

UBC Database Interface Project

Edmond Ao

Teddy Ao

Angeline Tin Htun

Linda Wong

University of British Columbia

COMM 436

June 2001

Disclaimer: "UBC SEEDS provides students with the opportunity to share the findings of their studies, as well as their opinions, conclusions and recommendations with the UBC community. The reader should bear in mind that this is a student project/report and is not an official document of UBC. Furthermore readers should bear in mind that these reports may not reflect the current status of activities at UBC. We urge you to contact the research persons mentioned in a report or the SEEDS Coordinator about the current status of the subject matter of a project/report".

TABLE OF CONTENTS

I. Overview	1
A. Background	
B. Objective	
C. Scope	
II. Departments under study	1 – 4
A. Campus Planning & Development	
i. Organization	
ii. IT applications	
B. Classroom Services	
i. Organization	
ii. IT applications	
III. Relevant business processes	4
IV. Problems	4 – 5
A. Locating information	
B. Decentralized systems	
V. Constraints	5 – 6
A. Financial	
B. Human resources	
C. Technical	
D. Political	
VI. Solutions	6 – 11
A. Crystal Reports solution	
B. API solution	
C. Oracle 9i solution	
VII. Feasibility	11
VIII. Recommendations	11 – 12
IX. Future Issues	12
X. Implementation Timeline	12
XI. Bibliography	13
<i>Appendices A – J</i>	

APPENDIX LIST

Appendix A – Sample CPND Space Inventory Report

Appendix B – CPND Database Relationship Diagram

Appendix C – Classroom Services Utilization Report

Appendix D – Oracle Database Table Diagram

Appendix E – Object Oriented Enterprise Modeling Diagram

Appendix F – Process Diagram (Before)

Appendix G – Process Diagram (After)

Appendix H – Soap Overview

Appendix I – Cost / Benefit Analysis

Appendix J – Intangible Benefits / Technical Feasibility

I. OVERVIEW

A. Background

The University of British Columbia resides as the largest educational institution in British Columbia. Being such a large organization, UBC is divided into many divisions and faculties in order to function more efficiently. While most divisions are self-sustaining, there are many instances where the need to interact between divisions becomes inevitable. One department often requires the data found within another's database. It is during such instances where the incompatibility between systems creates both time and data inefficiencies. Since UBC is such a large institution, we will focus our study on two divisions, Classroom Services and Campus Planning & Development.

B. Objective

This project may be the first step in achieving the ultimate goal of linking all databases within all departments in UBC. We will aim to determine whether an interface system is available to link various databases between the two departments of our study. The primary criteria for our solutions will be to allow real-time access of other departments' data. An ideal solution will allow complete data integration. Our solutions will hopefully improve the efficiency of many business processes of the respective departments resulting in the following effects:

- Resource efficiency of humans and systems
- Reduce staff frustration and interruptions
- Ensure data consistency

C. Scope

This study will include a study of all business processes as well as current technical systems within the organizations. We will then propose feasible solutions, however due to the limited timeframe placed on our project, our solutions will not include possible implementation guidelines.

II. DEPARTMENTS UNDER STUDY

A. Campus Planning & Development (CPND)

i. Organization

CPND is a department within the university's campus sustainability unit, Land and Building Services. CPND maintains space and facility information for the whole campus. Not only does

CPND supply information to its supervisory department, but it also replies to many information and report requests from a number of on and off-campus departments. These departments include Classroom Services, IT Services, Networking Engineering Service and Key Office. In short, CPND acts as an information backbone for many departments that depend on up-to-date facility related information.

CPND's primary responsibilities are maintaining and monitoring the entire inventory of space on the UBC campus, performing statistical analysis related to space inventory, providing data and reports to all department within UBC as well as outside agencies and the government, serving as a resource for space information and as a support role in space planning. (See Appendix E for roles and responsibilities) In addition to these responsibilities, CPND needs to generate reports about the space utilization at UBC from its database.

One of the major problems that plagues CPND is that people from different departments have different interpretations about the structure of each building. For example, CPND views Henry Angus building as two separate unites, one unit for offices and the other unit for classrooms, while other people view this building as only one unit. The difference in interpretation has lead to frustration and confusion when other departments are trying to understand information sent out by CPND.

ii. IT applications

The UBC Space Inventory Database (SID) was initially designed using D-Base in the 1970's. The senior analyst, Mr. Peter Jia, later transformed the entire database into a relational database under MS Access, which now resides on the Campus Planning & Development server. (See Appendix B for database tables) The SID includes the physical attributes (names, dimensions, structures) of about 400 buildings and 27000 rooms in both text and diagram formats. It also records the funding allocated to each room and building (found in clsrn\$ table). Our study only focuses on the text-based information found in the database (tables: Bldg-Index, Bldg-Name, Room, Space). The most frequently used data in the SID are the fields: building number, official name (current name), secondary name (any former names), address, position on map, age, total area, category, funding (either by the university or privately) and structure. The most important field is BLDGNO (building number), the primary key of the main building index as well as various other tables. The database can clearly identify at anytime the current dimensions of each room.

The SID is rather small and primitive. Whenever data must be updated, a CPND clerk must manually input the new information into the database. The database is also only accessible by CPND staff as it only resides on the CPND server and does not allow for external connection. Like previously mentioned, CPND's staff will frequently run queries to generate reports for other departments and outside agencies. An example of a report generated by CPND is shown in Appendix A, which lists all the rooms in each building at UBC as well as their dimensions. One of the most important use for such reports is to show government the usage of available space, whether the space within buildings is properly allocated for learning purposes. Other departments may receive similar reports that identify the classroom space in their buildings.

B. Classroom Services (CS)

i. Organization

CS' underlying function is to ensure efficient utilization of campus teaching space in a uniform and non-discriminatory manner. In its day-to-day operations, Classroom Services is responsible for classroom maintenance and renovations, classrooms and teaching laboratories management including the coordination of associated rooms and AV equipment, course scheduling, exam scheduling, classroom scheduling as well as ad hoc room bookings (bookings not related to a course or exam, usually one time usage). (See Appendix E for roles and responsibilities) CS books rooms for all classrooms and teaching space at UBC other than athletic fields and restricted department space (space booked directly by respective departments). In addition, CS needs to act as a consultant regarding utilization patterns of space for UBC and has to periodically analyze and generate reports on classroom utilization and usage. Appendix C shows a classroom utilization report which could be submitted to the government to show the effective utilization of available classrooms.

ii. IT applications

CS retrieves the majority of their information from the centralized Student Information System (SIS) that runs on an Oracle server at UBC. The SIS maintains information to support all functions of the Registrar. This includes student, course and enrollment data. CS uses course and enrollment data to schedule appropriate classrooms for each course. The classroom schedule is also stored in the SIS.

Since the information kept in the SIS is so abundant, it is not feasible to view and analyze all the information in its raw form. Instead, CS staff will often export a portion of the data into

other programs such as MS Access or MS Excel in order to manipulate the data easier and produce comprehensible results. They will also use Crystal Reports to retrieve information from the SIS in the form of a printable report by setting specific queries.

III. RELEVANT BUSINESS PROCESSES

Due to the difference in database platforms between CS and CPND, many inconveniences and obstacles exist for the two departments when gathering data from each other. Each department spends much of its time locating data and contacting other departments for required information. The room scheduling process for CS can best illustrate the problems that currently exist during the interactions between the two departments. (See Appendix F for depiction of the processes)

Before each semester, CS needs to reserve rooms for scheduled classes and associated activities. This event triggers two processes. First, the CS clerk needs to login to the SIS in order to run queries and retrieve course lists for the upcoming semester, as well as updated space data found in the SIS. Next, the CS clerk locates the contact information for the departments with data not available in the SIS, and contacts these departments for information such as room dimensions and sizes. These departments are most likely to be CPND and departments with restricted spaces. CS often needs information about space availability, room dimension, and updated building information from CPND. Once CPND receives a request for space information, a CPND clerk will run a query to extract the requested data from its MS Access database and send the results back to CS. For departments holding restricted space, department clerks access their own database, generate a list of rooms and times available to other departments, and forward the list back to CS. After receiving other departments, CPND and the SIS, the CS clerk will combine all the data. The clerk will verify whether the information returned fulfills their needs. If the information matches, then the CS clerk can assign rooms to courses and complete the room-scheduling request. If however, the information does not match, then the CS clerk will need to restart the process, beginning with 'locating contact information' and the process is repeated until the received information matches what the clerk originally desired. Only then, is the room assigned and the request is completed.

IV. PROBLEMS

Although the SIS Oracle server is an enormous database that holds registration and scheduling information, oftentimes, instead of expanding the SIS system, CPND built its own database to

hold additional information for its own convenience and also to keep information within its department. Problems arise when CS needs information from CPND's database and vice versa.

A. Locating information

CS staff members are unsure of where to locate information not present in the SIS system. At present there is no documentation indicating where different information is located. As a result, CS staff must often consult with many other departments before locating the correct information.

B. Decentralized systems

CS staff can only generate reports from the SIS using Crystal Reports, however, they cannot reproduce such reports from other databases. CS cannot directly access other databases such as CPND's Space Inventory Database because the databases are not connected. Consequently, CS has to waste time contacting, gathering, combining and validating data from various departments.

V. CONSTRAINTS

In our search for possible solutions, we were faced with various constraints, shaping the various feasible solutions.

A. Financial

In discussing with Mr. Marples, Director of Classroom Services, we decided that we would not have a set budget for our project. Instead, we would propose all viable solutions regardless of price, thus presenting a spectrum of solutions. Still, since UBC is a public institution with limited funding, it would be unrealistic not to consider the financial impact of our solutions.

B. Human resources

Both departments under study are rather small and have limited expendable staff members. Both departments also have limited technical personnel so whichever solution implemented will require hiring additional temporary staff. There should also be little or no technical maintenance required after implementation.

C. Technical

We must keep in mind our goal of connecting the majority of UBC systems in the future, thus our solution should also be scalable in the future to additional systems.

D. Political

Since our interfacing of different departments will place large amounts of data into the open, we must ensure that only data that should be shared is shared. Users should only gain limited access to information in respect to their needs. Such sharing of information should be create political conflicts between departments.

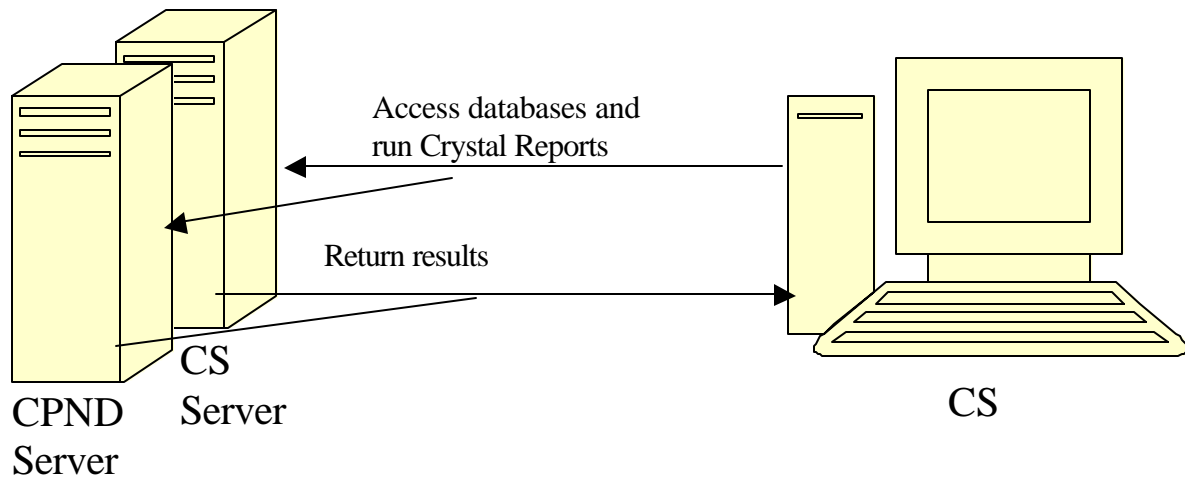
VI. SOLUTIONS

The ultimate goal is to have an interface that can automate communication between different department's systems and create a common platform that will enable all departments to perform their tasks while maintaining data consistency. Our team has devised three possible solutions, ranging from low to high-end in terms of both cost and time requirements. Since our objective is to suggest possible improvements to the current systems, the status quo solution (maintaining everything as is), although feasible, does not meet the project's goals and requirements. This alternative is thus not considered. Our project will instead focus on the following three alternatives:

A. Crystal Report 8.5 solution

The Crystal Report 8.5 solution is a low-end solution, which enables CS to run Crystal Reports queries directly on CPND's central database. CS's staff are given limited access through the network to CPND's central database (e.g. the directory where the Access database is stored will can be made available on CS computers as a local directory). Since MS Access is a file-based database, CS will be granted read-only access to the database file. Accessible data may include the data in the following tables from CPND's database: BldgIndex, Room, BldgName and Space. (Refer to Appendix B) The staff can now run Crystal Reports on the MS Access database and freely extract the data that is accessible to them. The latest version of Crystal Reports, version 8.5, allows users to simultaneously extract data from a variety of data sources, including relational databases. In this case, CS can produce a single report that includes the information from both CPND's database as well as their own. Crystal Reports will extract and display data from the data sources in the form of reports, but it will not store the information locally. Therefore, further manipulation of the extracted data is not possible.

The following diagram depicts the process:



The following are the advantages and disadvantages of the Crystal Report Solution:

Advantages:

- Classroom Services can retrieve information in real time whenever they require so
- Minimal software is required; therefore, lowering the cost
- Reduced need for excessive human communication between departments

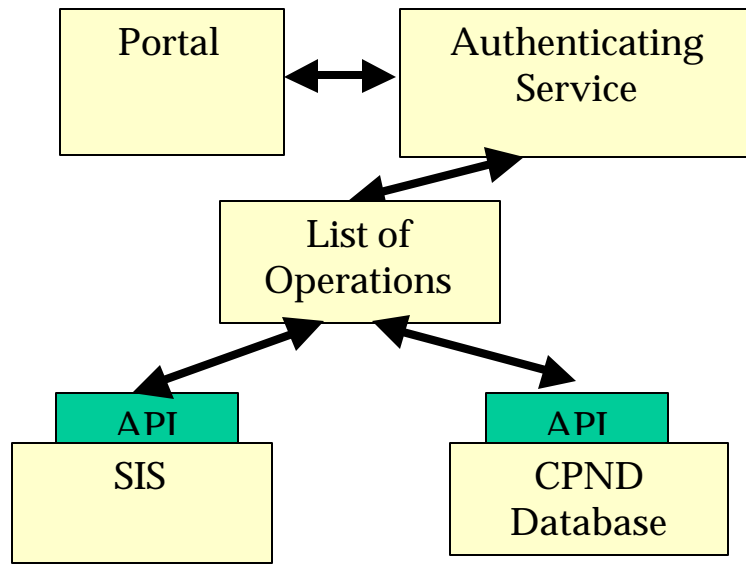
Disadvantages:

- Information access is only limited to interactions between CPND and CS
- Cannot achieve the ultimate goal of data integration and consistency (real-time updating is still not possible)
- Not expandable in the future when more departments wish to be linked

B. API solution

API is a server program, which interprets and performs XML commands. These programs reside in front of CPND and CS' database to create a common platform. The APIs will serve as a piece of middleware that will connect the respective databases to a common web-based portal. This portal will contain a list of operations that may be performed on the databases. Users (department staff) will have access to the common web-based portal at all times. Whenever they wish to perform any operations, they will first be required to log onto the portal with their ID and password. Once their identity has been authenticated, the portal will reveal a list of operations within the scope of their department's activities that they may perform.

When a user chooses to execute a specific command on the portal, the portal will send XML commands to the appropriate APIs, which in turn will retrieve the necessary information from the databases. The portal will combine all the retrieved information from the APIs and return a data file of the results to the user. Users may choose to further manipulate the returned data file. The following is an illustration of the process:



In order to implement this solution, all the APIs must conform to a set standard such as the SOAP (Simple Object Access Protocol) standard. “SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment” (Simple, 2000). (Please see Appendix H for more information on the SOAP standard) The standard for the APIs will have to be agreed upon by the two departments prior to building the APIs.

The idea for the API solution comes from UBC's IT Services department. UBC IT Services has already initiated several API projects, and hopes to have a set API standard by September 2002. If CS and CPND chooses to implement this solution, they would have the choice of pursuing either joint effort with IT Services, or an independent project. IT Services' E-commerce department has much resources reserved for such projects and would be very willing to share these resources with other UBC departments. Joining IT Services' project would allow CS and CPND to share development costs and project risk with IT Services and other departments involved. The downside is that the project may not be commenced immediately. If instead CS and CPND chose to develop the solution independently, they would have much

higher development costs. The APIs also may not conform to the same standards as APIs of other departments. If other databases wished to be joined in the future, some redevelopment of APIs may be necessary. We feel that at this time it would be more beneficial if CS and CPND joined IT Services' project for the aforementioned reasons.

The following are the advantages and disadvantages of the API solution.

Advantages:

- Users can login through a single portal and perform the tasks they want (backend is invisible to users)
- Lower development cost compared to Oracle 9i solution since costs may be shared with other departments
- More control over the resources since the API programs are developed internally within UBC
- It is expandable in the future when more departments want to be involved, the functions of the portal can also be expanded

Disadvantages:

- Does not solve data integration problems unless adding a costly data-update feature in the portal, or else department staff are still required to update their database manually
- All departments will have to agree with the common API standard
- Joining other departments such as IT Services may slow down the progression of the project

C. Oracle 9i solution

For this solution, our team recommends that UBC purchase and implement an Oracle 9i database. The Oracle 9i program will reside on a central server, maintained by an "external" department such as UBC IT Services. Both the CPND and the CS databases will be connected to the Oracle 9i database, creating a distributed database environment. The central server, on one hand, acts as a "hub" for all the other databases; however, its functions extend much further. The most appealing feature of the Oracle 9i database is its ability to connect disparate data sources. Oracle 9i has introduced some new technologies to achieve this feat.

Oracle Generic Connectivity creates transparent connections between Oracle 9i databases and low-end data sources such as Access and dBase. It uses industry database standards to create generic connectivity solutions and is included in the Oracle 9i package. This would be the

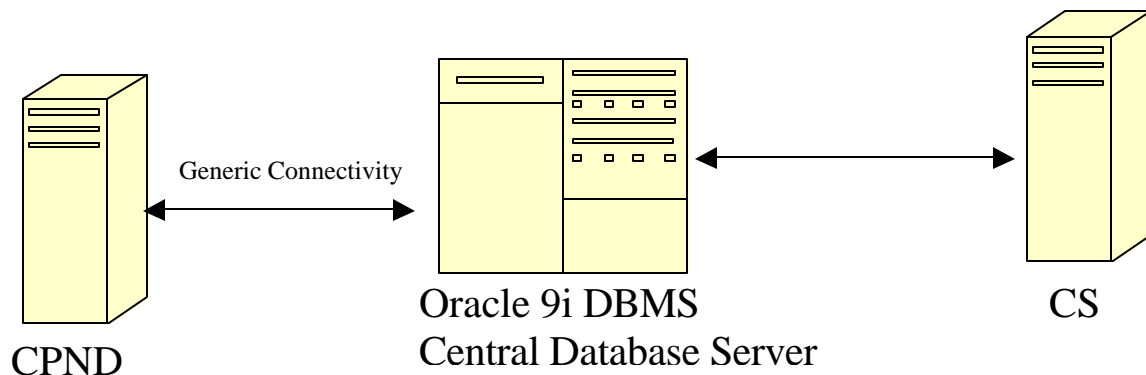
main tool used to connect the Oracle 9i database and the CPND database. Of course, older Oracle databases such as CS' database will have no problem connecting to the Oracle 9i database. Oracle 9i also features another technology called Oracle Transparent Gateway, which are tailored solutions for accessing high-end non-Oracle databases. While "Oracle Generic Connectivity relies on industry standards, ... Oracle Transparent Gateways accesses the non-Oracle systems using their native interface" (Oracle, 2002). Oracle Transparent Gateways would be used for connecting more complex data sources such as Sybase, Informix and Microsoft SQL Server. This technology would not be necessary for the two departments involved in this project, but may be useful if other departments wish to be connected in the future.

Creating these transparent gateways between the various databases will integrate all the respective databases. Now, users no longer have to know where the tables they are searching are located, as long as they know which tables they need, the main server will retrieve the data for them.

The Oracle 9i database interfaces all the other databases that connect to it. All users will only be required to access the central database, but will be able to locate information in all connected databases. The Oracle 9i database will link the various databases, joining the building tables in the CS database with those in the CPND database. The "RoomNo" field appears to be the most likely candidate that could be used to join the two databases.

The Oracle 9i solution will achieve complete data integration and data consistency. It can support large number of users accessing and updating the same data simultaneously. What it means is if someone on CPND's side was inputting new data, CS would still be able to run its required queries for reports.

The following diagram depicts the proposed solution for two departments, CPND and Classroom Services:



The following is a summary of the advantages and disadvantages of this solution:

Advantages:

- Enables data and documents from disparate systems to be accessed
- Single point of administration, also provides data integration and consistency
- Fully scalable

Disadvantages:

- Software is fairly new so the technological risks are high
- Lack of on-campus experts who are familiar with Oracle 9i
- Implementation time will be much longer due to its large scale and complexity

VII. FEASIBILITY

In different situations, the benefits and possible negative effects could vary with different solutions. We have attempted to quantify the effects that could result from implementing each proposed solution. Please refer to Appendix I and J for a detailed analysis of the costs, benefits and feasibility of each solution.

VIII. RECOMMENDATIONS

After comparing the requirements, constraints, and the feasibility studies, our team would suggest implementing the API solution, which meets the primary requirements including real time information access, system expansion, and constant information availability. This could be considered a short-term solution since it can be relatively quick to implement in order to meet current needs. When Oracle 9i is more tested and widely used, UBC can adopt Oracle 9i as a long-term solution which fulfils the primary requirements and also achieves data integration and complete data consistency. In the absence of the constraints, Oracle 9i is obviously the best solution because it generates the highest NPV (from the feasibility study), however, it cannot necessarily be justified since it depends on how useful and practical the solution is. Realistically, the API solution meets many of the constraints and fulfills the primary requirements. In addition, since IT Services has already started the API project, they can support the implementation of the API solution, which makes this solution the most feasible for now.

By employing the recommendations that we proposed, the room scheduling process will be much shorter. (See Appendix G for new process diagram) The need to contact other departments, to retrieve data from other departments' databases, and to send information back and forth

between departments will be eliminated. These improvements should significantly improve information flow and efficiency for all departments.

IX. FUTURE ISSUES

At the beginning of the project, many issues were left undecided in order to allow flexibility with the solutions generated. CPND and CS must resolve a number of issues before they can continue to proceed with the interface project.

1. Budget - set at least a range of what they are willing to spend on the project
2. Time - decide on the urgency of the project and a time frame for completion
3. Parties - decide which other departments or parties may be involved, now and in the future

Once these issues have been agreed upon, they may use the results to decide upon which solution(s) they will follow. It is possible given your decisions that our recommendations are not the most appropriate at this time for your organizations.

X. IMPLEMENTATION TIMELINE

If our recommendation is followed, here is a possible schedule for implementation:

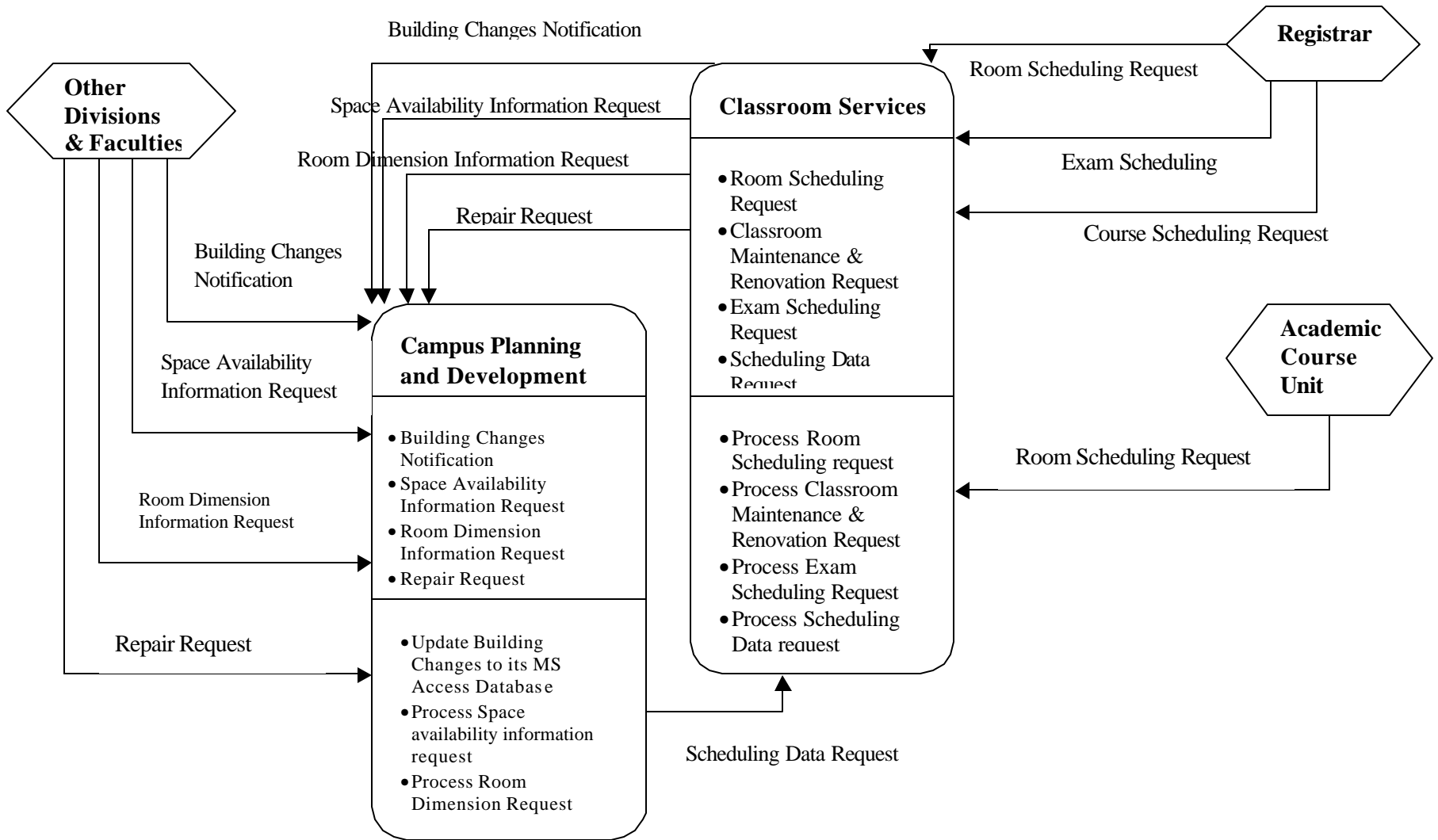
1. Resolve issues - 1 month
2. Meet with other parties (IT Services) – 1 month
3. Set standard for API - 4 months
4. Designing the web-based portal - 2 months
5. Building and implementing the API and portal - 5 months
6. Testing and Documentation - 1.5 months
7. Training - 1 month
8. Evaluation - 2 weeks
9. Maintenance

XI. BIBLIOGRAPHY

Crystal Decisions. (2001). *Crystal Reports Version 8.5*. Retrieved March 27, 2002, from http://www.crystaldecisions.com/products/crystalreports/downloads/cr85_brochure.pdf

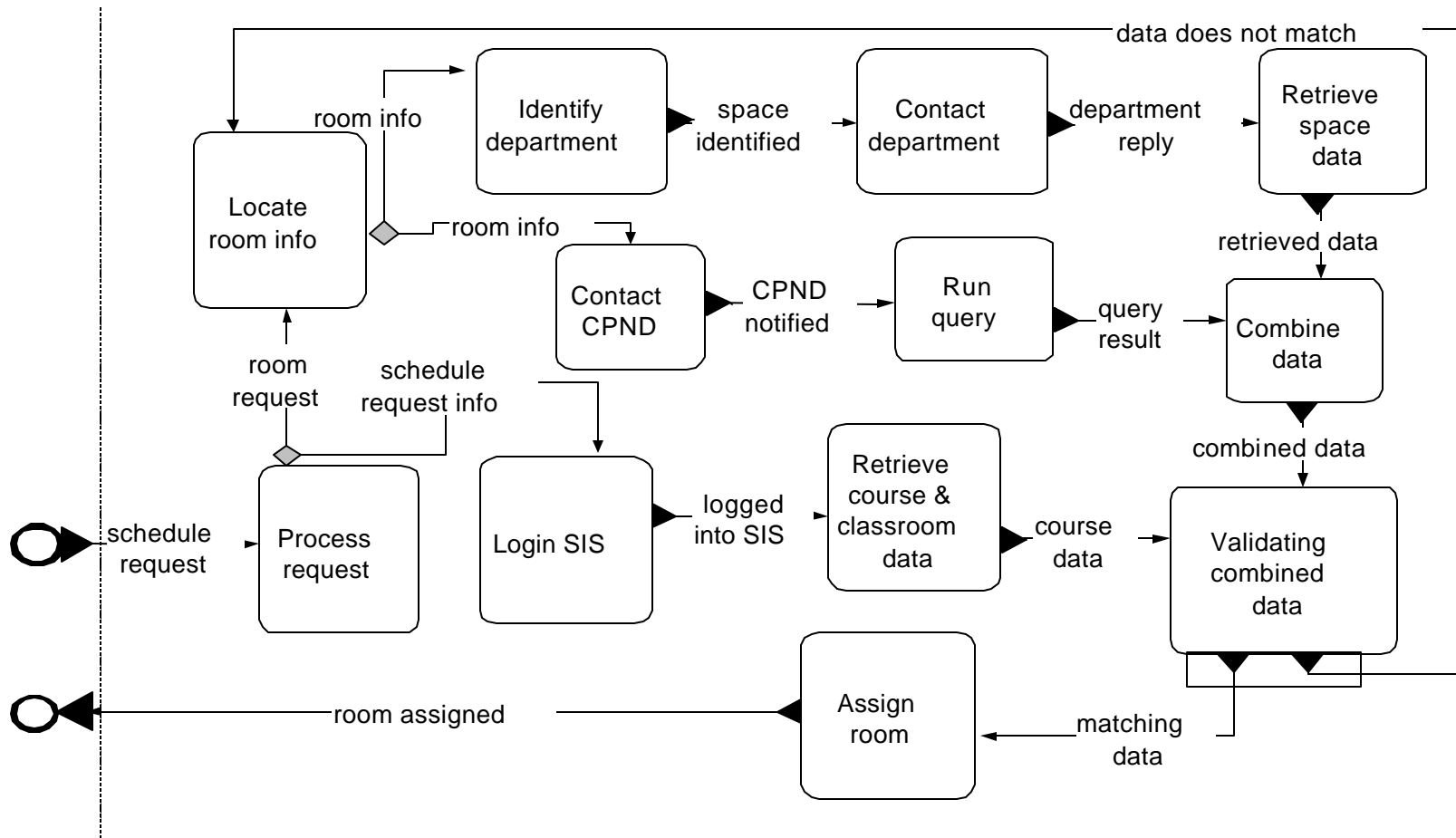
Oracle. (2002). *Oracle 9i Database*. Retrieved April 5, 2002, from http://www.oracle.com/ip/dep/otn/database/oracle9i/index.html?oracle9idb_features.html

World Wide Web Consortium. (2000, May 8). *Simple Object Access Protocol (SOAP) 1.1*. Retrieved April 15, 2002, from <http://www.w3.org/TR/SOAP/#XML>



Appendix E- Object Oriented Enterprise Modeling Diagram

Appendix F - Process Diagram (Before)



Appendix I - Cost / Benefit Analysis

Cost/Benefit Analysis (4 years):

Crystal Report Solution	Year 0	Year 1	Year 2	Year 3	Year 4
Upgrading Cost	\$ (388.50)				
Training Cost	\$ (240.00)				
Recurring Maintenance		\$ (100.00)	\$ (100.00)	\$ (100.00)	\$ (100.00)
Improved flexibility (.25 hr/day)		\$ 1,562.50	\$ 1,562.50	\$ 1,562.50	\$ 1,562.50
Total Cost	\$ (628.50)	\$ 1,462.50	\$ 1,462.50	\$ 1,462.50	\$ 1,462.50
Discount Rate = 10%					
NPV =					\$3,643.12

API Solution	Year 0	Year 1	Year 2	Year 3	Year 4
Deployment Cost	\$ (25,000.00)				
Training Cost	\$ (320.00)				
Recurring Maintenance Cost		\$ (450.00)	\$ (450.00)	\$ (450.00)	\$ (450.00)
Increased Speed of Activity (.5 hr/day)		\$ 3,125.00	\$ 3,125.00	\$ 3,125.00	\$ 3,125.00
Improved Management (1 hr/day)		\$ 7,500.00	\$ 7,500.00	\$ 7,500.00	\$ 7,500.00
Total Cost	\$ (25,320.00)	\$ 10,175.00	\$ 10,175.00	\$ 10,175.00	\$ 10,175.00
Discount Rate = 10%					
NPV =					\$6,303.07

Oracle 9i Solution	Year 0	Year 1	Year 2	Year 3	Year 4
Software Cost	\$ (1,276.00)				
Implementation Cost (through Oracle Consulting)	\$ (48,000.00)				
Training Cost	\$ (957.00)				
Recurring Maintenance Cost		\$ (1,000.00)	\$ (1,000.00)	\$ (1,000.00)	\$ (1,000.00)
Increased Speed of Activity (.5 hr/day)		\$ 3,125.00	\$ 3,125.00	\$ 3,125.00	\$ 3,125.00
Improved Management (1hr)		\$ 7,500.00	\$ 7,500.00	\$ 7,500.00	\$ 7,500.00
Saved time in update data (1 hr / day)		\$ 9,100.00	\$ 9,100.00	\$ 9,100.00	\$ 9,100.00
Error Reduction		\$ 2,500.00	\$ 2,500.00	\$ 2,500.00	\$ 2,500.00
Increase Flexibility		\$ 1,562.50	\$ 1,562.50	\$ 1,562.50	\$ 1,562.50
Total Cost	\$ (50,233.00)	\$ 22,787.50	\$ 22,787.50	\$ 22,787.50	\$ 22,787.50
Discount Rate = 10%					
NPV =					\$20,000.28

Notes:

Crystal Report Solution

1. Improved flexibility – we estimated that having the two system connected, the system staff can save .25 hour doing other stuff.
2. Training Cost = 3 employees X \$20/hour (salary for each employee) X 4hours (training hours)

API Solution

1. Deployment cost includes the cost of setting up the API standard, designing the web-based portal, and developing the API and the portal. Since the cost will be shared with IT Service, the cost shown in the spreadsheet is the portion that CS and CPND need to pay for this project.
2. Training cost = 4 employees (2 employees from CS and 2 employees from CPND) X \$20/hour (Salary for each employee) X 4 hours
3. Increased Speed of Activity – Having the common portal, the staffs can save time in entering commands and making phone calls. We estimated the API solution can save 0.5 hour per day in performing the same activity comparing to the existing system.
4. Improved Management – The classroom services will save about an hour per day in looking for where the data stores. Common portal provides all the accessible information to the users.

Oracle 9i

1. Increased Speed of Activity – Having the Oracle 9i Central Database, the staffs can save time in entering commands and making phone calls. We estimated the Oracle 9i solution can save 0.5 hour per day in performing the same activity comparing to the existing system.
2. Improved Management – The classroom services will save about an hour per day in looking for where the data stores. Distributed database environment allows different heterogeneous databases to share information.
3. Save Time in Update Data – Oracle 9i will automatically update the central server and the remote system which will save the IT stuff to update the database everyday. It also ensures data consistency. It is estimated to save about 1 hour per day.

Appendix J – Intangible Benefits / Technical Feasibility

Intangible Benefits:

<p>Crystal Report Solution:</p> <ul style="list-style-type: none"> • Increased department flexibility • Saved time in finding information • Increased efficiency • Faster decision making • More timely information 	<p>Oracle 9i Solution:</p> <ul style="list-style-type: none"> • More timely information • Single point administration • Better information quality due to data consistency • Saved time in searching information • Information process efficiency • Reduce workers frustration in allocating information • Faster decision making • Increased efficiency • Reduce boredom in manually update department databases
<p>API Solution:</p> <ul style="list-style-type: none"> • Increased department flexibility • Saved time in searching information • Increased efficiency • More timely information • Information process efficiency • Faster decision making 	

Technical feasibility:

	C.R Solution (low-end)	API Solution (mid-end)	Oracle 9i Solution (high-end)
High familiarity with DB Technology	Very low risk (Connection is easy to establish, and staff only need to run Crystal Reports to extract data)	Low risk (Once the API standard is agreed upon, development and implementation are easy)	Medium risk (Software is fairly new, and it involves a high learning curve for staff.)
Low familiarity with DB Technology	Low risk (Connection is easy to establish, and Crystal Reports is easy to learn)	Low risk (Staff do not need to know how it works but how to use the functions)	High risk (Staff require lots of training and UBC lacks Oracle 9i experts)