Data Collaborative Work Flow Model for Information Integration with the support of Web Services

J. K Zhang [†], K Liu[†], Y. Tang[†], M.C So[†], L. S. Ke[‡], A. L Mao[‡]

[†]Informatics Centre, University of Reading, U.K.
[‡]Airinmar Ltd, U.K.
jz@airinmar.com

Abstract – **In this paper, the author proposed a data collaborative work flow model with the support of web services. Nowadays, many E-Commerce businesses are relying on the data transmitted between businesses or their customers. In order to get the best value of data, the way of processing data is critical. So the author proposed a data collaborative work flow model with six stages (data fetching/transfer between businesses, data store, data refining, data analysis, data presentation, and data fade). The author believes the accuracy of data relies upon how the data is processed at the each stage in this model. In order to make this model more practical and understandable, it would be discussed with relevant case studies. While processing data, sharing data and applications between businesses is common. However, this raises some technical concerns in data transmission, data security, and network limitation. Among these issues, system interoperability are becoming more important since data and application integration is not an easy task due to differences in programming languages, system platforms, Database Management Systems (DBMS) used within different systems. To-date, a new distributed middleware technology 'Web Service' paradigm standardises the programmatic interfaces for application-to-application communication. To fully take advantage of this technology, this paper proposed the use of 'Web Services' to address the issue of interoperability. Therefore, another aim of this paper is to enable those businesses deploying Web Service to enhance their data processing with higher flexibility and efficiency.**

Keywords – **E-Commerce, Data processing, Data Collaborative Work Flow, Information Integration, Interoperability, Web Service**

I. INTRODUCTION

Long time ago, most of businesses treated data processing as a basic routine. But most of data are characteristically heterogeneous, dynamic, and time oriented and is contextually rich and complex. These features, together with needs to meet various business requirements, make the information management very expensive. It was estimated that the cost of data processing had comprised a high percentage of total business running cost. With the development of IT, E-Commerce businesses or applications is rapidly booming up. These businesses become more relying on the data processing and data exchanging between businesses and/or customers. Particularly for electronic commerce, data processing and data exchanging undoubtedly would account for a big daily work load. Accordingly, the success of electronic commerce depends on how well the data is processed. In order to more efficiently process the data, the author proposed a data collaborative work flow model. The proposed model comprises of six stages that are data capture/transfer between businesses and customers, data store/distribution, data refining, data analysis, data presentation, and data shrinking. In the main body of this paper, the author would explain how the proposed model works to make the data deliver its optimistic value.

Many businesses have intended to integrate their systems' functions and data. This intention raised some concerns, such as, data security, data transmission, network limitation and so on. Among these issues, the issue of system interoperability is becoming a major barrier for the integration of functions and data of different systems. In practice, systems were developed using different languages (e.g. Java, Visual Basic, C++, etc.), different system platforms (e.g. Microsoft Windows operating systems, Linux operating system, etc.) and DBMSs (e.g. Microsoft SQL server, Oracle, Microsoft Access, etc.). These differences between systems are major barriers for the system interoperability. If a system is poor in its interoperability, any increasing functions or little changes could stop it working properly.

A new distributed middleware technology, Web Services, plays a key role in this proposed model. Based on open standards, the Web services technology allows any piece of software to communicate with each other in a standardised XML messaging system. Web services that use SOAP support a wider variety of data types. The data types supported by SOAP include most basic data types, as well as Dataset, DateTime, XMLNode and several others. SOAP also permits the transmission of arrays of all these types.

II. DATA COLLABORATIVE MODEL

Although there are massive data processed daily by various businesses, it is not easy for them to get the optimistic value from each piece of data. The author believes how much a business could benefit from those data is heavily relying on how the data is processed in different time, such as, the time when the data is fetched or transferred to others, the time when the data needs to be sorted or analysed, etc. In that case, the author proposed 'Data Collaborative Model' to provide a methodology for dealing with the data at different stages. The proposed model concluded six stages for those data dealt in different time, illustrated as follows:

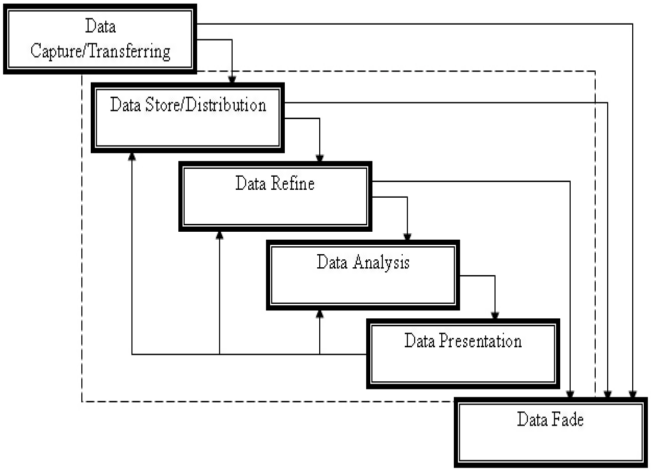


Figure 1. Data Collaborative Model

A. DATA CAPTURE/TRANSFERRING

Airinmar Ltd receives considerable data every day, such as, customers' orders, customers' enquiries, customers' comments, orders put to suppliers and communication between Airinmar Ltd and suppliers/customers. Some electronic mechanisms are deployed in this company to receive and transfer data, such as, FTP, computer-based applications, emails, Internet, etc. As discussed above, the proposed model is aiming to make the data deliver its optimistic value from each stage. In order to achieve this purpose, some data processing mechanisms are applied, illustrated in the following figure:

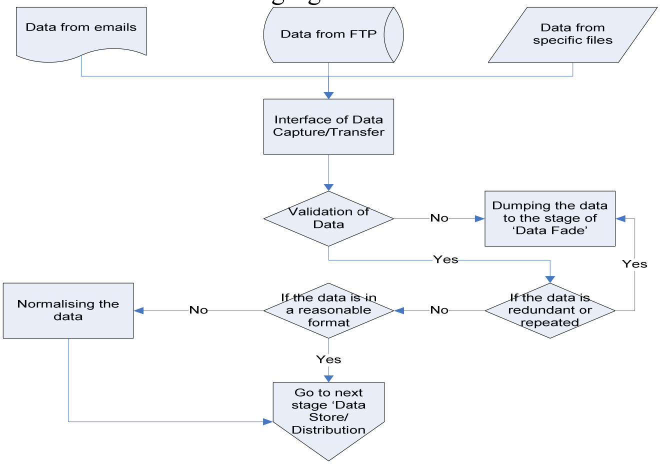
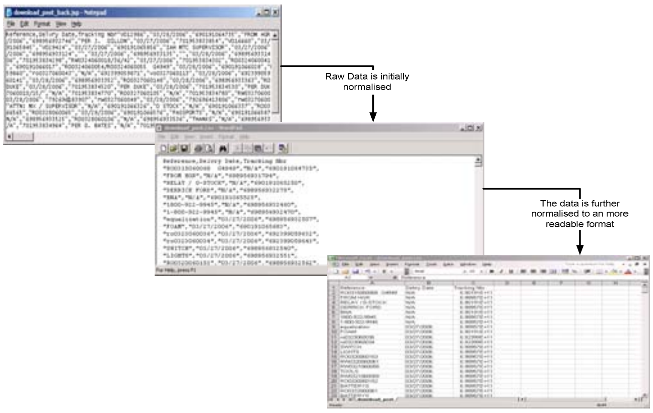

Data from emails
Data from FTP
Data from specific files
Interface of Data Capture/Transfer
Validation of Data
No
Dumping the data to the stage of 'Data Fade'
Yes
Yes
If the data is redundant or repeated
No
If the data is in a reasonable format
No
Normalising the data
Yes
Go to next stage 'Data Store/ Distribution

Figure 2. The procedure of data capture/transfer

At the stage of data capture/transfer, raw data will be from various data source like emails, FTP, etc. Traditionally, when businesses get the data from its customers or suppliers, they just pass and save them directly into the data storage. However, this method is not really appropriate in most of current situations, which would result in wasting the data storage and inaccuracy processing result for future stages. For example, different countries have different formats of data time. If they are not properly validated and converted to required formats, the result of data analysis would be messed up. So the raw data should be filtered in couples of steps in this stage. Firstly, once the data comes in from various sources, a validation mechanism should be applied to validate the data. If the data is found invalid, it would be taken to the stage 'data fade' to dump it. Secondly, the data would be checked if it has been redundant and repeated compared to the existing data. If so, the data should be removed and taken to the stage 'data fade'. Finally, a mechanism would be used to normalise the data before going to next stage since raw data might be received in an unreadable format.


Raw Data is initially normalised

The data is further normalised to an more readable format

Figure 3. The example of normalisation of data

The above figure represents a good example of data normalisation in Airinmar Ltd. Everyday, Airinmar Ltd receives some data from its customers in specific files, such as CSV files. Sometime, these data is arranged in a very mess format that is unreadable at all, like up left screenshot. For the further processing, the data would be firstly formatted in a reasonable order, like the middle screenshot. However, it is still not readable for those people who do not understand CSV file. As the result, it would be further normalised to be an appropriate format, such as a data table, spreadsheet like down right screenshot.

B. DATA STORE/DISTRIBUTION

After the stage of data capture/transfer, the data should be distributed to the data storage. This stage in the proposed model has a big weight since it could severely affect the result of following stages. Traditionally, the business may populate the data straight to the data storage. Although this method is very easy and fast, it could make following stages to be more complicated and difficult to handle. In this stage, the data should be re-arranged and categorised in order to satisfy any potential requirements for the further data processing before distributing them to the data storage. The following figure would illustrate it:

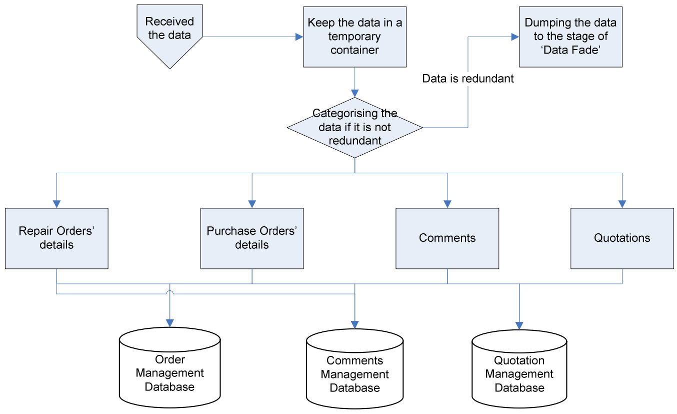
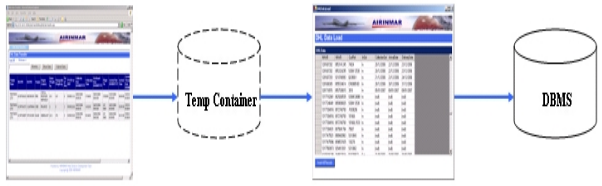

Received the data
Keep the data in a temporary container
Dumping the data to the stage of 'Data Fade'
Data is redundant
Categorising the data if it is not redundant
Repair Orders' details
Purchase Orders' details
Comments
Quotations
Order Management Database
Comments Management Database
Quotation Management Database

Figure 4. The procedure of data store/distribution

In Airinmar Ltd, after the data is received and processed in the first stage, it could be usually categorised into part repair orders, part purchase orders, comments from customers/suppliers and quotations from suppliers for the further processing in the following stages. Nevertheless, some duplicated and dummy data could be sent to Airinmar Ltd by mistake sometime. In this case, they would be straight dumped to the stage 'Data Fade'.



The application for the stage of data capture

The application for the stage of data store/distribution

Figure 5. The example for the stage of data store/distribution

The above figure illustrates an example about how Airinmar Ltd to deal with the data in the stage of data store/distribution. In fact, the company provides customers/suppliers with an Internet-based application to upload their data. This application applies steps specified at the first stage (figure 2) to ensure that all data received from customers/suppliers are correct for next stage. All of received data would be temporally kept in a temp container. And then, another windows-based application is developed for internal staff in Airinmar Ltd to process those data at the second stage. According to second stage's rules (figure 4), this application would firstly take all the data out from the temp container, and then filter, categorise and populate them into relevant databases for the next stage's processing.

C. DATA REFINE/DATA ANALYSIS/DATA PRESENTATION

Stages of data refine, data analysis and data presentation are tightly related each other, as the result, the author would discuss them together in the following section. Since last stage, most of data should have been correctly stored in DBMS. But in order to meet various requirements of data analysis, the data should be reorganised and refined before going any further. In Airinmar Ltd, there are lots of data analysis applications for different purposes. The author would take one of data analysis applications as a case to talk through following stages. The selected application is used to track increasing (decreasing) costs for each part number and repair type, and to take any corrective actions if necessary. Before doing the major analysis, this application would refine the data according to different conditions. The procedure of data refine for this application is shown in the following figure:

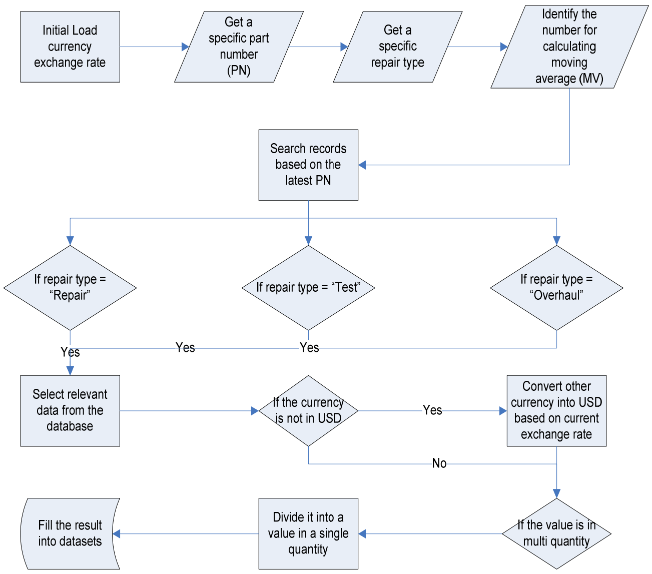

Initial Load currency exchange rate
Get a specific part number (PN)
Get a specific repair type
Identify the number for calculating moving average (MV)
Search records based on the latest PN
If repair type = "Repair"
If repair type = "Test"
If repair type = "Overhaul"
Yes
Yes
Yes
Select relevant data from the database
If the currency is not in USD
Yes
Convert other currency into USD based on current exchange rate
No
Fill the result into datasets
Divide it into a value in a single quantity
If the value is in multi quantity

Figure 6. An example for the procedure of data refine

The above figure describes how that application refines the data before doing the analysis. Main criteria of data refine for this application is based on customer accounts, part numbers, repair types and repair costs. Airinmar Ltd sets unique customer accounts for different customers. In this case, the data would be firstly filtered by different customer accounts. For some reasons, each part may have multiple part numbers. With respect to the requirement for this application, only latest one would be applied in the application. Each part has various repair types with different repair codes. Regarding to the requirement, the application would categorise them into three general repair types (repair, test and overhaul). Additionally, the value of repair cost is usually recorded in various currencies and multi-quantity parts. In order to ensure the accuracy and consistency of data analysis, all cost value would be calculated into US dollars in a single value. Finally, the data analysis could be executed based on these data. The following figure illustrates a whole procedure from data analysis to data presentation.

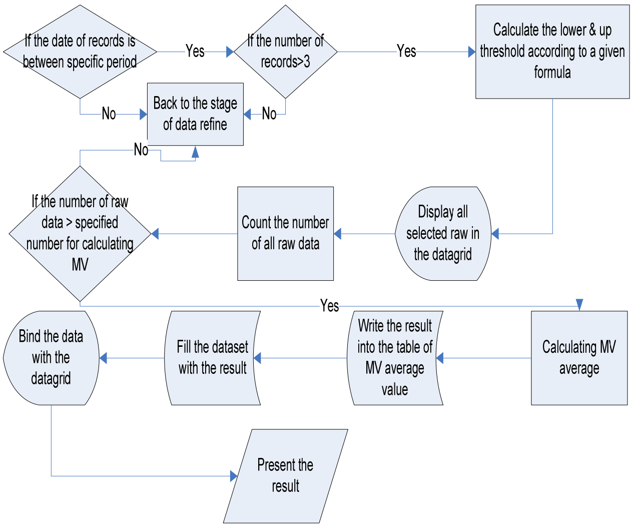

If the date of records is between specific period
Yes
If the number of records>3
Yes
Calculate the lower & up threshold according to a given formula
No
Back to the stage of data refine
No
No
If the number of raw data > specified number for calculating MV
Count the number of all raw data
Display all selected raw in the datagrid
Yes
Bind the data with the datagrid
Fill the dataset with the result
Write the result into the table of MV average value
Calculating MV average
Present the result

Figure 7. The example for the procedure from data analysis to data presentation

Before going to the stage of data analysis, the application still needs to find out if the data passed from the stage of data refine is good enough for carrying onto next stage. For instance, in figure 7 the data would be checked if it is within a qualified period and it has sufficient number of records for the analysis. If not, the application would roll back to previous stage to adjust the criteria for the selection of data. But if everything is sorted, the application would start to analyse and calculate the data based on specific rules and formulas as described in figure 7. Once the analysis and calculation is done, the result would be presented to end users in different forms like illustrated in the following figure:

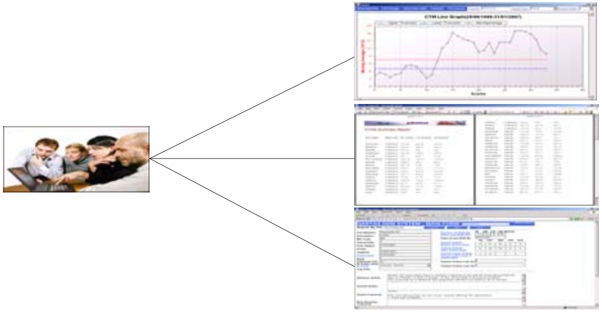


Figure 8. The example of data presentation

The result of data analysis could be presented in different forms in order to meet different requirements from users. In figure 8, the application presents the result in the line graph, the report in word document and the report in the web form.

III. THE ISSUE AND SOLUTION OF INTEROPERABILITY

A. THE ISSUE OF INTEROPERABILITY

The interoperability can be defined as “the capability with which two or more programs can share and process information irrespective of their implementation language and platform” [1-3]. The above section explains the procedure and data flow of proposed data collaborative work flow model. However, in order to implement different parts of the proposed model as a whole body, it is critical to integrate functions and data together in relation to the proposed model. In this case, it raises technical concerns like data transmission, data security and network limitation. Among these issues, interoperability concerns also rises as one of the most urgent priorities of enterprise information systems. Most enterprise applications are determined by a huge variety of heterogeneous and independent work places, most of them equipped with specialised hardware. These systems are also used by people with different levels of expertises and needs. Once the interoperability problem is solved, the development and maintenance of systems can be streamlined with data reuse, code reuse, application reuse and choice of an appropriate computing environment using object-oriented technology. The following content would explain the interoperability problem from different perspectives [1, 3]:

Database System Interoperability: records are often located in different database systems; however, data from different database systems cannot be exchanged with each other and deployed by applications based on different DBMSs.

Language Interoperability: Usually, since different systems are developed by different IT providers, developers may make use of different programming language to build their applications. This would make the reuse and share of applications between different systems very hard because of the incompatibility between different programming languages.

System-platform Interoperability: System platform interoperability means OS interoperability, but over the last few years the Internet browser has emerged as a platform in itself [4]. As different systems could be developed based on different development platforms, this feature would limit those systems only work on some certain system platforms.

Semantic Interoperability: some interoperability problems are caused by semantic differences. Semantic interoperability assumes that the components of the distributed application will have different meanings. Usually human intelligence is required to solve it, however this problem has been beyond the scope of this paper.

B. THE SOLUTION OF INTEROPERABILITY

The interoperability with applications and data from different systems is a challenging task. Traditionally, solutions of interoperability normally involve developing middleware applications to communicate the non-interoperable applications using the messages, of which the technology is also named as the distributed middleware technology [5]. CORBA and DCOM are two most typical distributed middleware technologies. Nevertheless, CORBA or DCOM is very complex, a task requiring special expertise. Quite often, the achievement of a good interoperability strategy is significantly constrained by many implementation restrictions in CORBA or DCOM [5-7]. Web services have emerged as the next generation of integration technology [6, 8, 9]. Based on open standards, the Web services technology allows any piece of software to communicate with each other in a standardised XML messaging systems[5, 6, 10]. It solves and eliminates above issues of DCOM/CORBA [6]. As a new type of software service, Web services are modular self-describing, and self-contained applications that can be published, located, and dynamically invoked across the Web. The Web-services technology is built on the foundation of open standards and common infrastructure. The Web-services framework is divided into three areas – communication protocols, service descriptions, and service discovery, of which each is specified by an open standard [2, 11].

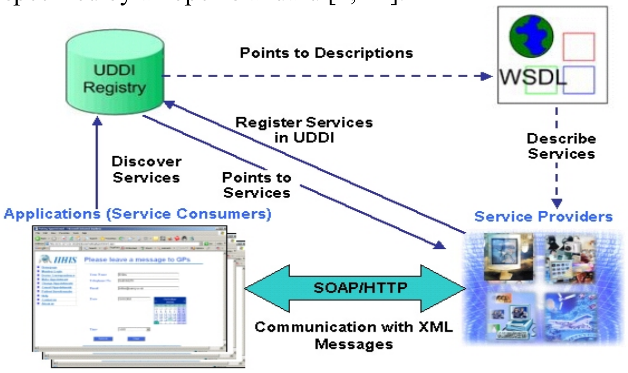


Figure 9. A detailed architecture of Web service

Figure 9 also indicates how web services works [2]. In general, web services consist of two major technologies (XML – extensible Markup Language and SOAP – Simple Object Access Protocol) and two assistant technologies (WSDL – Web Services Description Language and UDDI – Universal Description, Discovery, and Integration).

- Firstly, service providers would make use of WSDL to describe their web services
- Following the above step, service providers would register and publish their services in UDDI
- Applications or service consumers find services via UDDI which would direct service consumers to relevant services according to the description of web services
- Regarding to previous step, applications or service consumers are able to invoke relevant web services using SOAP transmitted via HTTP on the Internet.

Web services encoded in XML, SOAP provides a way to communicate between applications developed with different programming languages and running on different operating systems. In fact, Web services provide a distributed computing technology for integrating applications on the Internet using open standards and XML encoding. The use of standard XML protocols makes Web services platform-, Language- and vendor-independent, thus an ideal solution for use in application integration.

C. CASE STUDY - AIRINMAR LTD

The following case study is based on a company name as Airinmar Ltd that is founded around 90s last century. It is an independent company specialising in managing the repair of aircraft parts for many of the world's largest and most successful airlines. During past seventeen years, Airinmar Ltd developed lots of applications to manage those data about different parts of aircrafts. With the development of IT technologies, Airinmar Ltd aims moving to some new technologies, such as Microsoft .Net. Additionally, the company also intends to integrate some of applications into a whole system for the easy maintenance and expansion. However, the biggest obstacle of this objective and intention is the interoperability issue because of the following reasons:

- Different DBMSs: Within last seventeen years, there are various DBMSs applied in Airinmar Ltd to accommodate the data. There are still some seriously old DBMSs in use, such as, FoxPro database, Microsoft Access 97. In some departments, some unsuitable technologies are used as the data storage like Microsoft Excel that would become very unstable in term of performance and scalability when the amount of data is increasing. FoxPro and Microsoft Access have limitations to handle a large number of customers. In terms of robustness, scalability, security, data recovery and performance, both of FoxPro and Microsoft Access are all very weak. As the result, the company decided to migrate these data into an advance DBMS (Microsoft SQL sever) that would overcome those weaknesses from those old DBMSs. But this also leads to a big challenge for the integration of these data from different DBMSs.
- Different Programming Languages: Lots of existing applications in Airinmar Ltd were developed in various languages (e.g. VB, Java, JSP, etc.) by different developers. This situation makes the integration of different applications more difficult.

Web service is applied as a major solution for the interoperability issue in Airinmar Ltd. It plays as the role of middleware that hides all these differences in system platforms, programming languages and database systems to users and developers. Accordingly, from users' perspective, they would be able to get the access to the system regardless of their different system platforms. From developers' perspective, they can invoke or reuse applications built up in different technologies with the support of web services.

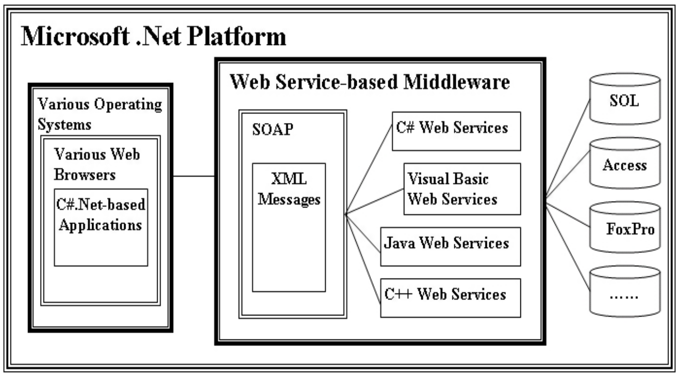


Figure 10. The way of Web service working in Airinmar Ltd with the support of Microsoft .Net technologies

Figure 11 explains how Web Service and Microsoft .Net technology are applied in Airinmar Ltd. The new system in the company is built upon Microsoft .Net platform, and Web service plays a role of middleware. Web services can be exposed from and consumed by any platform that can format and parse an XML message because using XML for the formatting of requests and responses. This allows XML-based Web services to bring together disparate pieces of functionality – existing or new, internal or external to an organisation – into a coherent whole. Core technologies of Web service are XML and SOAP. Once Web services receive requests from applications, web services would retrieve data from different DBMSs (e.g. SQL server, Microsoft Access, FoxPro, etc.) into datasets according to requirements of applications. All datasets would be written in XML messages, in additions, SOAP would act as an XML envelop to wrap those XML-based datasets into SOAP messages. And then these SOAP messages would be transmitted back to applications via HTTP. SOAP in Web service-based middleware provides a way to communicate between applications developed with different programming languages and running on different operating systems.

Additionally, Microsoft .Net platform is another important part of solution for the interoperability in Airinmar Ltd. It is language neutral. It is best thought of as an open programming platform into which a variety of languages can be plugged. It is achieved by translating all different programming languages into a common language called Intermediary Language (IL). Firstly, source code is translated into Microsoft Intermediate Language (MSIL). This IL code is language-neutral, and is analogous to Java bytecode. The IL code then needs to be interpreted and translated into a native executable.

IV. CONCLUSION AND DISCUSSION

In this paper, the author proposed a data collaborative work flow model to enhance the data processing for current businesses. With the growing businesses, the data is becoming characteristically highly complex, heterogeneous, dynamic, time-oriented and is contextually rich and complex. Under this situation, the data processing is becoming more difficult and critical. As the result, the author has an insight into this concern. With the real experience of data processing, a conceptual model is proposed. The author systematically divides this model into six stages. And each stage has its own logic and steps for processing the data. In this paper, the author explains this model with some practical examples to help various businesses easily understand how the data should be processed at different stages in order to let each peace of data deliver its optimistic value. But steps and rules for processing the data in each stage could keep changing based on different situations.

In order for this model to process the data from different sources and integrate applications built in different technologies, the interoperability issue is becoming another major concern of this paper. In fact, the interoperability issue also becomes a common concern for most of current businesses. For instance, some companies are keen to rebuild their systems with new computer technologies instead of old technologies. On the other hand, since most of legacy applications have been in use for many years, lots of users have been used to them. So companies could not directly get rid of original applications. In this case, a general solution is to let legacy applications and new systems run in the meantime for a while until all of data are completely migrated from legacy applications to new ones and users are fully confident with new systems. During this period, legacy applications and new systems would share some data and functions. However, the interoperability between legacy applications and new systems becomes a prominent obstacle. Only if the interoperability issue is solved, the integration of data and functions between legacy applications and new systems becomes realistic. In this paper, the technology 'Web Service' is applied as the major solution for the interoperability issue. Web service acts as a distributed middleware to allow any piece of software to communicate with each other in a standardised XML messaging system that significantly facilitate the interoperability of the whole model with the support of Microsoft .Net technologies.

ACKNOWLEDGEMENTS
Support from the KTP scheme- with grant reference number 1182.

Reference:

1. Pronab, G. and R. Pradeep, *Software Interoperability of Telemedicine Systems: A CSCW Perspective.* IEEE 2000.
2. Zhang, J.K., So, M.C., Liu, K., Tang, Y., Ke, L.S., Mao, A.L. *Web Service-based Solution for Enterprise Application Integration (EAI).* in *Educational Multimedia, Hypermedia & Telecommunications*. 2007. Vancouver, Canada.
3. Varge, B. and P. Ray. *Interoperability of Hospital Information Systems: A Case Study.* in *Enterprise Networking and Computing in Healthcare Industry, 5th International workshop*. 2003.
4. Albahari, B., Drayton, P., Merrill, B., *C# Essential - A Comparitive Overview of C#.* First Edition ed. 2001: O'REILLY.
5. Zhu, J., *Web Services Provide the Power to Integrate.* Power and Energy Magazine, 2003. **Volume 1**(Issue 6): p. Pages 40-49.
6. Banerjee, A., Corera, A., *C# Web ServiceS - Building Web Services With .NET Remoting and ASP.NET.* 2002: WROX.
7. Gokhale, A., Kumar, B., Sahuguet, A. *Reinventing the Wheel? CORBA vs. Web Services.* in *The Eleventh International World Wide Web Conference*. 2002. Honolulu, Hawaii, USA.
8. Chester, T.M., *Cross-Platform Integration with XML and SOAP.* IT Professional, 2005. **Volume 9**(Issue 4): p. Page 67-70.
9. Umar, A. *The Emerging Role of the Web for Enterprise Applications and ASPs.* in *Proceedings of The IEEE*. 2004: IEEE.
10. Deitel, H.M., Deitel, P.J., Listfield, J., Nieto, T.R., *C# How to Program - Introducing .Net and Web Service.* 2002, London: Prentic-Hall, Inc.
11. Curbera, F., M. Duftler, and R. Khalaf, *Unraveling the Web Services Web - An Introduction to SOAP, WSDL, and UDDI.* IEEE Internet Computing, 2002.