LSC Use Only No: LSC Action-Date:	UWUCC USE Only No. UWUCC Action-Date: Senate Action Date:
	02-80n App 4/15/03 App 4/29/03

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person Kustim Wibowo	Email Address kwibowo@iup.edu
Proposing Department/Unit MIS and Decision Sciences	Phone 357-2931

Check all appropriate lines and complete information as requested. Use a separate cover sheet for each course proposal and for each program proposal.

1. Course Proposals (check all that apply)

____New Course ____Course Prefix Change ____Course Deletion

__x__Course Revision ____Course Number and/or Title Change ____Catalog Description Change

IFMG 480 Distributed Business Information Systems	
***Current* Course prefix, number and full title**	***Proposed* course prefix, number and full title, if changing**

2. Additional Course Designations: check if appropriate

_____ This course is also proposed as a Liberal Studies Course.

_____ This course is also proposed as an Honors College Course.

_____ Other: (e.g., Women's Studies, Pan-African)

3. Program Proposals

____Catalog Description Change ____Program Revision

____New Degree Program ____Program Title Change ____Other

____New Minor Program ____New Track

***Current* program name**	***Proposed* program name, if changing**

4. Approvals		**Date**
Department Curriculum Committee	[signature]	4/8/03
Chair(s)	Elizabeth M. Pierce	4/8/03
Department Chair(s)	Louise B. Berkey	4/8/03
College Curriculum Committee Chair	[signature]	9 Apr 03
College Dean	[signature]	4/9/03
Director of Liberal Studies *		
Director of Honors College *		
Provost *		
Additional signatures as appropriate:		
(include title)		
UWUCC Co-Chairs	Gail S. Sechrist	4/15/03

* where applicable

RECEIVED
APR 23 2003
LIBERAL STUDIES

Syllabus of Record

I. **Catalog Description.**

IFMG 480 Distributed Business Information Systems — 3 class hours, 0 lab hours

Prerequisites: IFMG 250 — 3 credit hours

(3c-0l-3cr)

A study of the techniques involved in planning, designing, and implementing distributed processing systems. Distributed marketing, financial, and corporate accounting systems are included.

II. **Course Objectives.**

Students will be able to:

1. Gain the knowledge of the distributed systems where a collection of independent computers working together appears as a single coherent system to the users.
2. Understand how the users and application can interact with a distributed system in a consistent and uniform way, regardless of where and when interaction takes place.
3. Analyze and discuss the possibility to expand or to reduce the scale of the distributed systems.
4. Explain the importance of supporting heterogeneous computers and networks while offering a single system view.
5. Maintain the distributed systems to be available under any conditions.
6. Write, update, and enforce the policy and regulations that will protect the users and the resources and secure the systems.

III. **Detailed Course Outline.**

1. Introduction to the distributed systems (4 hours)
   Discuss the goals, hardware concepts, software concepts and the client-server model.
2. Communication (2 hours)
   Discuss the layered protocol, remote procedure call, remote object invocation, message-oriented communication, and stream-oriented communication.
3. Processes (4 hours)
   Discuss threads, clients, servers, code migration, and software agents.
4. Naming (2 hours)
   Discuss naming entities, locating mobile entities, removing un-referenced entities.
5. Midterm I and Evaluation Testing (2 hours)
6. Synchronization (4 hours)
   Discuss clock synchronization, logical clock, global state, election algorithms, mutual execution, and distributed transactions.
7. Consistency and Replication (2 hours)
   Discuss data-centric consistency models, client-centric consistency models, distribution protocols, and consistency protocols.
8. Fault Tolerance (2 hours)
   Discuss process resilience, reliable client-server communication, reliable group communication, distributed commit, and recovery.

9.	Security Discuss security threats, policies, mechanism, design issues, and cryptography. Security channels, access control, and security management.		(2 hours)
10.	Midterm II and Evaluation Testing		(2 hours)
11.	Distributed Object-based Systems Discuss CORBA, COM, DCOM, and GLOBE systems.		(4 hours)
12.	Distributed File Systems Discuss SUN, CODA, and other distributed file systems. Comparison of distributed file systems.		(4 hours)
13.	Distributed Document Based Systems Discuss World Wide Web and Lotus systems.		(4 hours)
14.	Distributed Coordination-based Systems Discuss several coordination models including TIB/Rendezvous and JINI.		(4 hours)
15.	Final Examination		(2 hours)

IV. **Evaluation Methods.**

1.	20%	Homework assignments, class-work, and quizzes. These will be based on material discussed in class.
2.	40%	Programming projects. About three to four projects of varying complexity based on material discussed during the semester.
3.	40%	Examinations. The examinations consist of what-if questions, short-essays, analysis, and explanations. Three exams (10%, 10%, and 20%) will be administered during the semester.

Grading Scale: A: $\geq$90% B: 80-89% C: 70-79% D: 60-69% F: <60%

V. **Course Attendance Policy.**

In accordance with University policy, individual faculty will denote an attendance policy on specific course syllabi.

VI. **Required Textbook(s), Supplemental Books and Readings.**

Required:
Tanenbaum & Can Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2002.

Supplemental:
White, Data Communications and Computer Networks: A Business User's Approach, Course Technology, 2001.

Reading:
Burd, Systems Architecture: Hardware and Software in Business Information Systems, 2nd Edition, Course Technology, 1998.

VII. **Special Resource Requirements.**

No special resource requirements.

VIII. **Bibliography.**

1. Alezander & Hollis, Developing Web Applications with Visual Basic .NET and ASP.NET, John Wiley & Sons, 2002.
2. Atkinson, Core PHP Programming, 2nd Edition, Prentice Hall, 2001.
3. Bai, Java Server Pages, Course Technology, 2003.
4. Bonazzi & Stokol, Oracle8i & Java From Client/Server to E-commerce, Prentice Hall, 2001.
5. Bontempo & Saracco, Database Management: Principles and Products, Prentice Hall, 1995.
6. Conger & Mason, Planning and Designing Effective Web Sites, Course Technology, 2000.
7. Connolly & Berg, Database Systems: A Practical Approach to Design, Implementation, and Management, 3rd Edition, Addison Wesley, 2002.
8. Deitel, Deitel & Neito, Internet & World Wide Web: How To Program, 2nd Edition, Prentice Hall, 2002.
9. Elmasri & Navathe, Fundamentals of Database Systems, Addison Wesley, 2000.
10. Kaparthi & Kaparthi, Macromedia ColdFusion, Course Technology, 2002.
11. Kroenke, Database Processing: Fundamentals, Design & Implementation, 8th Edition, Prentice Hall, 2002.
12. Kroenke, Database Concepts, Prentice Hall, 2003.
13. Lewism, Bernstein, & Kifer, Database and Transaction Processing: An Application-Oriented Approach, Addison Wesley, 2002.
14. Martin & Shelly, Visual Basic .NET at Work, John Wiley & Sons, 2002.
15. McFadden, Hoffer, & Prescott, Modern Database Management, 6th Edition, Addison Wesley, 2002.
16. Morneau & Batistick, Active Server Pages, Course Technology, 2001.
17. Morrison & Morrison, Enhanced Guide to Oracle8i, Course Technology, 2001.
18. Reselman, Active Server Pages 3.0 By Example, Que, 2000.
19. Riccardi, Principles of Database Systems with Internet and Java Applications, Addison Wesley, 2001.
20. Smith, ASP.NET By Example, Que, 2002.
21. Welling & Thompson, PHP and MySQL Web Development, SAMS, 2001.

COURSE ANALYSIS QUESTIONNAIRE

Section A: Details of the Course

A1 How does this course fit into the programs of the department? For what students is the course designed? (majors, students in other majors, liberal studies). Explain why this content cannot be incorporated into an existing course.

This is an elective course for students majoring in MIS. Other students in the Eberly College of Business and IT may take this course as an elective.

A2 Does this course require changes in the content of existing courses or requirements for a program? If catalog descriptions of other courses or department programs must be changed as a result of the adoption of this course, please submit as separate proposals all other changes in courses and/or program requirements.

This course will not require changes in the content of existing courses.

A3 Has this course ever been offered at IUP on a trial basis (e.g. as a special topic?) If so, explain the details of the offering (semester/year and number of students).

This course has never been offered as a special topic.

A4 Is this course to be a dual-level course? If so, please note that the graduate approval occurs after the undergraduate.

This course is not intended to be dual level.

A5 If this course may be taken for variable credit, what criteria will be used to relate the credits to the learning experience of each student? Who will make this determination and by what procedures?

This course is not intended to be taken for variable credit.

A6 Do other higher education institutions currently offer this course? If so, please list examples (institution, course title).

A number of institutions offer similar back-end office programming for MIS students. For example:

University of Illinois at Chicago: MIS 412 Distributed Business Systems
Organizational aspects and underlying concepts of distributed business systems, decentralization versus centralization issues, costs of distributed computing, and performance evaluation measures. Prerequisites: IDS 400 or 401, and credit or concurrent registration in IDS 410; or consent of the instructor.

Arizona State University: CIS 535 Distributed Information Systems
Introduction to distributed systems and their impact on information systems in business. Prerequisite: ACC 587 or CIS 505

University of Mississippi : MIS 405 Distributed Business Information Systems.
Course focuses on the conceptual and practical aspects of modern operating systems and development of concurrent and distributed software systems. Exploration of operating systems as managers of shared and distributed (or remote) resources. Design and implementation of complex concurrent and distributed software systems using the more recent technologies, computing tools, communication frameworks and programming languages.

A7 Is the content, or are the skills, of the proposed course recommended or required by a professional society, accrediting authority, law or other external agency? If so, please provide documentation.

Although not required, this course is useful for MIS students as an elective.

Section B: Interdisciplinary Implications

B1 Will this course be taught by instructors from more than one department? If so, explain the teaching plan, its rationale, and how the team will adhere to the syllabus of record.

The course is designed to be taught by one MIS instructor per semester.

B2 What is the relationship between the content of this course and the content of courses offered by other departments? Summarize your discussions (with other departments) concerning the proposed changes and indicate how any conflicts have been resolved. Please attach relevant memoranda from these departments that clarify their attitudes toward the proposed change(s).

This course does not overlap with any other courses at this university. Although other departments may offer courses with similar topics, this course is specifically designed for the needs, interests, and context required for our MIS majors. This course already exists. The update of the syllabus of record is mostly an updating of the tools and methods used to implement the content of the course. The basic nature and purpose of this course has not been changed.

B3 Will this course be cross-listed with other departments? If so, please summarize the department representatives' discussions concerning the course and indicate how consistency will be maintained across departments.

This course is proposed by the MIS-DS Department and will not be cross listed.

B4 Will seats in this course be made available to students in the School of Continuing Education?

Seats will be made available to Continuing Educations students meeting the prerequisite.

Section C: Implementation

C1 Are faculty resources adequate? If you are not requesting or have not been authorized to hire additional faculty, demonstrate how this course will fit into the schedule(s) of current faculty. What will be taught less frequently or in fewer sections to make this possible? Please specify how preparation and equated workload will be assigned for this course.

Faculty resources are adequate.

C2 What other resources will be needed to teach this course and how adequate are the current resources? If not adequate, what plans exist for achieving adequacy? Reply in terms of the following:

*Space The Eberly classrooms are adequate for this course.

*Equipment The Eberly computer labs are adequate for this course.

*Laboratory Supplies and other Consumable Goods The MIS-DS Department has enough software and computer supplies to support this course. However, the computer hardware and software will require periodic updates to meet the technological advancements and requirements.

*Library Materials The Stapleton Library has enough reading material for this course.
*Travel Funds No travel funds are needed.

C3 Are any of the resources for this course funded by a grant? If so, what provisions have been made to continue support for this course once the grant has expired? (Attach letters of support from Dean, Provost, etc.)

No resource for this course will be funded by a grant.

C4 How frequently do you expect this course to be offered? Is this course particularly designed for or restricted to certain seasonal semesters?

Once a semester.

C5 How many sections of this course do you anticipate offering in any single semester?

At least one section.

C6 How many students do you plan to accommodate in a section of this course? What is the justification for this planned number of students?

Approximately 30 students will be accommodated in a section of the course.

C7 Does any professional society recommend enrollment limits or parameters for a course of this nature? If they do, please quote from the appropriate documents.

No professional society recommends enrollment limits or parameters for this course.

C8 If this course is a distance education course, see the Implementation of Distance Education Agreement and the Undergraduate Distance Education Review Form in Appendix D and respond to the questions listed.

Presently, this course is not a distance education course.

Section D: Miscellaneous

Include any additional information valuable to those reviewing this new course proposal.

1. **Summary of the proposed revisions.**

   Syllabus of Record Change

2. **Justification/rationale for the revision.**

   The proposed revision is meant to bring the syllabus of record in line with what is currently being taught in the course. Because students are already learning materials based on the new syllabus of record, there will be no discontinuity between students who took the course under its old title and students taking the course in the future.

3. **Old Syllabus of Record Format: IM480 Distributed Business Information Systems.**

   This course is so old that we could not locate an old syllabus of record. See current syllabus.

IFMG 480 Distributed Information Systems	Spring – 2003
Section 001 (TR: 1:15 – 2:45 pm.)	Classroom: 213

Instructor: Dr. Kustim Wibowo	Office Hours: TR: 7:30 – 8:00 am. and 11:30 am – 1:00 pm; W: 9:30 – 10:30 am. or by appointment.
Office: ECB 207E	Phone: 357-2931
E-mail: kwibowo@iup.edu	Web Page: http://www.eberly.iup.edu/kwibowo/

Text: Distributed Systems, Principles and Paradigms. By: Andrew S. Tanenbaum and Maarten van Steen.

Schneider web site: http://www.prenhall.com/tanenbaum

Supplement: articles and instructor generated materials (handouts) and online materials.

Catalog Description: A study of the techniques involved in planning, design, and implementation of distributed processing systems. Distributed marketing, financial, and corporate accounting systems are included.

Course Objectives:

In this course the student will study distributed information systems. The students will learn how to design and create the distributed systems, incorporate all programming techniques necessary, and utilize several file/database processing methods to develop distributed business applications. More specifically, this course has six main objectives.

1. Students will gain an understanding of the principles and paradigms of distributed systems.
2. Students will be introduced to different types of distributed operating systems, network operating systems, middleware, and internal organization, such as, user-interface level, processing level, and data level.
3. As part of the design and construction of distributed information systems for business applications, students will be exposed to hardware, software, organizational uses of information, and related information technology concepts.
4. Students will see how distributed information systems applications can be used to support an organization's mission, goals and objectives.
5. Students should be able to explain how distributed information systems for business applications can be used to support an employee's decision-making, goal setting, trustworthiness and empowerment within an organization.
6. Students will learn how to develop the distributed information systems and software applications and utilize them in a controlled environment.

Prerequisites: IFMG 350 **Business Systems Technology**

Beginning with the Summer 2000 term, there will be absolute enforcement of every prerequisite requirement for the coursework offered by the Eberly College of Business & Information Technology. This means that students cannot postpone prerequisites and take them after the course in question.

The dean's office is responsible for monitoring course prerequisites. Students who manage to register for coursework in spite of the fact that they do not have the appropriate prerequisite will be subject to unilateral withdrawal after the course has commenced. At that time, no appeal will be accepted and adding a class after the official registration period will not be approved.

Course Withdrawal:

The university individual course withdrawal deadline date of Tuesday, March 25, 2003 will be enforced. A request for a deadline waiver must be sought through the Assistant Dean for Academic Services in Room 208. Requests will only be granted: 1) "contingent upon documentation of catastrophic circumstances" as stated in the IUP Undergraduate Catalog; and/or 2) through written feedback from the instructor noting advisement to the student to postpone withdrawing pending an additional test or assignment.

Course Outline:

A. Introduction

Distributed systems make it easy for users to access remote resources and to share them with other users in a controlled way. We will study that a distributed system should easily connect users to resources, it should hide the fact that resources are distributed across a network; it should be open; and it should be scalable.

B. Communication

Communication in distributed systems is always based on low-level message passing as offered by the underlying network. Modern distributed systems often consist of thousands or even millions of processes scattered across an unreliable network such as the Internet. We will discuss the rules that communicating processes must adhere to, known as protocol, and concentrate on structuring those protocol in the form of layers. We then look at four widely-used models for communication: Remote Procedure Call (RPC), Remote Method Invocation (RMI), Message-Oriented Middleware (MOM), and stream.

C. Process

Communication takes place between processes. We will take a closer look at how the different types of processes play a crucial role in distributed systems. An important issue is moving processes between different machines. Process migration or more specifically, code migration, can help achieving scalability, but can also help to dynamically configure clients and servers.

D. Naming

Names play an important role in all computer systems. They are used to share resources, to uniquely identify entities, to refer to locations, and so on. An important issue with naming is that a name can be resolved to the entity it refers to. Name resolution thus allows a process to access the named entity. In distributed systems, the implementation of a naming system is itself often distributed across multiple machines. We will concentrate on three different, important ways that names are used in distributed systems. First, names based on the organization and implementation of human-friendly names. Second, names are used to locate mobile entities. Therefore, naming systems for human-friendly names are not particularly suited for supporting large numbers of mobile networks. Alternative organizations are needed, such as those being used for mobile telephony where names are location-independent identifiers. Third, names that are no longer referenced, or can no longer be located and accessed, should be automatically removed. This operation is also known as garbage collection, with the collection of large-scale distributed objects, automatically collecting un-referenced objects is becoming increasingly important.

E. Synchronization

We have looked at processes and communication between processes. While communication is important, it is not the entire story. Closely related is how processes cooperate and synchronize with one another. Cooperation is partly supported by means of naming, which allows processes to at least share resources, or entities in general. We will examine how processes can be synchronized. As it turns out, synchronization in distributed systems is often much more difficult compared to synchronization in uni-processor or multiprocessor systems.

F. Consistency and Replication

An important issue in distributed systems is the replication of data. Data are generally replicated to enhance reliability or improve performance. One of the major problems is keeping replicas consistent. This means that when one copy is updated, we need to ensure that the other copies are updated as well, otherwise the replicas will no longer be the same. We will take a closer look at what consistency and replicated data actually means and the various ways that consistency can be achieved.

G. Fault Tolerance

A characteristic feature of distributed systems that distinguishes them from single-machine systems is the notion of partial failure. A partial failure may happen when one component in a distributed system fails. This failure may affect the proper operation of other components, while at the same time leaving yet other components totally unaffected. An important goal in distributed system design is to construct the system in such a way that it can automatically recover from partial failures without seriously affecting the overall performance. We will take a closer look at techniques for making distributed systems fault tolerant. We will study process resilience and reliable multicasting. Process resilience incorporates techniques by which one or more processes can fail without seriously disturbing the rest of the system. Reliable multicasting guarantees success of message transmission to a collection of processes.

H. Security

Security in distributed systems can roughly be divided into two parts. One part concerns the communication between users or processes. The principal mechanism for ensuring secure communication is that of a secure channel. The other part concerns authorization, which deals with ensuring that a process gets only those access rights to the resources in a distributed system to which it is entitled. In addition to traditional access control mechanisms, we also focus on access control when have to deal with mobile code such as agents.

I. Distributed Object-based Systems

We will discuss various paradigms that are used to organize distributed systems. The first paradigm consists of distributed objects. The notion of an object plays a key role in establishing distribution transparency. In principle, everything is treated as an object and clients are offered services and resources in the form of objects that they can invoke.

J. Distributed File Systems

Considering that sharing data is fundamental to distributed systems, it is surprising that distributed file systems form the basis for many distributed applications. Distributed file systems allow multiple processes to share data over long periods of time in a secure and reliable way. We will study two different distributed file systems, Sun Network File System (NFS) and CODA (Descendant of the Andrew File System). We also will consider three other additional systems: Plan 9, xFS, and SFS.

K. Distributed Document-Based Systems

The strength of the Web lies in the relative simplicity of its paradigm: everything is a document. Document-based systems give users the notion of a document as a simple and powerful means to exchange information. The model is easy to understand and often comes close to what people are already used to in daily life. For example, in an office environment, communication often takes place through memos, notes, reports, and so on. In this section, we will study the document-based systems, notably the Web.

L. Distributed Coordination-Based Systems

In this section, we will consider a newer generation of distributed systems that assumes that the various components of a system are inherently distributed and that the real problem in developing such systems lies in coordinating the activities of different components. In other word, instead of concentrating on the transparent distribution of components, emphasis lies on the coordination of activities between those components.

Teaching Method: Lectures covering the course material, homework and projects on development and application on appropriate distributed information systems, and exams.

Class Participation: Class attendance is essential. Every student is expected to be involved in class, homework, and projects discussions.

Course Evaluation:

Brief Description	**%**
12 Homework	15
2 Projects (15% and 20%)	35
2 Exams (20%, and 25%)	45
Class attendance	5
Total	100

Grading Scale: A 90 – 100; B 80 – 89; C 70 – 79; D 60 – 69; F 59 - below. **No extra credit will be given.**

Tentative Schedule for Lectures, Homework, Exams and Projects (revisions will be announced)

Week	Chapter	HW	Exam	Project
1	1, 2	-	-	Project1&2 description
2	3	1	-	-
3	3	2	-	-
4	3 & 4	3	-	-
5	4 & 5	4	-	-
6	5 & 6	5	-	-
7	6 & 7	-	Exam-1	Project1 due February 28
8	Spring Break	-	-	-
9	6 & 7	6	-	-
10	8	7	-	-
11	8 & 9	8	-	March 25 withdrawal deadline
12	9 & 10	9	-	-
13	9 & 10	10	-	-
14	11	11	-	Project2 April 18
15	Review	12	-	-
16	-		Final Exam	