Carnegie Mellon University
Master of Science in Information Technology
Software Engineering (MSIT-SE)
Practicum in Software Engineering (17-677) Approval Form

Student Name: Jane Doe Date: 9/19/2002

Project Title: Re-Engineer the B2B Interface for a major Customer

Project Proposal: Please attach a proposal that contains all of the components listed below.

1. Executive Summary of effort
2. A brief description of the industry, company, and specific facility at which the project will be conducted
3. Description of the problem
4. Project purpose and goals
5. Project approach and methodologies
6. Project deliverables
7. Timeline for deliverables
8. Identify Technical Advisor by name, contact information, position and responsibility
9. Identify Supervisor by name, contact information, position and responsibility

Approvals

Mr. Smith	Application Development		
Supervisor Name	Department	Signature	Date
Mr. Jones	Application Development		
Technical Advisor Name	Department	Signature	Date

Carnegie Mellon Approvals

Mel Rosso-Llopart		
Faculty Advisor (Mentor)	Signature	Date
Mel Rosso-Llopart		
MSIT-SE Director	Signature	Date

1. Executive Summary of Effort

SAP (which stands for Systems Applications and Products in Data Processing) is the ERP (Enterprise Resource Planning) package that is used by all of the Wireline facilities in North America. The SAP Business Connector is the integration tool component of the SAP system that allows the ERP application to pass transactions between different systems, translating the transaction from the sending system for the receiving system

The SAP Business Connector B2B (Business to Business) application was poorly designed. The current application is maintained in Production, because server names and URLs are written into the code. The server names and URLs are different for the Development and Production environments. The code cannot be migrated from development to Production, because of the hard coding and the risk of customer orders being loaded in Development instead of production.

Support and maintenance are difficult. There is an incomplete audit trail and developers are not notified of failures. Tracking down problems is very time consuming.

There is no developer in the Wireline Business Unit that has experience with this application.

The first phase of this project will be upgrading the Development SAP Business Connector to the latest version. During this phase the developer will become familiar with the SAP Business Connector Development Tool and will study the current application.

The second phase is writing coding Standards for developing in the SAP Business Connector and writing the requirements for new application.

The third phase is design and coding of the new application. Testing will be done in Development and Certification. The certification SAP Business Connector Development server will be upgraded, before the new application is migrated to that server.

The fourth phase is upgrading Production and moving in the new application. Thorough monitoring will be done for two weeks to ensure that everything is functioning properly.

The last phase is the wrap-up. Documentation will be written to give guidelines and instructions to future developers that will be maintaining this interface. A final project will report be written for CMU. A presentation will be given to The Company and CMU.

2. Background

A brief description of the industry, company, and specific facility at which the project will be conducted.

Industry and Company

The Company is a leading global communications and IT company with employees around the world and sales in over one hundred countries. It supplies advanced communications solutions and products and services for the Internet.

The Company provides products and services from the telecommunications industry. The company provides equipment for wireless, wire line, and broadband communications. The Company provides high-performance communication solutions for voice, video and data communications.

The Company Wireline Business Unit will conduct this project. This business unit focuses on products for wire line communications.

Facility

This project will be managed from The Company facility in northern, Ohio. At this facility they manufacture power supplies. These systems are configurable and are often unique for each customer. This facility is one of the plants that comprise The Company Wireline Business Unit.

SAP is the ERP package that is used by all of the Wireline facilities in North America. They currently have about fifteen plants using this ERP package. SAP is a German-based software firm, which has designed and developed SAP R/3. This system is fully integrated. Financials (Accounts Payable, Accounts Receivable, General Ledger), sales, production planning, and materials management are all connected. When a transaction is completed in one area, the other functional areas are immediately updated and have access to this information.

In addition to manual data entry, orders can be placed against The Company through EDI (electronic data interchange) and through B2B (Business-to-Business) processes. Documents can be automatically faxed to vendors and customers from the SAP system. Order acknowledgements, shipping notices, and invoices can be sent out of the system through EDI.

The SAP Applications Development team will manage this project and perform most of the tasks. The majority of the team is located at the facility in northern Ohio. The SAP Basis team, is located at The Company offices in southern Ohio, will have a small role in this project.

3. Problem Description

High Level Problem Description

A major customer sends purchase orders to The Company using a Business-to-Business (B2B) process. The information passes through the SAP Business Connector. The SAP Business Connector application was written without documentation, and without standards. The code for the development environment is different than the code in the production environment because of the URLs and machine names are specified in the code. This application is very difficult to debug, to maintain, and is subject to frequent failures.

This project will rewrite this SAP Business Connector portion of the B2B interface. The SAP Business Connect is a version of the WebMethods EAI (Enterprise Application Integration) product. The code needs to follow our standards, it must be documented, hard coding will be removed, and additional error handling and debugging capabilities will be added.

Detail Problem Description

Currently one of The Company's customers can place orders to The Company through a Business-to-Business (B2B) process. The customer posts a XML document to the SAP Business Connector, and this triggers a routing rule and services in the SAP Business Connector. The SAP Business Connector then reformats this data into an IDOC. IDOC is a SAP term for the layout of data, with very specific segments and qualifiers, used to pass information between different systems. Through remote function calls the IDOCs are passed to the SAP server and processed by the SAP system to create sales orders in our system.

Once the sales order is loaded into the SAP system, the appropriate Customer Service Representative (CSR) is notified. The CSR will process the order. Within this process they will send two acknowledgements (an assignment and a confirmation) back to the customer. IDOCs will be sent to the SAP Business Connector and the SAP Business Connector then formats the data into XML, which is then posted to the customer's website.

There are numerous problems/issues with this interface. The interface was poorly designed, tested, and documented. The application did not follow any standards when it was written. There was no naming convention and no specified manner for documenting the code. There is little documentation on this application. No policies or instructions existed for migrating the code from Development to Production.

The Company agreed to participate as quickly as possible on a pilot B2B project for this customer. The main goal was to get it done as soon as possible. Unfortunately the consequence of this was that the interface was poorly designed, tested, and documented. No time or effort was spent creating an infrastructure that would help ensure usability, extensibility, and maintainability over the next several years. A major customer requested that transactions be sent between the customer and The Company, but there was no emphasis on the ease of resolving problems when transactions are not completed. We often cannot tell whether the transactions were dropped inside the SAP Business Connector or if it failed to reach the customer. There is insufficient error recovery and data integrity of the entire transaction. The data is used to create a legal and binding sales order against The Company and we cannot guarantee that all requests for sales orders become sales orders or that all acknowledgements are sent back to the customer without a manual reconciliation.

Error tracking mechanisms and audit trails were not put in place. At times transactions come into the SAP Business Connector from the customer or from SAP and the transaction does not complete. It is very difficult to read through the system logs to try and find where a problem has occurred. There are email notifications of successful transactions, but no notifications of failures. The developers receive two emails when a transaction is successful. If the developers receive one email, then he or she knows there is a problem. Unfortunately when the developers receive no emails, he or she cannot tell if this is due to the customer not sending an order or if there were severe problems. Problem resolution is very difficult and time consuming.

There is no easy or convenient way of checking the return code when data is sent to the customer. The developers have to search through raw XML files to look for the return code. Each transaction is a separate file, so this is an extremely tedious process.

Often problems occur, but it may be several days before the customer notifies us of an issue. There is currently no easy way of verifying that everything is running smoothly.

Hard coding was used for the routing rules and for URLs. Theses rules are used to point to the appropriate server (test server vs. production server.) Since the rules were hard coded (rules specific to an environment are embedded into the code) the versions of the code between development and production are different. No procedures (i.e. instructions) were put in place to explain what needed to be changed before moving a package into Production. Currently the team relies on the developer to test the code in development, then identifying all of the hard coding and make the appropriate changes to reflect the Production environment, and then moving the code (package) to the Production SAP Business Connector. Due to the hard coding, maintenance is often done directly in Production. The developers have decided this is safer than migrating the package from Development to Production. When making changes directly in production, there is a risk of causing the application not to run or that the application will run but will produce incorrect information.

The application was not modularized; the code was written in one giant package. This means all of the code is tightly connected (coupled). Only one developer can safely work on the system at a time. If a developer makes a small change to one area the entire package must be moved to Production to put the small change into Production.

Migration from Development to Production is currently a difficult, laborious, and risky task. Due to poor design there is configuration that should be specific to an environment that is instead tied to the code modules that are migrated from Development to Production. There have been incidences where production data was written to our Development server and test data was written to the Production server. Migrating software to Production, or refreshing the Development environment with a copy of Production, runs a severe risk of crossing environments and data going to the wrong server (environment).

4. Project Purpose and Goals

Purpose

1) Rewrite the system to provide The Company with better error handling and monitoring. This will reduce the time and effort needed to resolve problems, and allow The Company to be notified when an error is detected. Currently one analyst spends 25% of her time researching application failures. This typically requires additional involvement from 2 - 4 other resources to resolve. This will be reduced to one resource to identify and resolve issues, as well as limiting the current research time to identify problems to 2.5% (10 hours to 1 hour per week) of her time.
2) Rewrite the system to follow the standards will increase maintainability. When new programmers are assigned to work on the application, they will find the application easier to understand if standards are followed. Currently it takes a new developer about one month to understand the application. Having standards will reduce this to one week.
3) Be able to migrate code from Development to Production without any hard coding. The same code can be run on the Production server and the Development server, which will reduce the risk of a change causing an outage in Production.
4) Modularize the code. Development time will be reduced because it will be easier to identify the components that need to be changed and there will be less of chance of the changes breaking other components in the application. Testing will be simplified, because the code will be modularized so less involved tests can be conducted to test the changes.
5) Eliminate maintaining software directly in Production. Reducing the changes in Production to zero will reduce the risk of having a system failure.
6) Develop an essential skill set that is critical to our strategic business unit. This will add one new resource to the pool of developers who can support this existing application, as well as perform new development.
7) Rewrite this system to produce a more extensible and reusable application. The developers will be able to more easily add additional processes and extend this to additional customers. Due to hard coding it is nearly impossible to reuse the existing application. With the new system a new customer, who wants similar transactions, could be added and fully tested in four months.
8) Have a more stable application. Currently almost every time a change is made, something else breaks. There is duplicate code in multiple places and it is very difficult for the developer to find all of the places that need to be changed. The side effects or defects found in Production after changes are implement will be close to zero after the rewrite. The rewrite will reduce the number of error occurrences by 25%. This will increase application availability to the customer to 99.9% (in line with Operation service level agreements).

Goals

1) Standards for developing in the SAP Business Connector will be written.
2) Add additional error tracking and monitoring and audit trails. All transactions incoming and outgoing will be logged to reduce incident investigation time.
3) Create an application that is easier to maintain. Small changes will take three to four hours instead of one week.

4) The system should require less maintenance and manual monitoring. There should no longer be weekly phone calls to the customer to see if everything is running smoothly. The audit trail will show when a transaction failed and when an error return code was received from SAP or the customer. Emails will be sent to notify developers of failures.
5) Create an application that can be easily extended to support new B2B customers. New customers requesting similar transactions can be added and fully tested in four months. We will be able to extend the revised application to all of our customers.
6) Have a more stable application. The current application fails fairly often, so that we are not meeting our service level agreements. The rewrite will reduce application failures and allow us to meet or exceed our service level agreements.
7) Take advantage improvements after the SAP Business Connector upgrade (planned completion within the next two months.) There are additional built in functions that we can take advantage of in the newer release.
8) Have a skilled resource within the Wireline Business Unit, able to support the B2B interface.

5. Approach and Methodologies

Approach

The project will follow the spiral method. The spiral cycles are: 1) requirements gathering 2) upgrading the development server, 3) write SAP Business Connector development standards 4) develop outbound orders, 5) develop inbound orders, 6) certification testing, and 7) implementation in Production.

Each phase, will have the following phases: 1) customer communication, 2) planning, 3) analysis 4) risk analysis 5) design, 6) code and unit testing, and 7) customer evaluation.

Customer evaluation is always the last phase in a loop of the spiral. The customer must approve one loop, before work on the next loop can begin. The customer can be different for each loop of the spiral. Part of the planning phase is to identify the customer to do the evaluation.

The SAP Business Connector will be upgraded in the development environment. Code will be written and tested in the development environment. When all of the code works, the Certification SAP Business Connector will be upgraded and new application will be migrated. In Certification The Company employees will test all of their scenarios. When those scenarios are successful then a full test will be done with the participation of the external customer. When the test with the external customer is completed successfully the new application can be moved to Production. The SAP Business Connector will be upgrade immediately prior to the new application being moved to Production.

Methodologies

During the requirements gathering, the developer/analyst will review all of the problems with the existing system with the other developers so he or she can get a firm understanding of the issues for resolution. The functionality that exists today will be needed in the new application. To identify the requirements and functionality, information must be gleaned from e-mail messages sent to several resources. The requirements will be put into a single document. The design and coding phases will be skipped on this loop of the spiral.

Upgrading the Development SAP Business Connector server is the second spiral. Regression testing of the Order Status application, which runs on the same Business Connector, will be done during this phase.

During the writing the standards phase the developer will study the existing system and the Order Status SAP Business Connector application. The standards will contain guidelines or rules for: naming conventions, how to specify a server name or URL addresses, commenting the code, and guidelines for modularizing the code. The standards will be revised, if necessary while the coding phases are done. The design and coding phases will be skipped on this loop of the spiral.

The inbound orders and the outbound orders will be the two loops of the spiral where design, code and testing will be done.

The designer will determine which design technique will be the most appropriate to use. The design will probably be object orientated and component based. The SAP Business Connector provides numerous built in functions or components. These functions will be used whenever possible instead of writing custom code. The application will be designed to have reusable components, so that this application can easily be extended for additional customers. Use cases, sequence diagrams, and class diagrams will be created during the design phase. The code will be modularized and each workflow will be tested individually. The SAP Business Connector breaks code into packages, services, and workflows. Workflow is the smallest element and package is the highest-level object in the hierarchy. Each workflow will be tested individually and independent of the other workflows. Once all of the workflows are complete for a package, then the package will be tested. Testing will involve production like data so that verification can be made that data will be sent properly between the machines. Testing will also involve error-producing data; to verify error handling and notification are function properly.

The implementation in certification testing spiral will be used to fully test the application. The Company IT business analysts will perform the initial testing. Then additional testing will be done with the external customer participation, verifying that information can be loaded into their system. If any problems arise, then the project will return to a previous spiral so that corrections can be made. The Order Status application will also be tested.

The last phase of the spiral is implementing the newly revised application in the Production environment. Once the new application is in Production, there will be close monitoring to en-sure that everything is running smoothly for two weeks. In problems arise they will quickly be addressed. If there are no critical defects found then this project will end. Critical defects are defined as the application not meeting the stated requirements or the application is missing business critical functionality that was overlooked when the requirements were written.

6. Project Deliverables

Deliverables for CMU (and how this reflects knowledge of core courses)

Requirements

- The requirements must be specified before the design phase can begin. The users/customer and the developers must have a common understanding of the goals of the project. The analyst/developer must understand the requirements, before designing or coding this application.
- Gathering requirements was covered in Managing Software Development and Methods of Software Development.

Software Project Management Plan (SPMP)

- The plan can constrain the scope and list the assumptions. Having a project plan is essential when there are multiple people working on the project. The plan specifies everyone's roles and responsibilities and when milestones and deliverables are to be completed. Having a plan gives something to measure against.
- The SPMP also contains risk analysis. This is important to notify management of potential risks and for the most likely ones, a mitigation/contingency plan can be put in place. A software project has a better chance of succeeding when the team is aware of potential risks, before the risk becomes a reality.
- Creating an SPMP will help show the skills learned during the course Managing Software Development.

Design

- Having a design will lead to a better-structured application. If the programmer(s) just begin coding, then the code will be more ad hoc. Having a design is essential when the work is being divided up among multiple programmers.
- Having a Class Diagram will show the developers how objects are related and their parameters for every method. This information is critically important when developers are working on two separate objects and need to transfer information back and forth.
- The architecture of the system needs to be understood and documented, before development begins. This also helps developers understand the system, when enhancements/changes are done years later.
- A sequence diagram demonstrates the flow of data through the system.
- Having a design will help the current developer(s) understand how modules fit together and depend on one another.
- The architectural design will show the knowledge gained in Architecture of Software Systems.
- The UML class diagram will show the object-oriented skills learned in Architecture of Software Systems.

Use Cases

- This can help the developers understand the functionality of the system. The Use Cases are written in manner that is also understandable to the users (non-developers.) Test cases can be built from the use case scenarios.
- This shows the knowledge learned in Methods of Software Development. The developer needs to create use cases to help him or her test the application.

Standards

- Standards need to be written. This will increase maintainability during future development and enhancements. Having standards in place will help multiple developers understand the code and objects created by the other developers.
- This shows what was learned in Analysis of Software Artifacts. The developer will have to look at an existing system and understand it, with very little documentation.

Time Tracking Spreadsheet

- This is the first time IT team has done a project like this. The project lead will track the estimates and the actual time, so he or she can see if the estimates were reasonable.
- Tracking percentage complete allows the team to monitor their progress, and to communicate the status of the project. (i.e. on track, off, ahead of schedule) This is particularly useful when there are dependencies that need to be scheduled such as testing with the customer.
- It is also important to track the time, so that management knows how much time was spent on this project.
- Tracking the estimates and time will help to improve the team's estimates in future projects.

Additional Deliverables for The Company

New B2B interface for a major customer

- This application runs on a special server and with special software. The software is purchased through SAP. The developer will not be able to create an executable that can be distributed to CMU.

Technical Documentation

- This documentation will contain information about where logs are stored, what kind of information each log contains, where files are stored, and instructions for migrating code from Development to Certification to Production, and solutions to frequent problems. The audience for this document will be new SAP Business Connector developers and anyone involved in issue resolutions.

High Level Timeline*

Aug – Sept 2002	Write the proposal. Get approval.
Sept – Oct 2002	Basis will upgrade the SAP Business Connector.
Sept – Oct 2002	Become familiar with the SAP Business Connector.
Oct 2002	Write the Standards, Project Plan and Test Plan
Oct 2002	Write the Requirements
Nov 2002	Write the Design document and Use Cases for outbound orders
Dec 2002	Code and Test outbound orders (from the external customer)
Jan 2003	Write the Design document and Use Cases for inbound orders
Feb 2003	Code and Test inbound orders (from SAP)
March 2003	Test in the Certification Environment
late March 2003	Go Live
April 2003	Technical Documentation
April 2003	Write Final Report and Presentation
end of April 2003	Give presentation to The Company and to CMU

Goal

Finish by the end of April.

* All dates are subject to change due to business requirements.

7. TimeLine for Deliverables*

Proposal (approval of final proposal by CMU and The Company)	Fri. September 6, 2002
Load the SAP Business Connector development tools Obtain a login to the SAP Business Connector server	Fri. September 6, 2002
Upgrade the SAP Business Connector – Dev.	Fri. September 27, 2002**
Project Plan	Mon. October 7, 2002
Requirements	Mon. October 21, 2002
Standards document (initial)	Mon. October 28, 2002
Test Plan	Mon. November 4, 2002
Design outbound orders	Fri. November 22, 2002
Use Cases inbound orders	Fri. November 22, 2002
Code outbound orders	Fri. December 20, 2002
Design inbound orders	Fri. January 31, 2003
Use Cases inbound orders	Fri. January 31, 2003
Code outbound orders	Fri. February 28, 2003
End to End testing (no bugs)	Fri. March 14, 2003
Upgrade the SAP Business Connector – Prod.	Wed. March 19, 2003**
Go Live	Wed. March 19, 2003**
Standard document (final)	Mon. March 24, 2003
Technical Documentation	Mon. March 24, 2003
Final Report	Mon. April 14, 2003
Time Tracking Spreadsheet	Mon. April 14, 2003
Final Presentation (written)	Mon. April 21, 2003
Presentation findings to The Company and CMU	Wed. April 23, 2003

* All dates are subject to change due to business requirements.

** These dates are subject to agreement between the SAP development team, the Basis Team, and the business representatives.

8. Technical Advisor

Mr. Jones
jones@The_Company.com
555 555 - 5510

Senior SAP programmer/analyst

Responsible for the analysis, design, development and testing of enhancements to The Company's SAP system. Provides technical and analytical leadership for projects. Defines development standards. Mentors the other team members.

9. Supervisor

Mr. Smith
smith@The_Company.com
555 555 - 5512

Application Development Manager

Manages development activities related to SAP R/3, EDI, eCommerce and data warehousing for The Company Wireline Business Unit. Sets priorities based on regular meetings with representatives from various areas of the business. Coordinates development activities with other disciplines within the global IT organization