003 Enterprise x64 platform with Service Pack 1.

The 64-bit .NET e-portal integrates further with Java Web Services, and a demonstration described in this paper can show how different types of services can work in harmony. Furthermore, the interdisciplinary area between e-Learning and e-Science has been investigated and in the process of further integrations. As an example for such integration, the 64-bit .NET e-portal is also an illustration for (1) the methods of technical implementations and (2) cost-effectiveness; and these two issues are also among the current research interests in e-Learning. Finally, the new insights identified by this paper have also achieved the two objectives: (1) learning about emerging technology and (2) understanding how Web Services can complete jobs.

2. Introduction

32-bit computing was first introduced in 386 models in 1985 but a surge of high demands began after the launch of Windows operating systems since the mid-1990s [12]. 32-bit computing revolutionized the world in science and technology, and also changed our life styles and career options [19]. Higher software and hardware demands for visualization, performance, speed, security, interoperability, usability, database integration and easy connections to the internet have been required since the 1990s [1, 2, 7, 12, 14, 19]. The term “bit” refers to the amount of data the computer can manipulate as a single slot. A 32-bit processor can perform operations on slots of data of up to 32 ones and zeros, and a 64-bit processor can handle slots of data up to 64 digits. This implies that a 64-bit CPU can perform better than a 32-bit CPU, and the assumption is strongly supported by the comprehensive testing outcomes from the first 64-bit academic research [3]. In addition, there is another difference between 32-bit and 64-bit computers-- a 64-bit computer can support up to 16 TG of virtual memory whereas a 32-bit computer can support up to 4 GB of virtual memory [3, 17]. Thus, a 64-bit computer is designed for higher computing demands where high consumptions of CPU and memory are required, and the assumptions are evidently supported by the stress testing and application testing [3].

With the release of 64-bit Linux and Windows operating systems between 2004 and mid-2005, many more 64-bit computers were manufactured, leading to the growing understanding of the 64-bit hardware configurations. However, this does not apply to the software technology, because few software applications have been built on the top of the 64-bit implementations. Even when 64-bit software applications are available, their purpose is to assist the operating systems to run smoothly, or to investigate the research implications of running 64-bit applications. An excellent example is the Sun Java technology, which provides the 64-bit 1.5.0 versions for Intel and AMD machines, and also for Windows, Linux and Solaris operating systems (OS) [9]. This enables the developers and testers to run the Java Development Kit (JDK), or execute Java programs on these 64-bit platforms, thus making Java as an interoperable and platform-independent web service language [10, 11]. However, there are few Java-inclined programs in the market, developing and building on the top of the 64-bit Java and 64-bit OS.

.NET is another interoperable and platform-independent web service language developed by Microsoft [15, 16]. Similar to JDK, .NET has its own version and “run time”, which is known as .NET Framework [15, 16]. Currently the most stable version is .NET Framework 1.1 with Service Pack 1, which can function most effectively on the traditional 32-bit

machines with Windows OS. Unlike Java and its JDK, .NET Framework has known compatibility problems with the Internet Information Service (IIS) Server on the current 64-bit machines and Windows OS. Therefore, it requires additional troubleshooting before stabilizing the .NET environment, depending on the levels of complexity of the .NET applications. In the author's experience, it ranged from one hour to three days, and the most difficult one was to get two different .NET Framework versions to co-exist without interferences to all .NET-inclined programs and servers.

Currently, there is a very limited understanding of 64-bit software, and how this can assist the organizations in improving their work. There are several problems associated to this, and the main reason is related to the usability issue -- "not sufficiently knowing how to properly make uses of 64-bit computing" [3]. Seeing this as a potential research area, the focus for this paper is on software implementations, rather than the hardware testing of Windows OS, with the two major objectives as described. The first objective is to present the techniques to develop 64-bit applications, and to simplify the process of development, because there is currently a lack of knowledge about development techniques. The second objective is to discover how 64-bit applications can make contributions to the existing e-Learning and e-Science communities, each of which has thousands of researchers and developers and is considered as an influential group in the United Kingdom. In order to achieve these two objectives, this paper is mainly focused on (1) the development process, which also includes the right combinations of technology and (2) types of "services" useful and relevant to e-Learning and e-Science communities. Both of the objectives are achieved, and are available, downloadable and presentable from the 64-bit .NET e-portal.

2. Development process: Application development on 64-bit .NET e-portal

Having the right combination of technology and developer platform, is the first step towards the success. The technology is based on Visual Studio .NET 2003, .NET 1.1 Framework with Service Pack 1, SQL Server 2000 with Service Pack 4 and IIS Server 6.0 on the Windows Server 2003 Enterprise x64 platform with Service Pack 1 after successful troubleshooting. The application is based on a dynamic repository that has functions such as registration, download, login-logout, product submissions, database storage and statistical reports. The functionality and technology-behind this, are described in Section 2.1 and 2.2 of this paper.

2.1 Functionality and implementations

.NET ASP.NET 1.1, VB.NET and SQL queries are the main development languages. The coding technique is one static ASP.NET in-front and one dynamic VB.NET code-behind. Visual Studio .NET 2003 is the platform for programming, compiling and testing. The term “repository” refers to dynamic pages that are presented as a browser-interface to the real database. The following is the description of the repository functionality.

(a) Registration: This is the first-step for all users. The three javascripts on the client side check whether the right information and all the “blanks” have been filled. On the server-side, each blank represents the data property for each entity, which is then processed by the dynamic .NET code upon submission. Upon successful registration, all the blanks are saved and stored into the database table “registered users” of SQL Server, and each entity on the database table has its own properties and permission. The .NET codes are separated into static and dynamic codes. The static code is the interface and the dynamic code creates a .NET function known as “register”, and can be configured or compiled in the Visual Studio .NET 2003.

The process of registration is divided into two stages, for which the earlier description refers to stage one. In stage two, another .NET function known as “registerok”, checks the status of the registration and recommends what users should do next based on the registration status. For successful registration, an email with a randomly-generated password is then sent to the user via SMTP protocol and recommends the user to login to the secure site. For unsuccessful registration, the repository asks for re-registration, or to receive a new password again.

(b) Login-logout: This technique is different than most .NET applications, because there is no .NET function dedicated to this. A .NET-authenticated directory, known as “repository” is created. All the .NET pages that require authentication are all stored safely under this directory. .NET authentication is achieved by configuring the “web.conf” file, which specifies which web directory is password-protected under the IIS server and .NET Framework. Security can be further enhanced by selecting “Integrated Security”, comprising both .NET authentication and Windows authentication, under the control panel of the IIS server. There is a “Sign-out” method, which is a specific .NET function that can get all the .NET pages to logout the authenticated server instantly. However this is a two-

step process. The first process requires the users to confirm logging out, and the second process then executes the logging-out functions.

(c) Downloads: This is the most critical part of the repository because registered users can download the latest software releases before June 2005. Each release is dedicated to a .NET page, which has ASP.NET as the code-in-front and VB.NET as the code-behind. All the release .NET pages are stored within the repository directory, which has the integrated security turned on. When users request for downloads via browsers, the repository prompts for the password authentication. Upon providing a username and password, the .NET page checks whether this information matches with the password in the database table of the SQL Server. If the user does not provide the right information, the .NET page then responds back. If this fails more than twice, the .NET page provides an option to go to the “forget password” functionality, which then finds the users in the database and sends them their new password.

After a successful login, the user can select to download software, which has the .gz and .zip files as the standard format. The IIS Server has a list of files for integrated security, which also include .gz and .zip formats. The size for downloads is not restricted.

(d) Product submission: This is for registered users who wish to submit their software product. On the submission page, the users provide information about their software and the public URLs to download it from. There are four fields for this page, and each field is equivalent to an entity of a database table. The .NET page can also check whether the information provided matches the required properties. Upon submission, the users are asked to accept the organizational terms and conditions. Whether the submission is successful or not, the .NET page can inform the user online about their submission status. For unsuccessful submission, the .NET page can get into the SQL server, retrieve the error message and display error message online. For successful submission, the submission information is safely stored in the database table and displays each submission status on another .NET page, which can only be seen and managed by the administrator. The administrator can choose to accept or reject the submission without downloads. If it is decided to download the software, the software is then downloaded on a directory called “Quarantine”, which is then checked by anti-virus software to ensure its safety. Upon successful checking, the software is then passed to the right person for further evaluation, which is an internal process within the organization.

(e) Database storage: SQL Server 2000 is the default database and its Service Pack 4 has 64-bit database application. There are three database tables created to facilitate the needs for (a)-(d), and one table holds all the information for the registered users, which can be obtained by performing SQL queries within SQL server. There are also four “Stored Procedures” (SP) to enhance authentication and database queries. SP are used to encapsulate a set of operations or queries to execute on a database server [5]. The syntax of SP looks like this:

```
dbconn = OpenOMIIDatabase() 'Now Open the database
and check the password
sqlcmd = New SqlCommand("CheckPassword", dbconn)
sqlcmd.CommandType = CommandType.StoredProcedure
sqlcmd.Parameters.Add("@password",
SqlDbType.NVarChar).Value=password
sqlcmd.Parameters.Add("@email",
SqlDbType.NVarChar).Value = user
```

```
CREATE PROCEDURE [CheckPassword]
    (@Email      [nvarchar](50),
    @Password    [nvarchar](8))
AS SELECT * FROM
[OMII].[dbo].[RegisteredUsers] WHERE
Email LIKE @Email AND
>Password=@Password
GO
```

(f) Statistical reports: The IIS server generates daily reports, which record information on the visitors’ online activities. Cygwin is used to retrieve the required information, and it is then summarized into a log file. Two software applications are used to generate website reports. The first is Analog, which is useful to generate the overall report that displays the visited files, domains and IP of the users and also browsers/OS used by the users. The second is WebLog Expert, which is particularly helpful in dealing with dynamic applications. It can track down the users’ IP/domains, files to download, number of downloads, dates/times of visits and reports any abnormal activities.

2.2 The entire flow and architecture for the dynamic .NET pages

The figure below shows the entire flow and architecture for the dynamic .NET pages. All the .NET applications and developer platforms can be installed on a 64-bit laptop and can function well without connecting to the internet, similar to a mobile server.

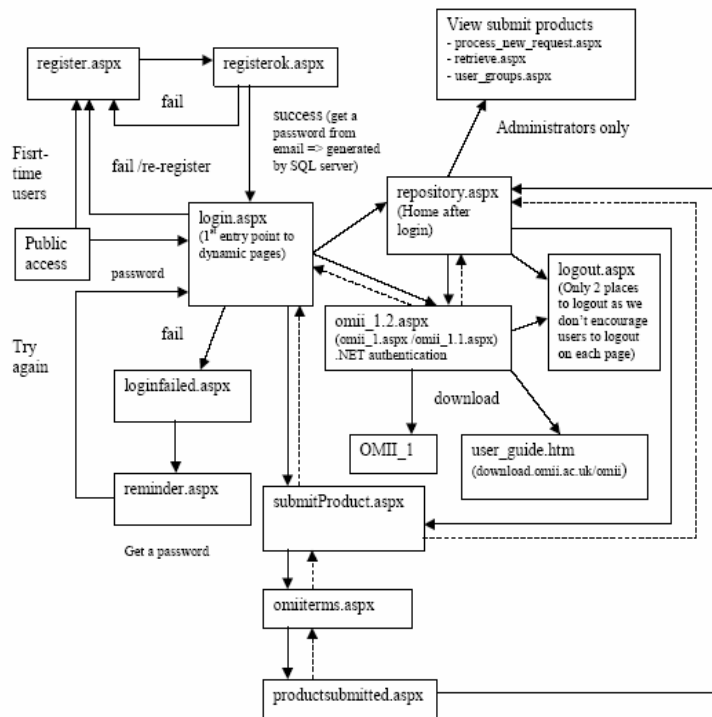


Figure 1: The flow and structure for 64-bit dynamic repository

2.3 Development process: using .NET Framework 2.0 and Visual Studio 2005

The next stage is to investigate the possibility of implementing .NET applications based on the latest Microsoft technologies. This includes Visual Studio .NET 2005, SQL Server 2005, .NET Framework 2.0 and IIS Server 6.0 on the Server 2003 Enterprise x64 platform with Service Pack 1. Due to incompatibility of IIS Server 6.0 and .NET Framework 2.0, it is highly likely to result in the instability of the Server 2003 Enterprise x64. Since it is possible to run 32-bit operating systems (OS) on the 64-bit machines, the alternative is to install these applications and develop them with the 32-bit OS on the 64-bit machines. Hence the implementations cannot begin unless (1) the developer and OS platforms on 64-bit computing become more stable and robust; and (2) software to support 64-bit computing becomes widely available and bug-free. However, this combination of technologies can function on the current 32-bit Windows Server 2003 [4].

3. Types of “services” for e-Learning and e-Science communities

Before moving onto the main topic, “Areas of common research areas and objectives” and “Contributions for e-Learning” need to be introduced.

3.1 Areas of common research interests and objectives

It is common to see different research subjects or projects merge together, to form into an independent area, or an emerging technology. An excellent example of this is the mobile-learning, which combines the mobile communication, education and information technology into a stand-alone research [6, 26]. The next example is Bioinformatics, which includes the integration of areas such as Biology, Engineering, Information Technology (particularly e-Science applications) and e-Learning [8, 13, 18]. There are no exception for the integrations between e-Learning and e-Science, where Web and Grid service architectures were firstly adopted and widely-used in e-Science but were recently applied into the e-Learning applications [20, 22]. Grid services were originally conceived as a technique in distributing high performance computation, as well as knowledge and resource sharing through the use of robust and secure Grid architectures [18, 23]. Such a functionality offered by Grid services was also adopted in the e-Learning communities, as demonstrated in the case of the Virtual Research Environment [4, 5].

3.2 Contributions for e-Learning: Technical implementations

Based on the summary of several scholarly research papers, there are two main streams in the current e-Learning research activities. The first main stream is about the research of educational implications and improvements, which include (1) the use of ICT for education, focusing on the process of learning; (2) the pedagogy of e-Learning and (3) the policy and best practices for e-Learning. The second mainstream is related to the learning technology, which includes (1) seeking technological solutions to match emerging theories of learning and (2) investigations of the educational affordances of emerging technologies. An important element and driving factor for the second mainstream, with regards to the methods of technical implementations, which include “the uses of emerging technology”, “the integrations of different technologies or standards” and the “cost-effectiveness of the implementations”. The earlier example, 64-bit .NET e-portal, is an illustration for the methods of technical implementations. It is cost effectiveness due to (a) the short period for implementations; (b) the free evaluation software from Microsoft and Redhat and (c) no maintenance cost, apart from the fact of purchasing 64-bit machines.

3.3 Types of services for e-Learning and e-Science communities

“Services” are commonly used in e-Science communities, and often it comes with the term “Web Services”, implying that the automated resources are accessed via the Internet. Web

services are software-powered resources or functional components whose capabilities can be accessed at an internet URI. Standards-based web services use XML to interact with each other, which allows them to link up on demand using loose coupling. [24]. The current Web Services standard is Web Service Description Language (WSDL) Version 2.0, a standard initiated and supported by W3C [25].

In order to justify the definition for “Services”, qualitative-based research is adopted and the interview, either face-to-face or via phone/email, is the research technique. Several academic researchers working in this area were interviewed, and the summary is that “Services” may imply ‘the type of functions that specific “Web Services” can offer’. This phenomenon is very applicable to OMII_2.0.0, which is a collection of tested, documented and integrated software components that provide a standard platform for integrating e-Science middleware as well as a simple, secure web service-based Grid infrastructure for new e-Science users [21]. The OMII_2.0.0 server provides four major services for the e-Learning and e-Science communities: account service, job service, database service and the resource allocation service. In order to test whether these services work, the multi-platform testing was carried out, which included creating an account on the server, obtaining the account approval and ensuring the job submission was completed.

3.3.1 Testing and integrating with Application and Java Development Kit (JDK)

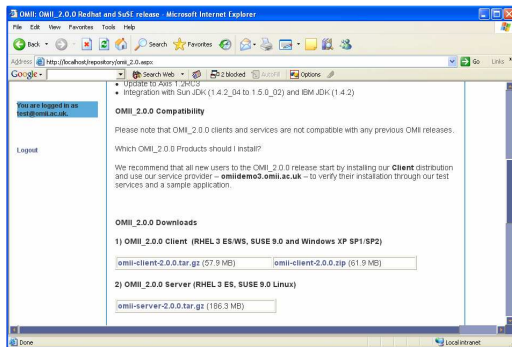
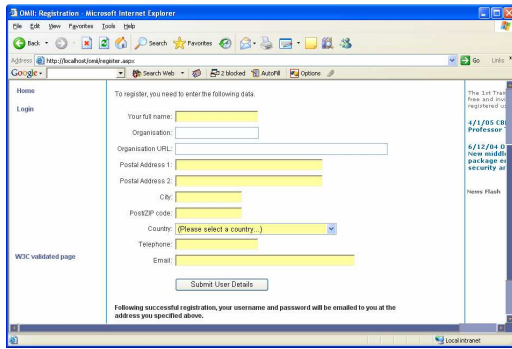
The comprehensive testing results prove that the OMII_2.0.0 server is able to work with several JDK versions, ranging from Sun 1.4.2_01 to Sun 1.5.0_02. The OMII_2.0.0 application, also known as Cauchy, was the core component used to test whether JDK versions were running smoothly. Upon the successful running of Cauchy and JDK, the software prompted for PlotWS services, which then plotted the graph from the requested job. There was a new feature, Graphical User Interfaced Cauchy (GUI-Cauchy), which simplified the entire process with a few clicks on the GUI-Cauchy itself. As a self-explanatory software with friendly interfaces, it presents the complicated “Web Services” as a visual demonstration with clickable buttons to execute the requested job, which makes it easier for users to understand the entire process of running different services. Thus, the e-Learning and e-Science communities were given the new insights, which include (1) learning for emerging technology and (2) understanding how Web Services can complete jobs.

4. 64-bit .NET e-portal: Integration of .NET and Java Web Services

The next task is to investigate the possibility of integrating different types of Web Services, and this is a topic with growing interests. In this paper, the integrations do not imply combining services such as data service, account service and job service, but imply the integrations of the fundamental part, or the base of the Web Services, which means the two main stream, .NET and Java. Therefore the integration of .NET and Java is the focus of the research theme. The 64-bit .NET e-portal can achieve this by the following techniques: (1) developing the e-portal on the .NET Framework and .NET-dependent platforms; (2) running the software that are JDK dependent; (3) ensuring the 64-bit OS are supporting the 64-bit dependent .NET Framework and JDK and (4) thoroughly testing the e-portal and software. Since the first three techniques are mentioned in the earlier part of the paper, the fourth technique, testing the e-portal and software, is described as below.

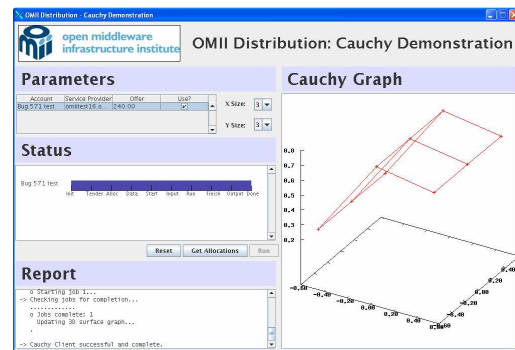
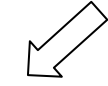
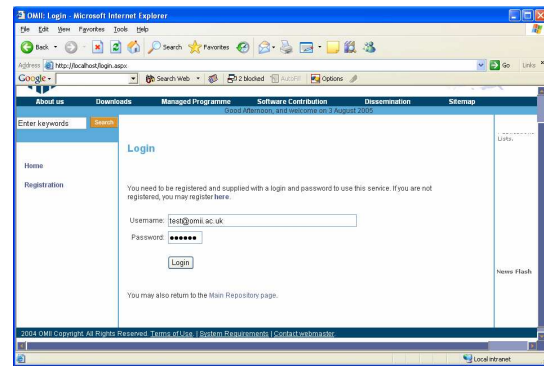
The 64-bit e-portal was first tested under the “local host” environment on the 64-bit machines and laptops – which could be functioned independently, or with the combined uses of broadband, LAN or wireless network. The 64-bit e-portal was then tested on different platforms such as 32-bit/64-bit Windows and 32-bit/64-bit Linux, and followed by a cycle of testing and quality engineering process when bugs were identified. Upon completing the testing cycle, the 64-bit e-portal can be ported to and configured on an ISP domain, so that it can function as a platform-independent “Web Service” – which is available, downloadable and presentable on the 64-bit .NET e-portal. Figure 2 on the next page shows the sequence of using the combined .NET and Java Web Service:

Step 1



Step 3

Step 2



Step 4

This demonstration above shows there are four steps, working in the chronological order. Based on the .NET Web Service, the first three steps are “registration”, “login / authentication” and “download”, and the details of how these three steps work, are described in the earlier part of the paper. The final step is based on the Java Web Service, and it is about running the Cauchy application of OMII_2.0.0. The screenshots show how the data service, job service, account service and resource allocation service of OMII_2.0.0 can work on the 64-bit Windows and Linux clients.

4.1 64-bit .NET e-portal: Interoperable on 32-bit and 64-bit Windows and Linux OS

The 64-bit .NET e-portal is capable of running on the 32-bit and 64-bit Windows and Linux OS, and this is strongly supported by the thorough software quality engineering and testing. The following table sums up in what platforms that this 64-bit .NET portal can work.

32-bit Machines: Support only 32-bit OS

Windows: client	98, NT, 2000, XP Home, XP Pro, Media Centre, Server 2003
-----------------	--

Linux: client	SuSE 9.0, Redhat Enterprise 3 WS/ES/AS, Redhat Pro 9.0, Fedora Core 3
Macintosh: client	Macintosh 10.x

64 bit Machines: Support both 32-bit and 64-bit OS

Windows: client and server	All 32-bit Windows (client) XP Pro x64, Server 2003 x64 (client and server)
Linux: client and server	All 32-bit Linux as stated on the table above (client) SuSE 9.2, Redhat Enterprise 3 WS/ES/AS, Fedora Core 3 (Server-side applications require additional troubleshooting and debugging)

5. Conclusion

This paper presents a new technique for technical implementations -- the .NET 64-bit e-portal, which is the combined result of three aspects. Firstly, it is the technical integrations of the 64-bit hardware, the 64-bit OS, .NET Framework 1.1, Visual Studio .NET, SQL Database Service, Internet Information Services and Report Services. Secondly, the 64-bit .NET e-portal has achieved part of the .NET and Java Web Service integrations. Thirdly, it provides the platform support and software to enable web services in action, so that e-Learning and e-Science communities can benefit the fact to (1) learn about emerging technology and how different services can work in co-operation, and to (2) use Web Services to complete the desired task.

The usability issue -- “not sufficiently knowing how to properly make uses of 64-bit computing”, is currently on further research work. This paper presents the software implementation techniques for the integrated .NET and Java Web Services, and the outcomes of this research can contribute to the improvement of the usability issue.

References

1. Adams M., Staron D. et al, “Security Complete”, Sybex, ISBN 0-7821-2968-4.
2. AMD White Paper, “x-86 64 Technology White Paper”, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/x86-64_wp.pdf

3. Chang V., "Two research contributions in 64-bit computing: Testing and Applications", IADAT-tcn2005, International Conference on Telecommunications and Computer Networks, September 7-9 2005, Portsmouth. UK.
4. Hall W., Hugh D., et al "A Grid Services Implementation for a Virtual Research Environment", 1st International ELEGI Conference on Advanced Technology for Enhanced Learning, Napoli, Italy.
5. Hall W., Hugh D., et al, "Towards Grid Services for a Virtual Research Environment". The 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan.
6. Hentea M, "Multi-Agent Security Service Architecture for Mobile Learning", IEEE ITRE 2004 Conference, London Metropolitan University, London, UK (mobile comm.)
7. Hertzog U., "Linux in No Time", Prentice Hall, ISBN 0-130-31976-7.
8. The Integrative Biology Project, <http://www.integrativebiology.ox.ac.uk>, University of Oxford, UK (bio).
9. Java Development Kit Archive, <http://java.sun.com/products/archive/>
10. Java Web Services Developer Pack, http://java.sun.com/webservices/webservicespack_1.0.html
11. Java 2 Platform (platform independent), <http://java.sun.com/java2/whatis/index.html>
12. Lind K.S., "MCAD/MCSD: XML Web Services and Server Components Development with Visual Basic .NET", Osbourne, ISBN 0-07-222653-6.
13. Luck M et al "Agent Interaction for Bioinformatics Data Management", the paper for Bioinformatics programme of the EPSRC and BBSRC, UK.
14. Jr Lucas H.C., "Information Technology for Management 7th Edition", McGraw-Hill, ISBN 0-07-116967-9.
15. Microsoft .NET Homepage, <http://www.microsoft.com/net/default.aspx>
16. Microsoft .NET Framework Developer Centre, <http://msdn.microsoft.com/netframework>
17. Microsoft Windows Server Systems, "64-bit computing with Windows Server 2003", <http://www.microsoft.com/windowsserversystem/64bit/bulletin.aspx>
18. Moreau L et al "On the Use of Agents in a BioInformatics Grid", MyGrid Project Working Paper, <http://www.mygrid.org.uk>, UK
19. Mukhar K, Lauinger T. et al, "Beginning Java Databases", Wrox, ISBN 1-861004-37-0.
20. Okamoto T, Kayama M, "The Learning GRID for Knowledge Building", 4th IEEE Conference on Advanced Learning Technologies, Finland, 2004.
21. The OMII User Guide, accessible from <http://www.omii.ac.uk> .
22. Rodriguez J et al "How can the Web Services Paradigm improve E-learning?" 3rd IEEE Conference on Advanced Learning Technologies, July 9-11, Athens, Greece, 2003.
23. Tao F et al "Managing the Semantic Aspects of Learning using the Knowledge Life Cycle",. the 5th IEEE International Conference on Advanced Learning Technologies, Kaohsiung, Taiwan.
24. Web Services Definition (recommended by Google), [http://looselycoupled.com/glossary/web services](http://looselycoupled.com/glossary/web%20services)
25. Web Service Description Language (WSDL) Version 2.0, <http://www.w3.org/TR/wsdl20>, W3C.
26. Wilson, M. L., Russell, A., Smith, D. A., Owens, A. and schraefel, m. c. (2005) <<http://eprints.ecs.soton.ac.uk/11101/>> mSpace Mobile: A Mobile Application for the Semantic Web. Submitted to End User Semantic Web Workshop, ISWC2005, Galway, Ireland.