UTILIZING SAP’S SYBASE UNWIRED PLATFORM AND MOBILE BUSINESS OBJECTS TO CREATE A MOBILE PLANT MAINTENANCE APPLICATION.

by

ALLEN CARROLL

Advisor
DR. HALA ELAARAG

A senior research proposal submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in the Department of Mathematics and Computer Science in the College of Arts and Science at Stetson University DeLand, Florida

Spring Term
2012
ACKNOWLEDGMENTS

This was made possible due to the software technology provided by System Analysis and Program Development and the server access they so graciously provided to both myself and my advisor.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS ............................................................................................................. i

TABLE OF CONTENTS ............................................................................................................. ii

LIST OF FIGURES ................................................................................................................. iii

ABSTRACT .................................................................................................................................. 1

1. INTRODUCTION ................................................................................................................... 2

2. Related Work ....................................................................................................................... 4

3. Implementation .................................................................................................................... 10

4. Results ................................................................................................................................ 13

5. Conclusion .......................................................................................................................... 13

REFERENCES ......................................................................................................................... 14
LIST OF FIGURES

Figure 1: The SAP HANA database architecture.................................................................6
Figure 2: Memory alignment and resulting access patterns for row and column stores........7
Figure 3: Accuracy of the runtime estimation......................................................................8
Figure 4: Example of SOA application using Sybase Workspace........................................9
Figure 5: Example of a service order application using MBO’s in SUP...............................11
Figure 6: SAP GUI - Plant Maintenance Service Order.......................................................12
ABSTRACT

The purpose of this project is to use System Analysis and Program Development’s (SAP) Sybase Unwired Platform (SUP) to create a custom web-centric application for the explicit purpose of plant maintenance. Using database information obtained from Stetson’s proprietary access to the University of California’s SAP server ‘Perth’, we will be creating a mobile platform that is going to be optimized with the intention of an Android tablet being the primary hardware platform it will be run on. Being a web-centric application, this program will be accessible from other web-enabled devices, but the main focus will be emphasized on creating a working GUI environment designed with Android usage in mind. We may decide later that it is worth our effort to include other popular web-enabled devices in our design of the GUI environment, but for the time being our main emphasis is going to be on the backend processes that will happen regardless of which device the end user utilizes.
1. INTRODUCTION

With the ever growing need and usage of mobile devices in today’s society, the mobile development market has seen a subsequent rise in recent years. Sybase, taking notice of this growing trend, began pursuing development towards implementing a set of tools back in 2005 that were intended for development usage in the mobile market. Some may have thought that this was a sign Sybase was moving away from its already well-established stance in database implementation and design, however as CEO of Sybase John Chen stated [FON12], this was actually a sign of the opposite happening. For Sybase, this was a golden opportunity to not only develop for and in the mobile market, but for them to bring their own level of database expertise to the mobile market in general.

With Sybase’s belief that creating tools for the explicit use of mobile application development is more important than simply taking advantage of the growing mobile market as-is, they developed their own set of tools known as the Sybase Unwired Platform. The Sybase Unwired Platform (SUP) is essentially a series of tools that were created specifically with mobile platform design in mind. The general consensus is that if you have the proper tools that are optimized for the explicit task of mobile platform development, then one can create an optimized application for said device. With tools such as these, creating an Enterprise Resource Planning (ERP) system with a mobile interface is made surprisingly easy when compared to using more basic tools, with the core [TAN07] of Sybase’s Unwired Platform’s tool being structured around the open-source Eclipse coding program in such a way that the environment is highly immersive.
to the point where the need of manual coding is drastically reduced while still maintaining a sense of efficiency.

System Analysis and Program Development (SAP), a software company well known for its foothold in the information systems of businesses world-wide bought out [WOR10] Sybase in late 2010 for $5.8 billion USD. The deal struck between these two conglomerates was mutually beneficial for both parties in the long run. As a result, each company had viable assets that each of them could mutually benefit from. Sybase was able to gain access to SAP’s own database technology [SYB10] as well as their already well-established large business information system market. SAP would also gain full access [WOR10] to Sybase’s SUP tools and their expertise in mobile platform development, providing them with a golden opportunity to gain a foothold in the mobile platform development market. With this buyout, SAP would have everything they needed internally to not only place a foothold in the mobile development market, but to add their own database and business structure into the market itself.

In late 2012 [BOR12], SAP executive Sanjay Poonen announced to the world that SAP is planning on getting heavily involved in the mobile application development market, with plans of bringing their software to at least a billion people by 2015. This is a bold statement, but SAP plans to increase development of their products with mobility in mind. Already possessing a mobile application store [PR12] for Blackberry, Android, and iOS devices, SAP has nearly all the resources they need ‘in-house’ [BOR12] to achieve their goal of putting themselves in the mobile market. Whether or not they will achieve the success they are striving for remains to be seen, although SAP has already seen success in applications such as Recalls Plus [LEV12], on Apple’s App Store, an application for the purpose of alerting parents on up-to-date information pertaining to recalls and safety information.
2. Related Work

2.1 SAP ERP

One of SAP’s most notable developments in the business world is their development of their own Enterprise Resource Planning (ERP) systems. ERP systems are by definition [VOG02] “any software system designed to support and automate the business processes of medium and large businesses.” In the case of SAP, this information system conglomerate makes explicit usage of their database technology to provide a sturdy backbone [BÖG11] for their ERP application systems through which business data can be efficiently and reliably transferred both internally and externally in a business.

One important thing to note about SAP’s ERP systems is that its ultimate performance is largely dependent [BÖG11] on the performance of two different pieces of technology, an application server and a database management system. SAP uses their own application server in their ERP systems, but they do offer the option for the use of several different database management systems. This provides a certain layer of flexibility and further demonstrates the amount of customization that SAP’s ERP offers to its target users.

While SAP’s ERPs are gaining, and have gained, ground in the economic market, it should be noted that just because such a system is implemented in a business’s infrastructure does not mean that said business will gain any improved benefits as a result. If fact, the opposite results can be expected when the system is either improperly implemented or the organization that will be hosting and/or supporting it is either improperly trained or ill-prepared [WIL11]. As with any other information system or technological change that occurs within a company or
organization, successful integration is key, not just with the hardware and software, but with the end users as well. With this in mind, SAP ERP users must make sure that the system is properly integrated in a manner that is both efficient and reasonable [VOG02]. SAP ERP’s can be tricky [WIL11] to use, as depending on the level of authority the intended users will have over the actual database said system runs on, mistakes both trivial and serious may be hard to correct, meaning more effort wasted and profit lost. Luckily, there are alternatives in preparing for SAP ERP integration. Several companies, such as WileyPLUS [WIL11] offer virtualized environments that can replicate the SAP ERP environment for training purposes. What makes this an invaluable tool is that unlike an actual SAP ERP system, resetting database information and application data is relatively easy. It is not an alternative for the actual ERP, but it is a valuable tool for training purposes. One such simulator provided by WileyPLUS is a Java-based [WIL11] platform that is accessed via a Java-enabled web browser. Because of this, it requires no special installation features, save for installing and enabling Java on the machine intended to run the simulator. This also has the benefit of being near-hardware independent, allowing for cross-platform usage.

2.2 SAP HANA DBMS

Of the technology that SAP has pioneered in, database management is one of their most well-known achievements. Most of today’s SAP’s ERP systems run with the intention of using “in-memory” [BOE12] data storage. What this means is that while the standard practice of storing and recalling database information, such as SQL tables for one example, from a physical hard disk drive to memory is still a viable option, SAP does things differently through means of storing the information directly in the active memory rather than to just a physical storage
device. By doing this, SAP takes advantage of the huge latency decrease that comes from not having to read and write to physical media, which is useful as it is generally assumed that most information stored on a database is intended to be kept available on demand in its most up-to-date status. One such database that makes use of this principle that SAP commercially uses is the SAP HANA database management system (DBMS).

![Figure 1: The SAP HANA database architecture](image)

SAP’s HANA DBMS is unique among the DBMS’s that exist on the market in part due to its significant flexibility that it offers. However, one of its notable features is how it makes efficient use of the hardware [FÄR12] that it runs on, it able to take advantage of the server’s drive type, number of cores, memory capacity, and memory locations. The DMBS is designed from the bottom up to be geared for heavy business usage and was thus designed with the ability for smooth integration with applications designed by SAP and is capable of using the same application’s servers to allow for shared-data communication [FÄR12] between the DBMS and the application in question. This DBMS is equipped with the following three different types of
query engines, as seen in Figure 1, to meet current and future needs in a business environment that is still in the process of evolving: an SQL-based engine, a text search engine, and a graph engine. The SQL-based [BOE12] engine allows for easy SQL access to structured data and allows for easy transfer of data in and out of the DMBS. The text search engine is standard and is meant to exist as a flexible, generic way to sift through structured, semi-structured, and unstructured data as pure text. The graph engine is a set of algorithms that allow for graphical data to be comprised from business information, an expected feature from software designed for business functions.

2.3 OLAP and OLTP

![Figure 2: Memory alignment and resulting access patterns for row and column stores [RÖS12]](image)

Online Analytical Processing (OLAP) and Online Transactional Processing (OLTP) are two important types of activities that are used in databases that have been structured around a column or row format [RÖS12]. OLAP is generally considered to be geared towards the storage of data in column format. The reasoning behind this is that column-based storage is typically the primary choice for storing data that is not only going to be recalled on the fly [SIK12], but is also not expected to be changed or updated often. As a result, OLAP is optimized for fast recollection of data and is not the most efficient choice for making inserts within databases. On the other hand, OLTP is geared towards row-centric storage and is made with the understanding that data
accessed through this means is going to be updated on a normal basis and thus, while it may not be as fast at recalling data on the fly, it is very accurate. Figure 2 shows how the same set of data can be stored in either row or column format for use with each process.

Figure 3: Accuracy of the runtime estimation [RÖS12]

OLAP and OLTP are both processes that have had an impact on how future databases are formed. SAP’s HANA DBMS takes advantage of both of these processes with its hybrid-database [RÖS12] structure. HANA is able to partition its own internal memory and store sections of its own data in either a row-based or column-based structure. By doing so, each section of data can be optimized based on its intended usage. Figure 3 illustrates the end results of a test run to determine the effectiveness of an ‘advisor’ application that can be used in conjunction with DBMS’ such as SAP’s HANA DBMS. The figures shown in the data illustrate how effective row-centric or column-centric storage can be for certain categories of data, and that having the correct storage type can greatly affect the time expended overall. For data that isn’t expected to change often, like a list of street addresses, it can be stored in column format,
while data such as a list of momentary exchange rates can be stored in a row format, allowing for levels of optimization that are not possible in a database that can only handle one structure.

2.4 SOA

Service Oriented Architectures (SOA) are by definition [TAN07] systems of frameworks, patterns, and principles of development and design for the purpose of creating applications that are based on different services and designs in a manner that allows for interconnectivity between similar applications. The principle behind this idea is that the functions of the application are exposed to other applications and the users as services rather than just abstract functions. This is very useful for web application development as the services created by the application can be designed to be accessed by multiple clients and because they are designed to be loosely connected to other software and services, the design of the services and sometimes the services themselves can be reused as templates in other projects and applications. Figure 4 illustrates a generic result of what the final result of such loose interconnectivity of differing applications and functions can be if implemented properly.

![Diagram of SOA application using Sybase Workspace](image)

**Figure 4: Example of SOA application using Sybase Workspace [TAN07]**
Application Programming Interfaces (API’s) are a prime example of the capabilities that SOA has to offer in terms of loose compatibility. API’s are the interface [BEA08] that is used to bridge the gap between the programmer and the language library used, and it makes sense that having the ability to recycle and reuse structures and designs in the productions of an API would be of benefit in future projects. Enterprise Service-Oriented Architecture (E-SOA) makes use of this in its API design. E-SOA is quite an adaptable platform that is used and maintained by SAP and, as a result, is quite adept at interfacing and coupling with SAP’s own backend databases. As of 2008, there are at least 1,700 different web services provided by SAP’s E-SOA alone that are at disposal to developers for the creation of their own applications. This, combined with the fact that SUP provide integrated tools [TAN07] that support the design of SOA applications, makes SOA a powerful structural design pattern to use.

3. Implementation

We are going to use Sybase Unwired Platform’s utilization of Mobile Business Objects (MBO) to create a mobile application for the explicit purpose of allowing plant workers using SAP’s database services to place orders for plant maintenance purposes on the fly via remote devices such as tablets and cellular phones. The boxes seen in Figure 5 are GUI representations of what an MBO is and the relationships they can have with each other.
We shall begin by creating a shell application within SUP and populating it with MBO’s, each of which represent a data container. Each of these will contain object data that is called from the SAP Perth server we will be using. This data is going to be bound to the objects that we create in the GUI environment in SUP. We will match this data to the dataflow required and present in the ‘order’ form used by SAP’s GUI program, specifically found in the submenu ‘Maintenance Processing’. An example of the GUI can be seen in Figure 6, which is what a typical worker/student would see if they were placing or viewing an order. Once the data is properly bound to the correct MBO’s, which will look similar to Figure 5, we will either create a native or hybrid-web application from the structure we will have created.
At this point it would come down to fine tuning on the part of the native application. From what we’ve gathered so far, there is a high probability that we will have to alter the syntax of some of the created methods and attributes in the finished Java code to be compatible with the device in question. In the event that we are going to design this to run natively on an iOS device, we will need to have access to an Apple computer running Xcode. We will need to connect to the Apple as a server with my machine and share the final code from our designated Windows computer with Xcode so that we may create a native application that we may test and run on an iOS device such as an iPhone or iPad. No such measures will be required for Android and/or Blackberry application development as Sybase has all the tools necessary save for certain SDK’s, which can be installed into SUP easily.

The hybrid-web application will require minimal design changes and is almost
guaranteed to run on any web-enabled device so long as it is Java-enabled. The structure should look similar in design to that used in Figure 6 by default.

4. Results

Currently, we have access to the software that is required for the creation of an application built for the purpose of plant maintenance; however we have yet to acquire the proper backend privileges necessary to proceed any further.

5. Conclusion

As it currently stands, we have the same privileges as a typical student and thus do not have proprietary access to the back-end business objects that we will need access to. For the foreseeable future, we can go ahead continue our structuring of the framework of our application, and continue testing once we have the proper privileges.
REFERENCES

[BEA08] Beaton, Jack K.; Myers, Brad A.; Stylos, Jeffrey; Jeong, Sae Young (Sophie); and Xie, Yingyu (Clare). “Usability evaluation for enterprise SOA APIs.” In Proceedings of the 2nd international workshop on Systems development in SOA environments (SDSOA ’08). ACM, New York, NY, USA, 29-34. DOI=10.1145/1370916.1370924 http://doi.acm.org/10.1145/1370916.1370924


[BOR12] Bort J., “This Exec Has An Outrageous Plan To Make SAP Bigger In Mobile Than Apple”, Business Insider, Nov. 4 2012.


