THE DESIGN, IMPLEMENTATION AND EVALUATION OF INCA: A PROPOSAL FOR AN AUTOMATED SYSTEM FOR APPROVAL CODE ALLOCATION

A THESIS PROPOSAL SUBMITTED TO THE THESIS COMMITTEE

By

Jitender Miglani

Thesis Committee:

Philip Johnson, Chairperson

Wesley Peterson

Chris Hundhausen

September 24, 2001

ABSTRACT

The ICS department of the University of Hawaii has faced problems surrounding approval code distribution as its enrollment has increased. The manual system for approval code allocation was time-consuming, ineffective and inefficient. INCA is designed to automate the task of approval code allocation, improve the quality of course approval decisions, and decrease the administrative overhead involved in those decisions.

Based upon informal feedback from department administrators, it appears that INCA reduces their overhead and makes their life easier. What are the old problems that are solved by INCA? Does INCA introduce new kinds of problems for the administrator? What about the students? Are they completely satisfied with the system? In what ways does the system benefit the department as a whole?

In this thesis, I will discuss design, implementation and evaluation of INCA. I will evaluate INCA from the viewpoints of students, administrators, and the department. I will do an email analysis to prove that INCA reduces the administrative overheads. I will conduct a user survey to investigate whether INCA improves the predictability and understandability of students. Finally, I will analyze the INCA database to extract the information useful to the departments for course curriculum planning. The evaluation of INCA will provide us with useful insights for future improvements of INCA and improving the student experience with academic systems in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EIS Enterprise Information Systems

EJB Enterprise Java Beans

ICS Information and Computer Sciences

ITS Information and Technology Services

INCA INternet Course Allocation

ISIS Integrated Student Information System

J2EE Java 2 Enterprise Edition

JSP Java Server Pages

PA'E Phone Assisted Enrollment

SIS Student Information Systems

UH University of Hawaii

US United States

Chapter 1. Introduction

1.1 The course allocation problem

Universities in United States use “student information systems” (SIS) to manage student records. These systems manage information on student admissions, student academic activities, student financials, and course enrollment.

The University of Hawaii (UH) uses ISIS (Integrated Student Information System) to manage student and course information. ISIS manages admissions information, student biographic information and student academic histories. It also manages the course catalog, schedule of classes and course enrollment. In Fall 1986, ISIS was used to carry out the registration for the first time. In Fall 1996, ISIS was interfaced with the PA’E phone registration module. PA'E (Phone-Assisted Enrollment) is a computer system that can be accessed by either a touchtone phone or a personal computer connected to the Internet. PA’E allows students to view or listen to course schedules and register for classes.

PA’E uses a single priority rule - seniority - to allocate seats in a class. According to this priority rule, seniors get to register before juniors, juniors before sophomores, and so on. However, some departments need to apply their own rules and eligibility criterions before students can be offered seat in a course. These departments needed to take control away from PA’E for the purposes of deciding who would receive seats in their courses.

Various departments within the College of Arts and Sciences, the College of Business, the College of Education, the College of Engineering, and the School of Travel Industry Management offer classes with restricted enrollment. If a student wishes to

enroll in courses offered by these departments, the student must obtain the approval of an authorized representative of the department or college offering the course. Once approved, the student is given "phone approval codes" to register over PA'E for the restricted classes.

1.2 Issues with approval code allocation in the ICS department

The ICS (Information and Computer Sciences) department at UH is an interesting example of a department with course allocation problems. Each semester, over 700 ICS majors and graduate students, along with hundreds of non-ICS majors and unclassified graduate students, compete for seats in the dozen or so ICS classes offered. To allocate approval codes to the students, usually an administrator from the department receives the requests for approval codes from the students via email, phone or in-person. Then, the ICS administrator grants or denies the request based upon the seats available and the student's background. This manual approach to approval code allocation creates the following problems:

1. **There is an administrative overhead involved in reading and responding to thousands of emails**. Every semester, the administrator needs to respond to thousands of emails from students. The emails usually contain course requests and queries regarding courses offered and student's qualifications. For every course request, the administrator checks student records and seats available in a course and accordingly accepts or denies a course request. If the request is denied, the administrator provides an appropriate explanation.
2. **Students miss their graduation dates if they don't get necessary courses on time.** The biggest concern among the students is that they won't graduate on time if they

don't get necessary courses. They write emails to the administrator and the department to complain if they don't get approval codes on time.

3. **There is no concrete data available to the department regarding courses in demand**. With thousands of requests to process, it is no longer clear whether the best possible decisions are being made on who should be admitted to the program. Furthermore, the requests themselves provide interesting, unexploited information about the kind of student interest in program and the kinds of classes they would like. Unfortunately, the overhead involved in this kind of analysis is too high to perform for each class and each semester using the manual system.

1.3 A guided tour of INCA

INCA is a web-based approval code allocation system designed to solve problems with manual approach to approval code allocation. The basic idea behind operation of INCA is simple. The students use a web-browser and submit a list of course requests to the INCA server. The INCA server collects all the course requests over a period of time. When the administrator is ready to allocate approval codes, the system selects the top ranked students based upon a set of rules and presents administrator with a list of those students for review. After approval by the administrator, the system automatically sends emails to all of the students simultaneously, informing each of them about the decision made on their course requests and providing them with approval codes, if appropriate. This process is repeated until the semester begins.

There are three types of users of INCA: students, course administrators and site administrators. Students use INCA, primarily, to add, delete, or modify course requests. Course administrators use INCA to allocate approval codes and edit student and course

data. Site administrators perform the tasks of site initialization and site maintenance. The following sections describe the student and the administrator usage scenarios.

1.3.1 INCA student scenarios

1. **Students visit the INCA website**. The students visit the website http://inca.ics.hawaii.edu to request the approval codes for courses they wish to enroll in. A student login page is presented upon visiting the specified link. Students need to register with INCA in order to login into the system. The password to enter into the system is emailed to the students once they register with INCA. The INCA login page also acts like a communication point between the administrators and students as the administrators can post any news on the student login page.

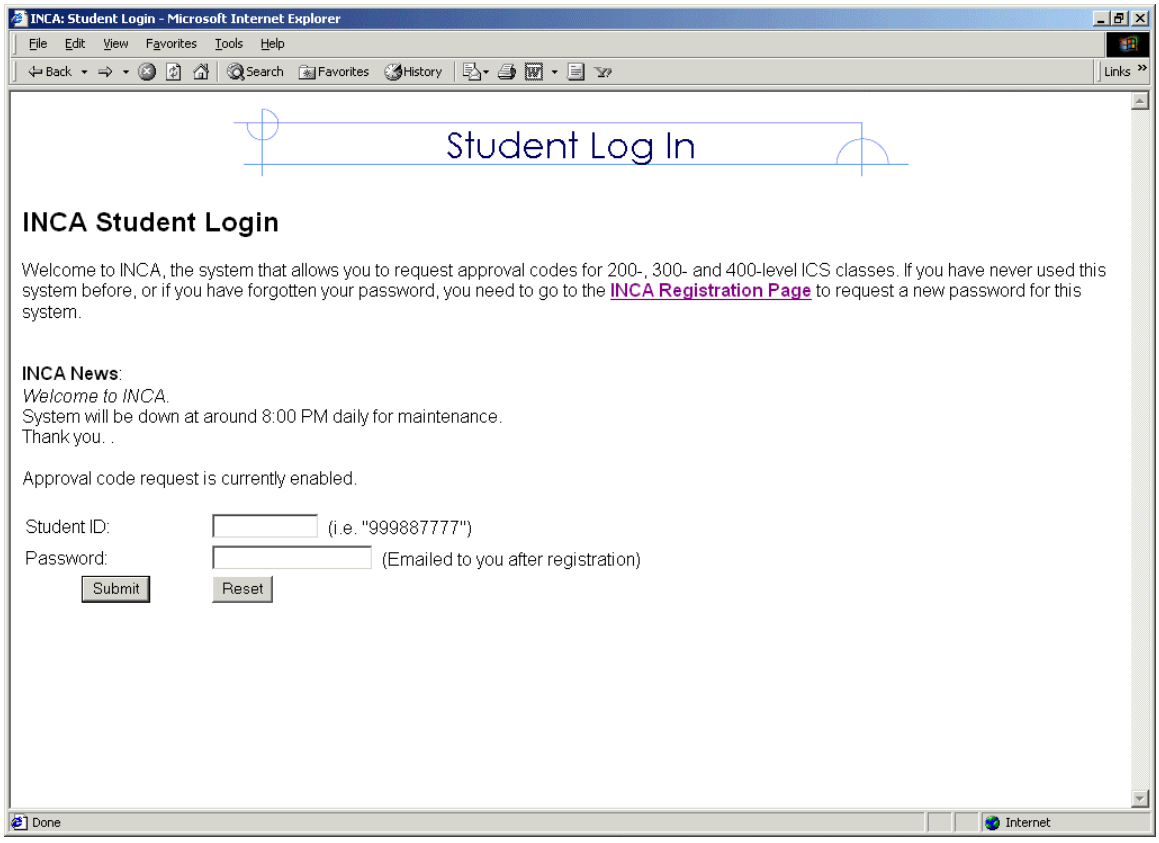


Figure 1 - Student login page

2. **Students register with INCA**. The first time students visit INCA, they must register in order to get a password. The registration process requires the students to have an ITS account, which is a user account on the UH UNIX system. If a student doesn't have an ITS account, they generally need to obtain one first. However, there is one exception. New and transferring UH students can email uhmics@hawaii.edu to explain their situation and the INCA course administrator can choose to manually register them. If students forget their password, they can re-register using the INCA registration page and a new password will be emailed to them. The requirement that students have an ITS account is an important component of user security mechanisms in INCA, discussed more thoroughly in chapter three.

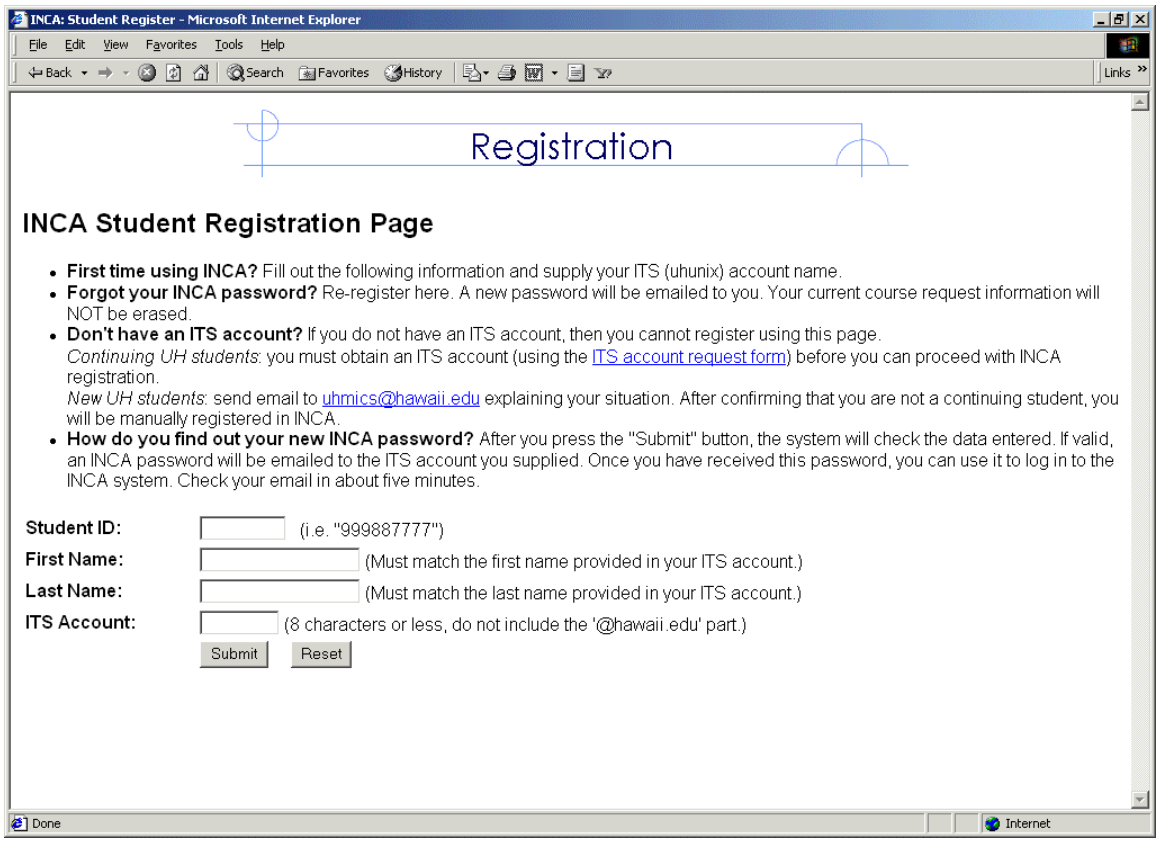


Figure 2 - Student registration page

3. **Students enter into the system and learn how to use it**. If the registration is successful, students obtain a password that allows them to enter into the system. Upon successful login, they presented with the student guide page that instructs them on how to use the system. The basic steps involve: checking their status for any missing or incorrect information; adding course requests; viewing course requests; and logging out of the system. In addition, the student guide page also explains how to do other tasks such as: modifying course requests; viewing the entire list of courses with qualification explanations; checking their rank in the approval queue and associated priority explanations; and changing their passwords.

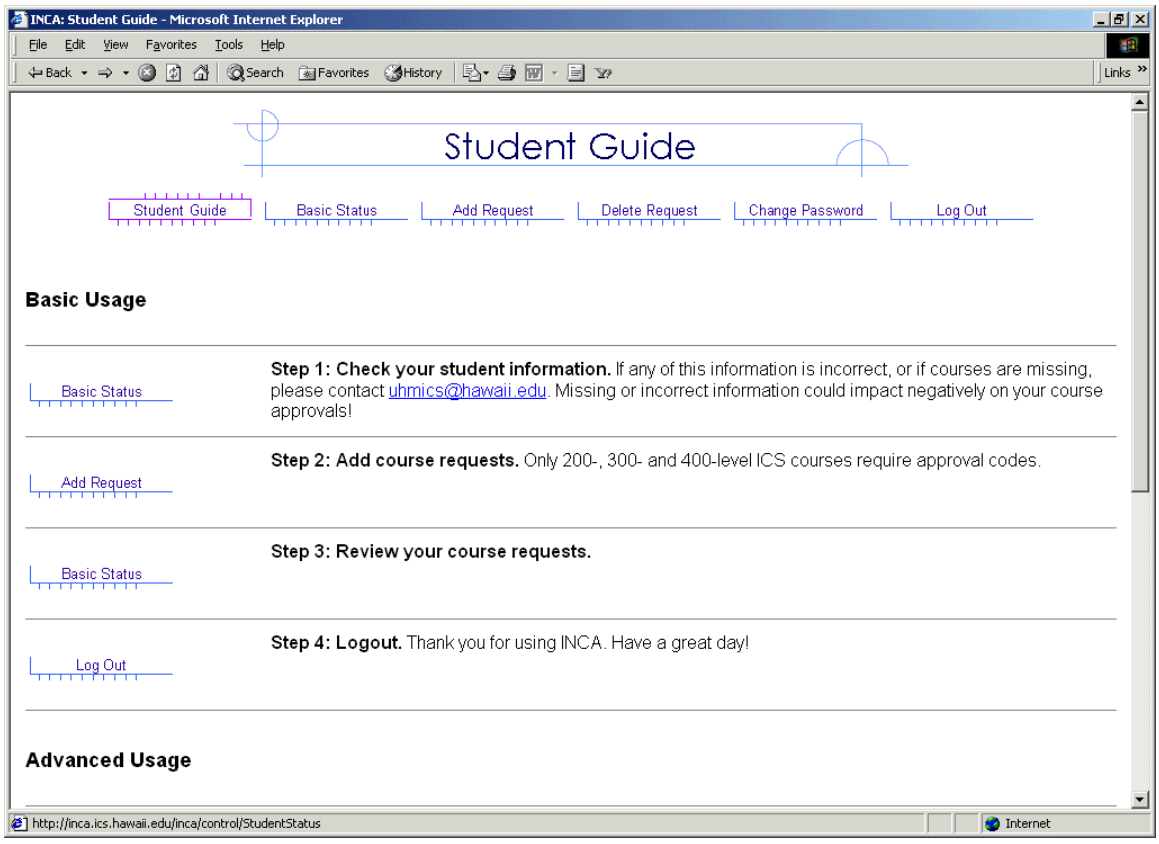


Figure 3 - Student guide page

4. **Students check their basic status and ICS course history**. The student status page has three sections: basic status, ICS course history, and request and approvals. The basic status section displays the student's GPA, and whether they are ICS majors, graduating seniors, honors students or continuing students. It also displays their major and whether they are at graduate or undergraduate level. The ICS course history section shows the courses taken at ICS department in the past and associated grades. Students review their basic status information and ICS course history and notify ICS administrator for any incorrect or missing information. Finally, the student status page also displays the information related to course requests made by students and the approval codes given to them.

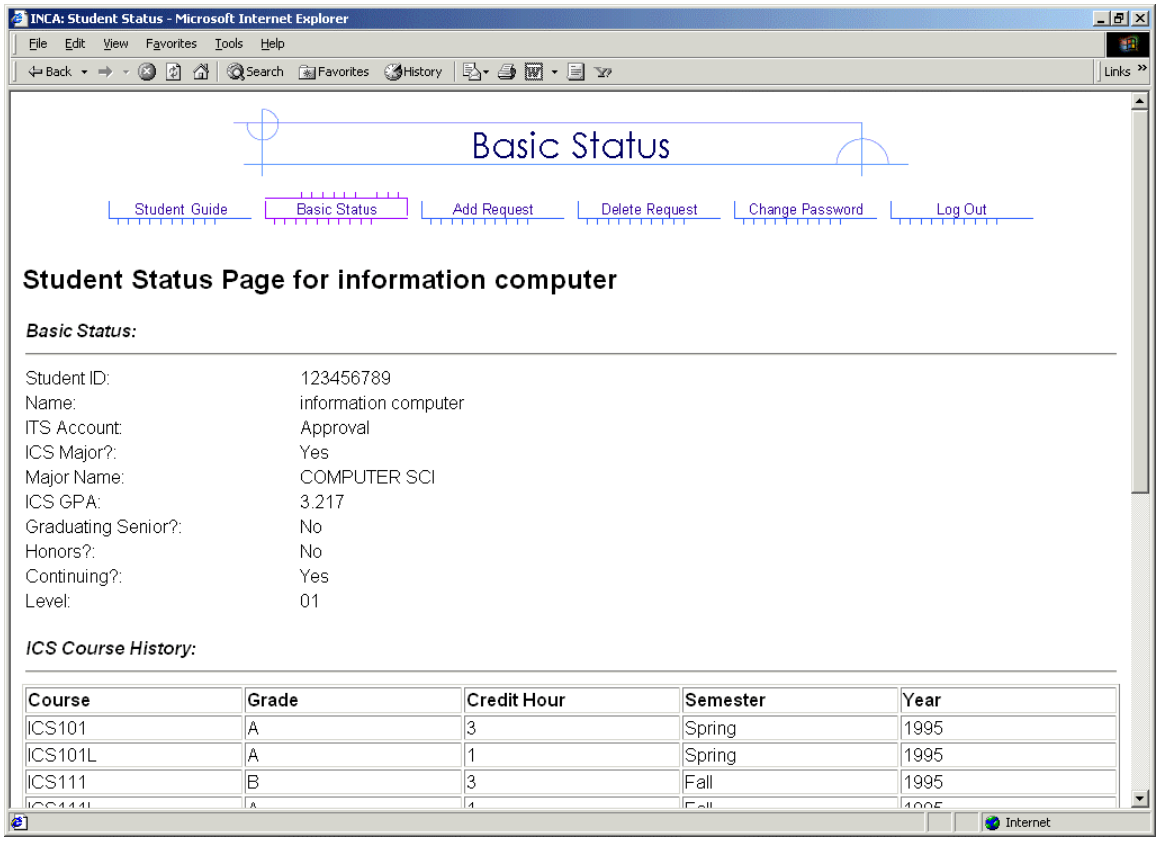


Figure 4 - Student basic status page

5. **Students request the courses.** The add request page presents the students with the courses they are qualified for and other information related to each of those courses. The information includes course name, sections for that course, the pre-requisites, co-requisite or con-current course requirements, the total number of seats offered in that course, the total requests that have been made, and the total number of available approval codes for that course. It also displays information on whether students are qualified for a course or not, their fine-grained priority points and their approximate ranking based upon those priority points. While submitting course requests, the students can rank requests from most preferred to least preferred.

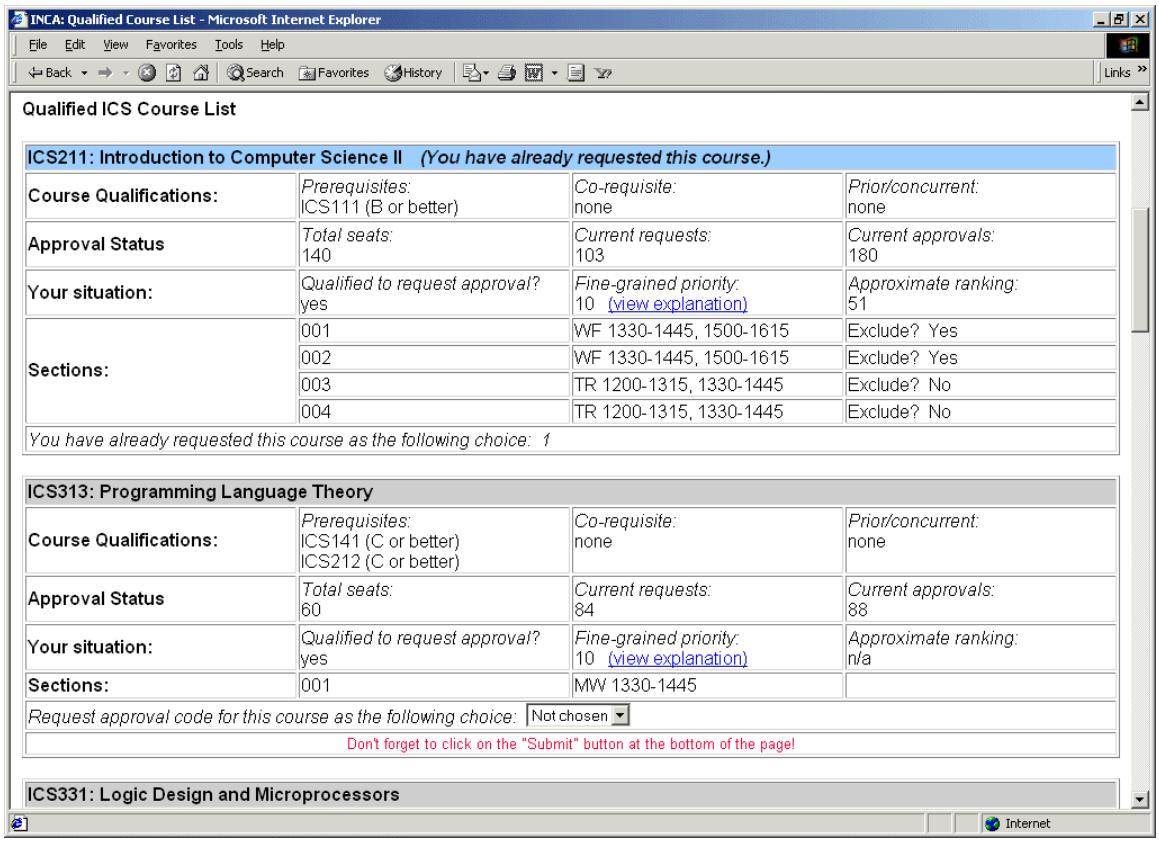


Figure 5 - Student qualified course request page

6. **Students view their priority points for a course**. The students can see the explanation for the fine-grained priority points on the priority point explanation page. The page lists various rules along with their description. It also displays the status of the students and priority points earned by them according to the rules. The rules are designed to favor ICS department majors and students with good academic credentials. The rules include weightings based upon the choice of requests and give more priority points to graduating students. The rules give negative points to the students repeating a particular course or taking a previously withdrawn course.

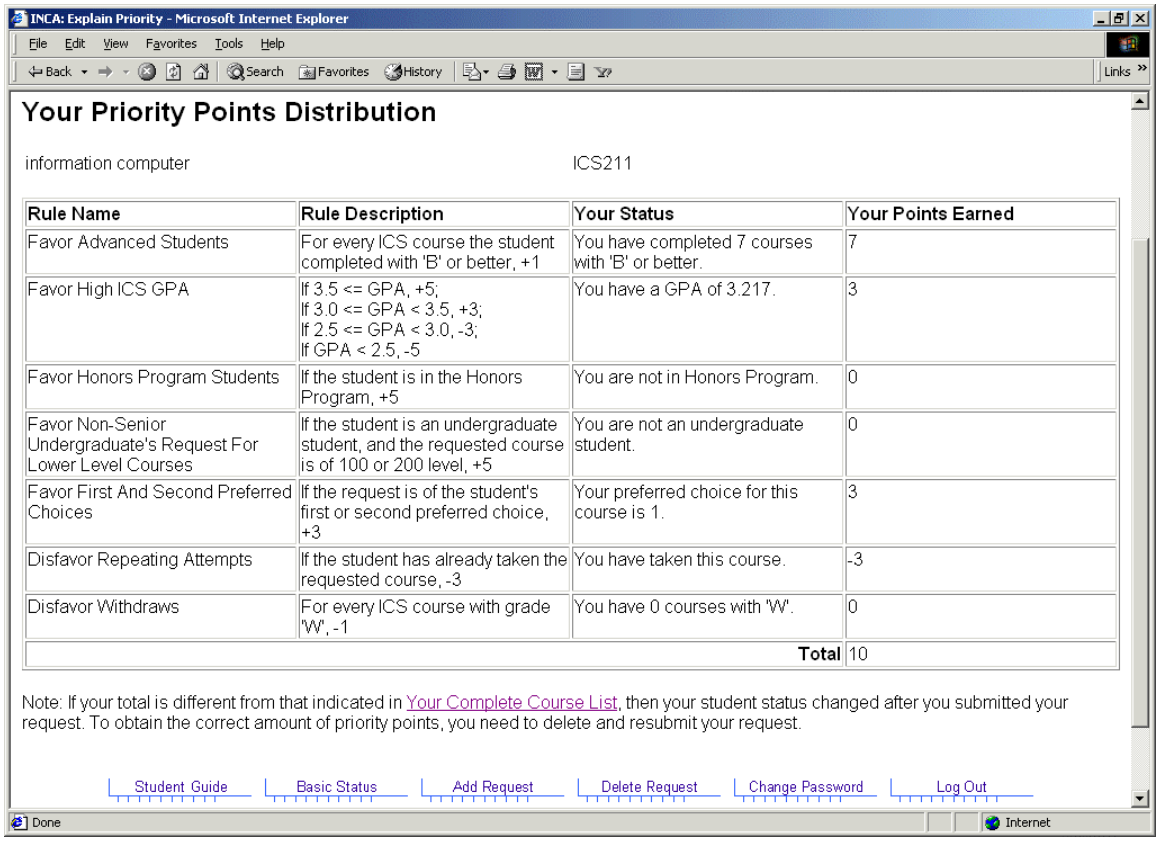


Figure 6 - Priority point explanation page

In addition to these basic tasks, students can do the following tasks:

1. **Check their ranks in real time**. Students can check their most current ranking as their place in approval code queue keeps changing with more and more students making requests. Students can log into INCA every few days, see their rank, and guess their chances of getting approval codes for popular courses.
2. **View list of all ICS courses requiring approval codes**. Students can view the entire list of ICS courses, which explains the qualification criterions for all courses and the standing of the students with respect to those qualifications.
3. **Modify course requests**. Students can delete their old course requests and submit new ones, to modify their course requests.
4. **Change their password**. Students can change their old passwords to new ones.

1.3.2 INCA administrator scenarios

The INCA course administrator and the INCA site administrator use same administrator account to do their tasks. Both of them are presented with same main page upon successful login. The administrator main page lists all the tasks. INCA site administrator uses some of the tasks and the course administrator uses some.

INCA site administrators perform the tasks of site initialization and site maintenance from the administrator menu. The site initialization tasks include uploading of course, student, and approval code data from XML files into the system. This is performed once a semester. The site maintenance tasks include exporting data from the database into XML files and merging new grades data, available late during registration period, with the old data. Now, I will describe some of the course administrator scenarios.

1. **The course administrator enters into the system**. The administrator main page is presented to the course administrator upon successful login into the system. The main page allows the administrator to do tasks such as: edit the student and the semester course data; allocate the approval codes one at a time or in bulk; update the news to be displayed on student login page; enable or disable the approval code request and the auto-allocation of approval codes; and change their password. It also shows site status numbers such as: total number of outstanding requests; total number of approvals allocated; and total number of course seats remaining. The details of these numbers are presented in requests and approvals reports.

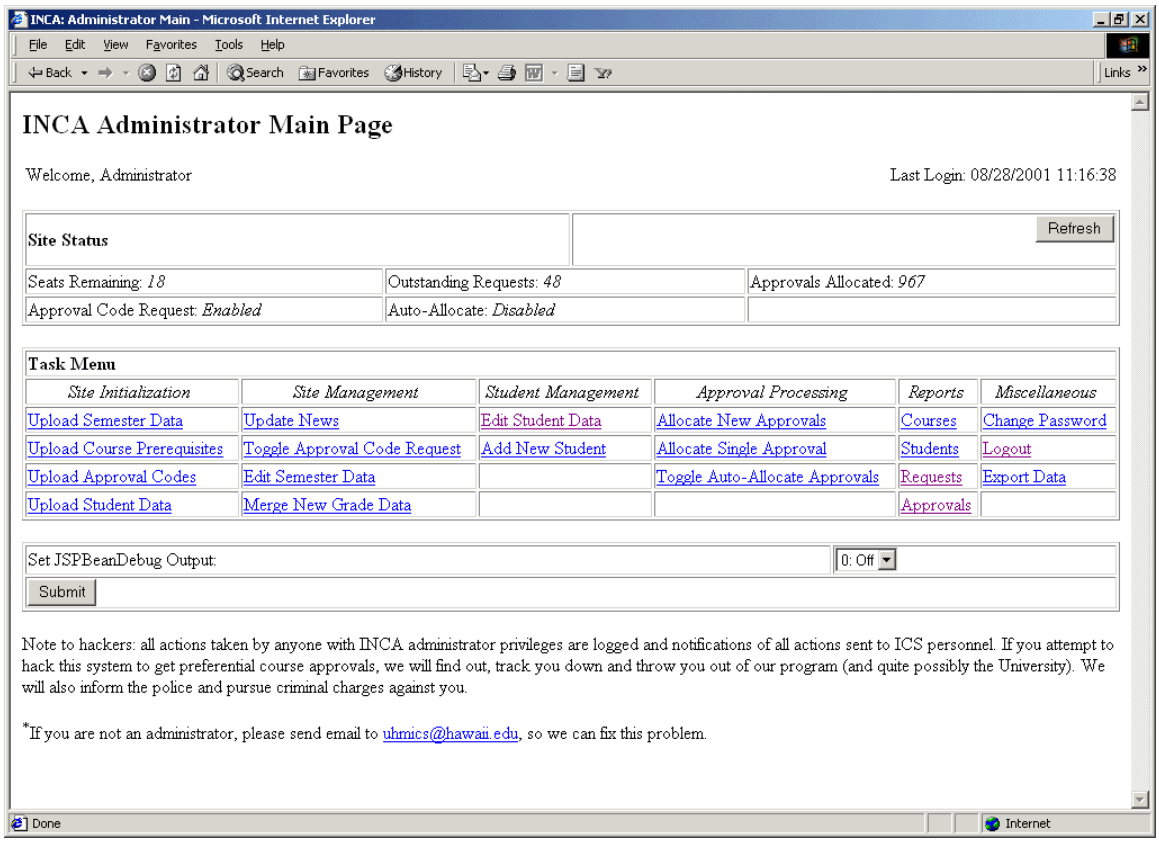


Figure 7 - Administrator main page

2. **The course administrator edits the student and semester records**. The edit student data page is divided into four sections: edit the basic status information of student; edit the ICS course history of student; edit the requests made by the student; and edit the consents for courses given to the student. The entire information related to the student is presented on edit student data page and a new edit page is presented for each section. In addition to editing the student records, course administrator can also add new students into the system. INCA allows the course administrator to add, delete, and modify the course sections information as some sections are added and removed late during the registration period.

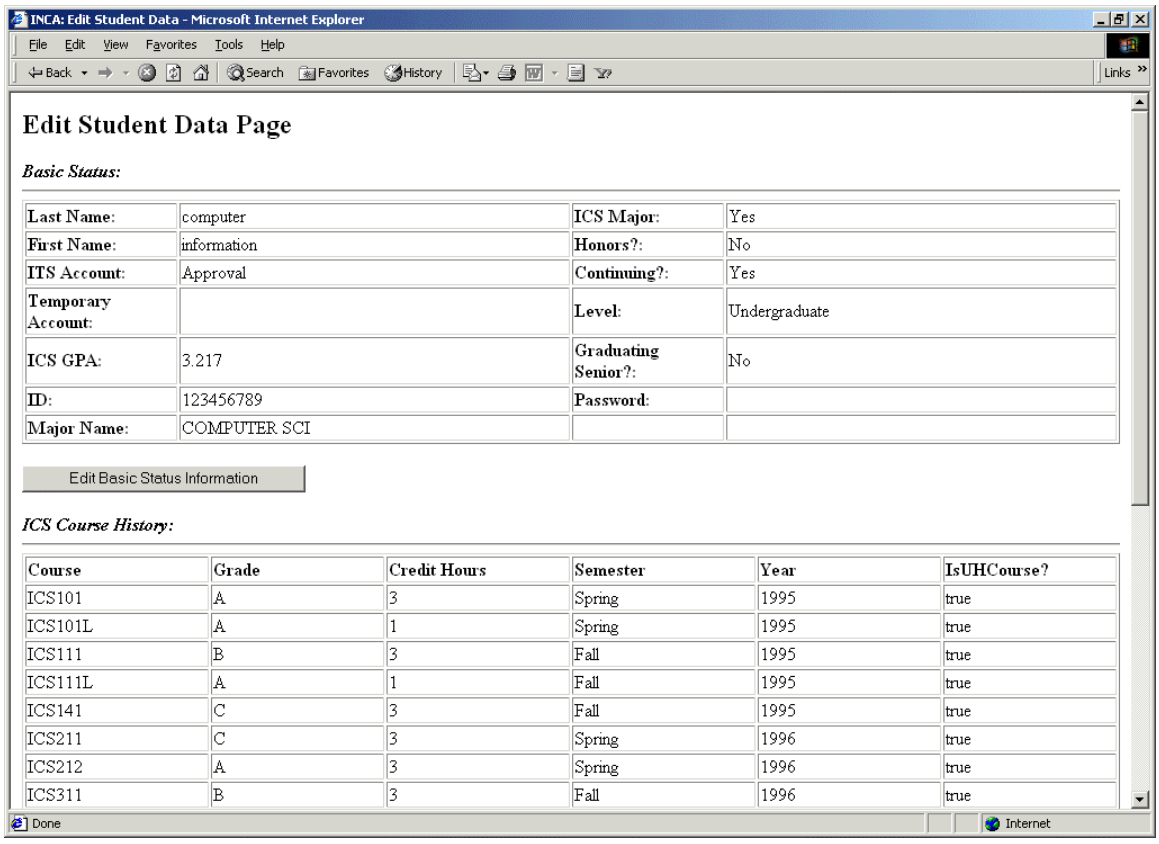


Figure 8 - Edit student data page

3. **The course administrator allocates the approval codes**. The course administrator can allocate the approval codes one at a time or in bulk. The course administrator needs to set an allocation policy in order to allocate the approval codes in bulk. The allocation policy lets the administrator to distribute a certain number of approval codes at a time. Once the administrator sets the policy, INCA presents the administrator with list of all the students: students that have been given the codes; students that haven't been given codes; and students that system is going to give the codes. The administrator reviews the list and makes final changes, if any. Then, INCA sends out automatic emails to all the students containing the approval codes.

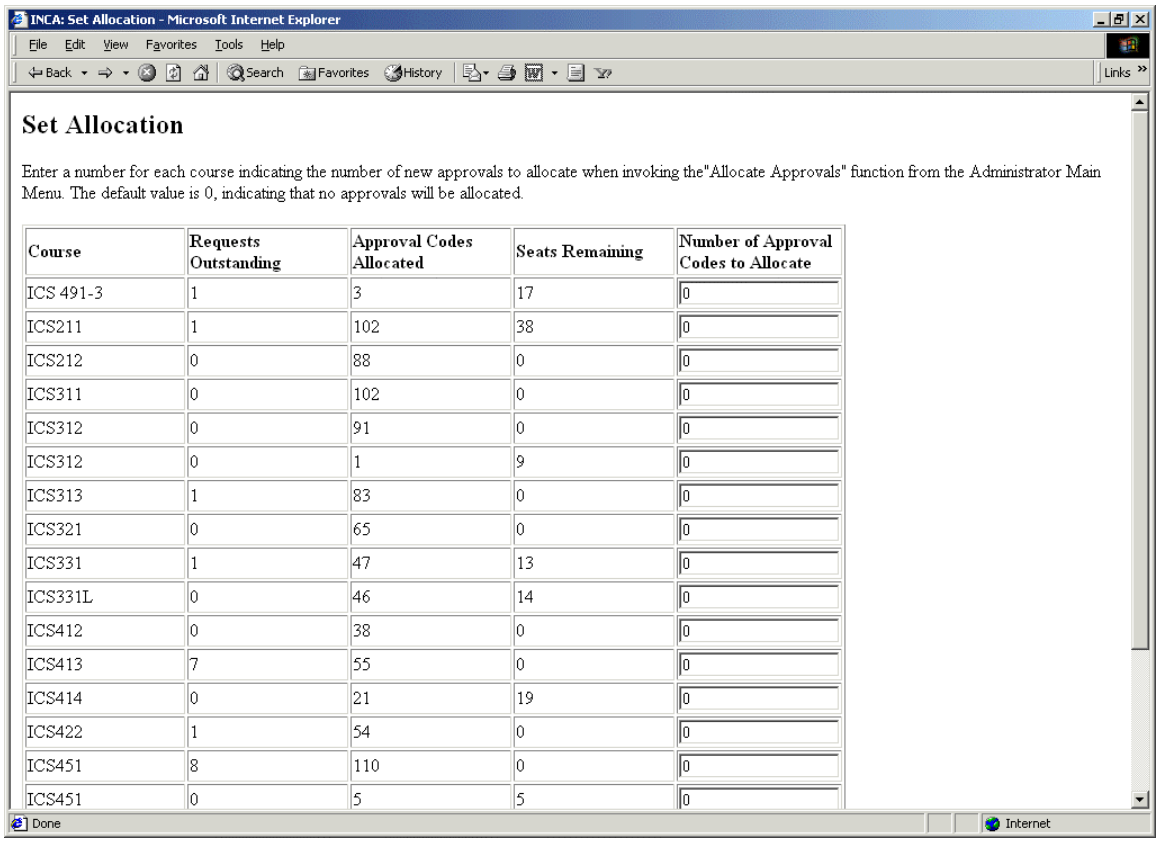


Figure 9 - Set allocation policy page

1.4 Research issues

INCA is a new system. As with every new system, many issues must be faced to make the system widely acceptable. The following table lists the most important research issues from six perspectives: administrator, student, department, designing allocation systems, commercial, and technological.

Table 1 - The research issues from different perspectives

Perspective	Research issues
Administrator	• Does INCA reduce the administrative overhead or does it introduces new kinds of problems?
Student	• Does INCA makes the approval code allocation process visible, predictable and understandable to the students?
Department	• What kinds of useful data INCA can provide to the departments? • In what ways, INCA is helpful to the faculty? • Does INCA really improve the quality of ICS program?
Designing allocation systems	• How should we devise the rules? • What weights should be assigned to each rule? • Should the ranking systems be transparent or opaque?
Commercial	• What are the other areas in which we can apply INCA allocation technology? • Can other departments across University of Hawaii use INCA? • Can other universities across US use INCA? Why or why not?
Technological	• What are the different technologies available for building multi-tier systems and how does technology used to build INCA compares with those technologies?

1.5 Thesis statement

INCA is a web-based approval code allocation system that automates the approval code allocation process during registration sessions in a university and benefits:

1. *Administrators, by decreasing administrative overhead.*
2. *Students, by making the allocation process predictable and understandable to them.*
3. *Departments, by providing them with valuable data that can be used for feedback and planning on their curriculum.*

The following table gives the operational definitions of these claims:

Table 2 - The operational definitions of thesis claims

Term	Operational definition
Administrative overhead	The reduction in administrative overhead means reduction in total time spend on reading and responding to student emails.
Predictability	A predictable allocation process allows students to predict their chances of getting into a course, based upon their ranking in the approval queue. If they don't see their chance of getting a course, they can look for other course options.
Understandability	An understandable allocation process makes the students understand the decisions behind the approval code allocation. Students grades, their major, their advancement into ICS program, and prerequisite and co-requisite course requirements are the major factors that govern the approval code allocation decisions.
Valuable data	INCA provides valuable data to the department in terms of the demand of the different courses.

In order to evaluate first claim, I will analyze the emails sent to "uhmics" account during Spring 2001 semester (when INCA was not used) and Fall 2001 semester (when INCA was used). During Spring 2001 semester, students used uhmics account to submit their course requests by email. An ICS administrator used to check all the emails and respond to them in a timely manner. During Fall 2001 semester, students used uhmics account to report problems with INCA. I will look into the nature of problems that occurred before and after INCA was in use. To evaluate second claim, I will conduct a user survey of students who used INCA. To evaluate third claim, I will investigate the data stored in INCA.

1.6 Structure of the proposal

In the second chapter, I will look at the related work of INCA. I will discuss different types of information systems at universities and how INCA fits into the big picture. Then, I will discuss different technologies used to build multi-tier systems and how technology used to build INCA compares with them. Finally, I will discuss different type of allocation systems and social issues involved in designing allocation systems.

In third chapter, I will discuss the history, design, and implementation of INCA. I will discuss high-level architecture and design of each of three tiers in detail.

In fourth chapter, I will show how I plan to evaluate INCA from the perspectives of students, administrator, and the department.

In fifth chapter, I will describe the sample results and how they relate to my hypotheses.

Finally, in sixth chapter, I will discuss conclusions and future directions.

Chapter 2. Related work

In this chapter, I will discuss the related work from three different perspectives. Firstly, I will describe information systems used at UH and how does INCA fits into the big picture. I will also discuss the information systems similar to INCA used at other universities. Then, I will discuss INCA from a technological point of view. I will describe different technologies used to build multi-tier systems and how does technology used to build INCA compares with those technologies. Finally, I will discuss different types of allocation systems. I will describe the concepts of procedural justice and productivity in education and how they are relevant to INCA design.

2.1 Information systems

Universities need to manage information to do their operations in a systematic manner. The information systems used at universities can be roughly classified into following three categories:

Table 3 - Information systems used at universities

System	Tasks
HRMS (Human Resource Management System)	Human resources, Payroll, Stock administration, Payroll interface, Pension administration
Financials	Asset management, Payables, Receivables, General ledger, Projects, Budgets, Purchasing, Inventory, Billing, Contracts
Student Information Systems	Recruiting and admissions, Student academic records, Academic advising, Student financials, Course enrollment

The universities use information systems to do their administrative tasks. These information systems differ from university to university because of factors like funds

availability, administration, faculty, and staff. Some universities keep all their data on old mainframe systems, whereas, others have adopted modern systems. The development language and hardware platforms differ. Some systems are centralized and some are web-based. Some universities buy solutions from vendors, whereas others build in-house solutions. The needs and situations differ from university to university and so are the types of information systems. In the next section, I will describe the information systems being used at UH and how INCA fits with them.

2.1.1 Information systems at UH

University of Hawaii uses 10-15 different kinds of information system to manage the administrative and student information. The following table describes the different kinds of information systems being used at UH. At the end of the table, I describe INCA with regard to other systems at UH. It is to give an overview of how INCA fits into big picture.

Table 4 - Information systems at UH

System	Description
CAPIS	Coordinated Admissions Program Information System. Produces reports related to admission activities.
DARS	Degree Audit Reporting System. Academic advising. Compares a student's academic work with the requirements of an institution's academic program and prepares a comprehensive report detailing student progress toward meeting those requirements.

FMIS	Financial Management Information System. Maintains Financial information for entire University of Hawaii. Sub-systems include General Ledger, Accounts Payable, Contracts and Grants, Budget Level Summary, Payroll Inquiry, Purchasing for OPRPRM and fiscal officer authorized purchase orders, Accounts Receivable, Departmental Checking and Fixed Assets.
ISIS	Integrated Student Information System. Maintains course catalog, schedule of classes, admissions, student biographic data, registration, and academic history.
HRMS	Human Resource Management System
PFIS	Physical Facilities Information System. Provides information on space utilities
SAIS	Student Aid Information System. Maintains and tracks financial aid information
SECE	Student Employment and Cooperative Education. Allows students to search jobs and employers to post jobs
SIMS	Student Information Management System. Provides student and course data for planning, policy making and decision support
SIS	Student Information System. Provides system-wide management information on student enrollment and activities.

INCA	INternet Course Allocation. Used for approval code allocation with the departments.

The following table shows the development languages, vendor and hardware platforms for the systems mentioned above. Some of the systems are developed in-house, whereas, others are bought from higher education system vendors. Most of the systems are developed in COBOL and run on university mainframe system. The detailed

information about these systems can be obtained from MIS ITS website [see MIS ITS website]. Again, at the end of the table, I describe INCA with regard to the information shown in the table.

Table 5 - Development and deployment details of UH information systems

System	Language	Vendor	Platform
CAPIS	Natural and Cobol	In-house	Runs of University's IBM Mainframe platform. It is a batch system run once for fall and spring semesters to produce managerial and operational reports relating to admission activities for decision making
DARS	COBOL	Purchased from Miami University	Purchased by College of Arts and Sciences in 1994. Runs on University's IBM mainframe platform. DARS reports are also accessible via a web browser
FMIS	Developed from Software AG's Financial Records System (FRS) software	In-house	Installed on University's IBM Mainframe
ISIS	COBOL, Software AG's Natural	Purchased from System and Computer Technology Corporation (SCT)	Runs on UHM's IBM mainframe platform
HRMS	PeopleSoft tools	PeopleSoft	Version 5.12 on IBM RS6000 Version 7.5 on Sun Solaris

PFIS	COBOL	Developed by ITS MIS	Runs on University's IBM mainframe platform
SAIS	Natural	In-house	Runs on University mainframe
SECE	Java	Joint project between UH Manoa SECE and MIS	Java based web application. Utilizes LDAP for user authentication
SIMS	Natural	In-house	Runs on University's IBM Mainframe
SIS	COBOL and Natural	In-house	Runs on University's IBM mainframe platform

INCA	Java, EJB, JSP, servlets, XML	Developed in CSDL	It is an n-tier system. Requires a web server, an application server and a database server. The web server should be capable of supporting servlets and JSP. The application server should be capable of supporting EJBs. Faster processor and more RAM will results in better INCA performance.

2.1.2 Course allocation at other universities

Almost every university may have different list of information systems from that of above. INCA is closely related to ISIS and PA'E. For the purposes of this research, we will look at the registration systems being used at other universities, how they handle the problem of "course allocation" and what are the systems they have similar to INCA.

I looked into registration systems used at more than 20 different universities. I selected these universities at random. Most of the universities use phone-based or web-

based registration systems. Most of them also have concept of restricted classes. However, They adopt different approaches for handling that problem, like, student consulting academic advisor and filling out some kind of forms to get into classes. Only one of the 20 universities, University of Indiana at Bloomington, has the concept similar to approval codes, which they call "registration access codes". But, they also don't have system similar to INCA and they use manual method of allocating the registration access codes. The following table lists the registration systems used at different universities and whether they have concept similar to approval codes or not.

Table 6 - Registration systems used at other universities

University	Phone registration	Web registration	Approval codes	Registration page
Alabama	Yes	TideWeb	-	registrar.ua.edu
Arizona	RSVP	Webreg	-	www.registrar.arizona.edu
California	URSA	URSA	-	ursa.ucla.edu
Georgia	-	OASIS	-	www.reg.uga.edu
Idaho	-	-	-	www.uidaho.edu/registrar
Illinois	Yes	Yes	-	www.online.uillinois.edu
Indiana	-	Regweb	Registration access codes	registrar.Indiana.edu
Kentucky	UK VIP	-	-	www.uky.edu/registrar
Michigan	-	Wolverinacess	-	www.umich.edu/~reg
New Jersey	RTTRS	Webreg	-	registrar.Rutgers.edu
North Carolina	-	-	-	regweb.uit.unc.edu
Ohio-state	-	-	-	www.ureg.ohio-state.edu
Oregon	Duck Call	Duck Web	-	registrar.uoregon.edu

South Carolina	-	VIP	-	argo.regs.sc.edu
Tennessee	Stripes	Tigerweb	-	enrollment.Memphis.edu
Texas	TEX	ROSE	-	www.texas.edu/student/registrar
Utah	-	-	-	www.saff.utah.edu/regist
West Virginia	STAR	-	-	www.arc.wvu.edu/star
Wisconsin	-	Yes	-	www.wisc.edu
Wyoming	STAR	Hole in the wall	-	siswww.uwyo.edu/reg

2.1.3 INCA

INCA is a web-based system that can be used in conjunction with the PA'E to register for class. PA'E is university-level system, i.e., it allows all the students inside a university to register for the classes in whatever department they want. INCA is a department-level system, i.e., it allows the students in a department to request for approval codes from the department so that they can register for the classes over PA'E.

2.2 n-tier database systems

INCA is an example of an n-tier system. The client, usually web browser, makes the requests to the web server in the presentation tier. If the request requires the processing not possible at the web tier, then, it is passed to the application server in the business tier. If the processing of request needs access/changes to data in the database, the application server contacts the database server in database tier. The results of

processing are communicated back from application server back to web server and further back to the client.

There are a number of technologies that can be used to build n-tier systems. J2EE, Microsoft .NET and CORBA/C++ are three major ones used to build n-tier systems. INCA is implemented using J2EE technology.

2.2.1 Overview of multi-tier systems

The following figure illustrates the concept of multi-tier systems. It shows how to construct a 2-tier, 3-tier and n-tier systems via 3 scenarios.

1. Client directly accesses the database. This happens in 2-tier systems.
2. Client or web browser makes request to presentation tier and presentation tier contacts database. Or Client directly accesses the application server and application server contacts the database. This happens in 3-tier systems.
3. Client or web browser makes request to presentation tier. It passes on the request to business tier that further passes request to database tier. This happens in an n-tier system.

The information is passed back and forth between the tiers and client.

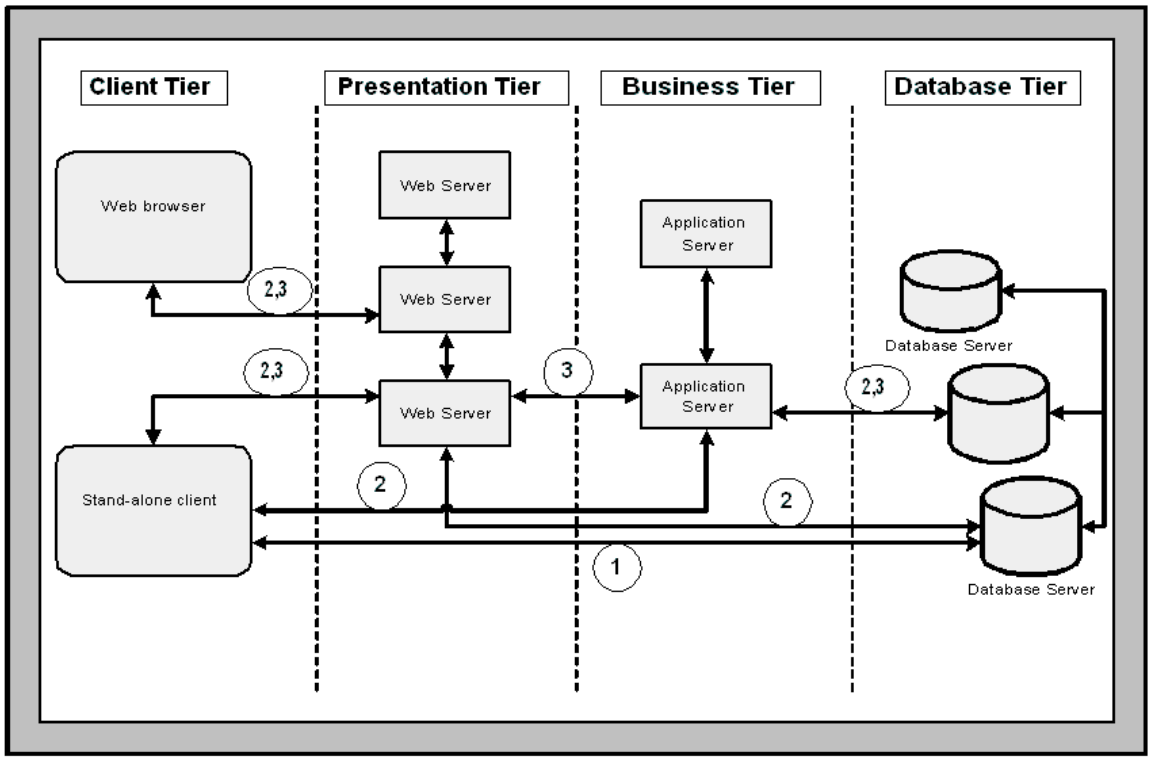


Figure 10 - Different scenarios in n-tier systems

2.2.2 J2EE

The Java 2 Platform Enterprise Edition (J2EE) was designed to simplify the development, deployment, and management of multi-tier enterprise solutions. J2EE is a set of specifications released by Sun Microsystems. It is not a product. Vendors like BEA, IBM and Oracle implement the J2EE specifications to build the enterprise products. In addition to the specification, Sun has also released a reference implementation of J2EE known as J2EE-RI.

The J2EE architecture is based on Java programming language. All J2EE components are written using Java programming language. These components are transformed into

Java byte codes and executed by a JRE at runtime. A J2EE application is hosted within a container, which provides qualities of service necessary for enterprise application, such as transaction, security and persistence services.

Java servlets and Java server pages are used for presentation in the presentation tier. They communicate with Enterprise Java Beans (EJB) in the business tier using RMI/IIOP. The EJBs communicate with the database in the database tier using JDBC.

Some of the popular products implementing J2EE specifications include Weblogic from BEA systems, Websphere from IBM, JRun from Allaire, and Jboss, an open-source application server. Currently, There are over 30 products that implement J2EE specification.

2.2.3 Microsoft .NET

Microsoft .NET is product suite that enables organizations to build smart, enterprise-class web services. Microsoft. NET is largely a rewrite of Windows DNA, which was Microsoft's previous platform for developing enterprise applications. Windows DNA includes many proven technologies that are in production today, including Microsoft Transaction Server (MTS) and COM+, Microsoft Message Queue (MSMQ), and the Microsoft SQL Server database. The new .NET Framework replaces these technologies, and includes a web services layer as well as improved language support.

Microsoft .NET architecture is based on Common Language Runtime (CLR), analogous to the JRE. .NET components can be written in any language like C++, CB, C# and even COBOL. All these get translated into Microsoft Intermediate Language

(MSIL), which is analogous to Java byte codes. The IL code is interpreted and translated into a native executable using CLR.

ASP.NET is used for presentation in the presentation tier. The ASP pages communicate with COM+ components in the business tier using DCOM or SOAP. The COM+ components communicate with the database in the database tier using ADO.NET.

.NET platform includes following .NET enterprise servers: SQL server 2000, Exchange server 2000, Commerce server 2000, Application center server 2000, Host Integration server 2000, Internet Security and Acceleration Server 2000 and BizTalk Server 2000.

2.2.4 A comparison of technologies

There are different factors that can be used to compare technologies used for building multi-tier systems. These include time-to-market, vendor support, legacy systems support, platform maturity, languages and web services support, and portability. For purposes of this research, I am comparing J2EE and .NET in terms of n-tier technology infrastructure support. The following figure draws the comparison.

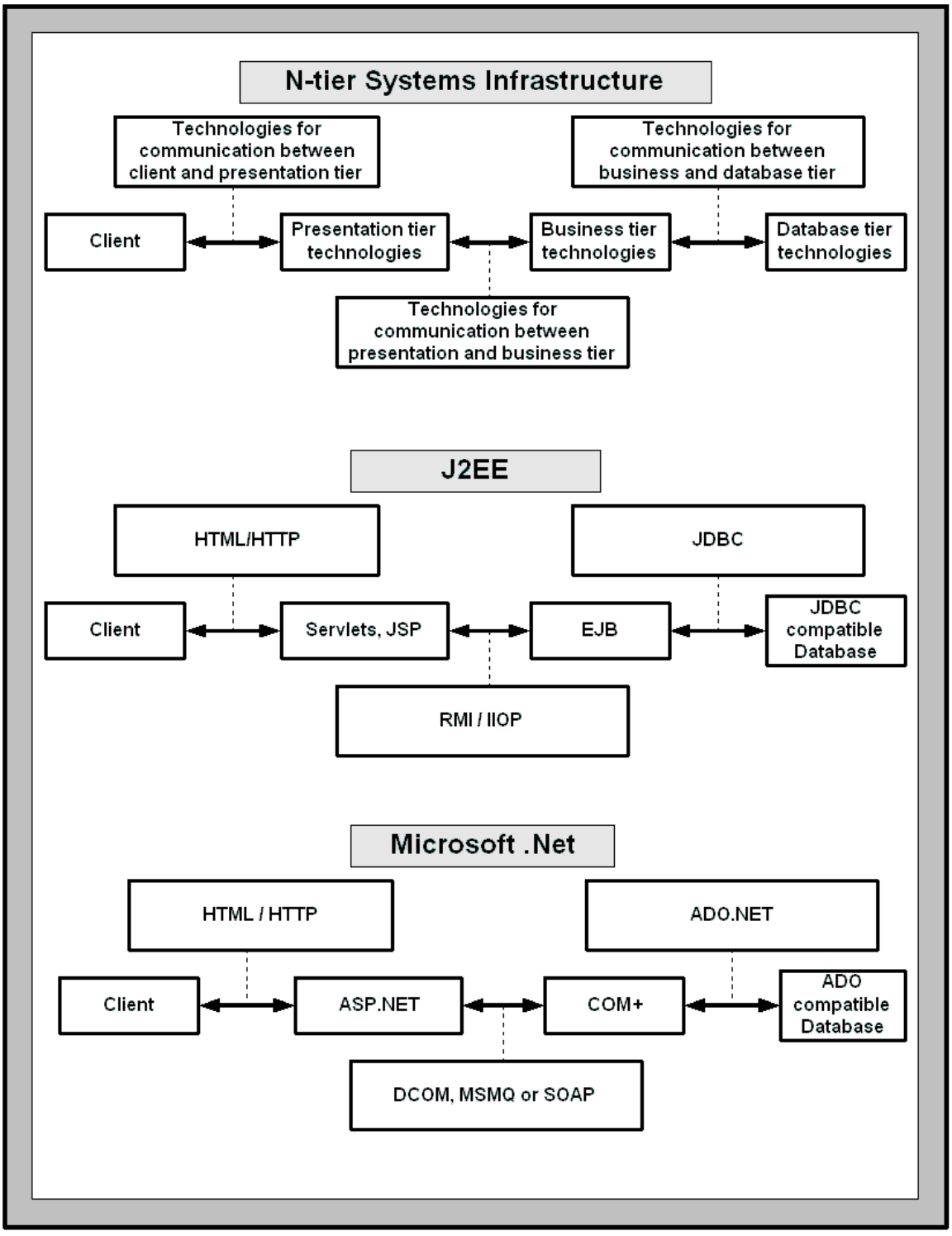


Figure 11 - Comparing J2EE and Microsoft .NET

2.2.5 INCA

INCA is implemented using J2EE technology. INCA was run on J2EE-RI (Java 2 Enterprise Edition, Reference Implementation) from Sun Microsystems, for fall 2001 semester. J2EE-RI packages web server, application server and database server into a single product. The web server is Jakarta Tomcat and the application server is from Sun Microsystems. The database server is cloudscape. INCA was also successfully deployed on other application servers. This includes the BEA weblogic that has web server and application server integrated into a single product. INCA was also deployed on the combination of "resin" web server, "Jboss" application server and "cloudscape" database server. This is one of the beauties of J2EE technology that you can choose application server, web server and database server of your choice as long as they implement J2EE specification.

2.3 Rule-based prioritization systems

INCA is a rule-based Prioritization System that applies certain rules and prioritizes the students and fulfills the requests of top-ranked students. Prior to INCA, the approval code distribution scheme was First Come First Serve (FCFS). Adopting a new technology in a organization has its own social implications. So, We need to evaluate INCA before it can be further used in the departments.

2.3.1 Allocation tasks and their classification

Elster [1992] has done useful work on allocation systems. He has examined a large variety of non-market institutions where scarce goods are allocated. Some examples of the allocation tasks considered are:

- Who gets a kidney for transplantation?
- Who is admitted to selective colleges?
- Who is selected for layoffs?
- Who is chosen for military service in time of war?
- Who is allowed to adopt children?
- Who is allowed to immigrate?

Elster classifies allocation tasks as follows:

- Selection – It involves ranking individuals and allocating the scarce good from the top to bottom, until it is exhausted.
- Admission – It involves comparing individuals against some threshold, and allocating the good to all who pass.
- Placement – It involves matching individuals to heterogeneous units of good.

Some of the criterions that play a role in decisions are as follows:

- Merit (as in selection to receive honors)
- Need (as in selection for space in intensive care units)
- Seniority (as in layoffs in unionized firms)
- Notions of equity.

Other examples of allocation systems include credit-card request processing systems. The systems apply certain rules before the customer's request for credit card is approved.

2.3.2 Procedural justice

"Procedural justice" refers to a research literature that examines how perceptions of the means (or procedures) through which decisions are made in organizations affect people's satisfaction with or attitude towards these organizations. This is a relatively large literature, but for our purposes, can be boiled down to this:

"The fairness of a firm's procedures may have a greater impact on organizational commitment than the fairness of personal outcomes that workers receive, perhaps because procedures define the organization's capacity to treat employees fairly. Thus, if they see procedures as fair, employees may view the organization positively, even if they are currently dissatisfied with such personal outcomes."

The implication is that if there is a visible (i.e. clearly articulated) procedure which is viewed as fair, then, even if someone is unhappy with his/her individual outcome under the procedure, he/she will probably still view the organization positively.

2.3.3 Productivity in education

Hartman and Boyd have written a simple introduction to the issue of productivity in education from diverse (economic, political and sociological) perspectives. They conclude that improvement in the resource allocation practices in education requires a shift in focus to results instead of inputs, a strong emphasis on student learning as the primary focus of decisions, and systematic evaluation of results.

2.3.4 Evaluation technologies

A lot of work has been done in the field of evaluation of technologies from a social standpoint. Tudd et al. discusses the research methods in social relations.

2.3.5 INCA

INCA is a selective allocation system. Approval codes used to be given to the students on a first come first serve basis, after doing an eligibility check, prior to INCA. With INCA, a ranking list is produced by giving points to the students based on various rules and then selecting the students from the list. Some students do complain about the selection policy implemented by INCA. So, INCA needs to be evaluated to determine whether it really improves the quality of ICS program. Also, what is the response of most of the students about INCA?

Chapter 3. INCA design and implementation

INCA is an n-tier database system developed using J2EE technology. INCA has 3 server-side tiers, namely, Web tier, EJB tier and EIS tier. The client requests are received at the Web tier. The Web tier serves pages to the client by combining together formatting information in the JSP page with the dynamic content provided from the EJB tier. The EJB tier consists of EJBs that implement the business logic of the application. The EJB tier receives the requests from the Web tier, applies the business logics, interacts with the persistent storage in the EIS tier and finally communicates the results back to the Web tier. The EIS tier contains a relational database system that contains a set of tables that implement the persistent storage for the information in INCA.

3.1 History of INCA

It was the beginning of fall 2000 semester when Professor Philip Johnson from ICS department started thinking about an automated system to solve the problem of manual approval code distribution. He started the project called “Courseapp” to build such an automated system while teaching his class ICS613, Advanced Software Engineering, during the Fall 2000 semester. He was teaching students, software engineering along with Java 2 Enterprise Edition (J2EE) technology and as a part of teaching that he designed the initial version of Courseapp. Around the end of the semester, 20 students started coding the system and they coded it for around two weeks. When the semester got over, the system was not even close to functional. Then, Six students under the guidance of Prof. Philip Johnson proceeded to complete the system so that it could be used for Fall2001 semester and system was renamed INCA. After one month, the team was

reduced to four. The remaining four students along with Professor Philip Johnson worked day and night for around four months and the system was first used by the Information and Computer Sciences department on April 30, 2001 for the approval code distribution of Fall 2001 semester. I was among the four students who worked on the system. I worked along with David Liang, Paula Nishida and Weifeng Miao. At the time of writing this thesis, only two students, weifeng miao and myself are maintaining this huge system consisting of more than 50,000 lines of Java code and 400 files.

3.2 High level architecture

< I will write this in detail later >

3.3 User interaction cycle in detail

< I will write this in detail later >

3.4 The web tier design

< I will write this in detail later >

3.5 The EJB tier design

< I will write this in detail later >

3.6 The database tier design

< I will write this in detail later >

Chapter 4. Evaluation of INCA

INCA was used for the first time during Fall 2001 semester registration to do approval code allocation. The manual method of approval code allocation was used in the previous semester, Spring 2001. The same ICS course administrator, who distributed approval codes manually during Spring 2001 semester, used INCA to allocate approval codes for Fall 2001 semester. The course administrator seemed happy and satisfied with the system and often commented that INCA is very fast, easy-to-use and performing very well. I heard her so many times saying that "It's great". However, the evaluation of INCA needs more than that.

I will evaluate INCA from the perspectives of the course administrators, the students and the departments. To support my hypothesis that INCA reduces the administrative overhead involved in the process of approval code allocation, I will look into the emails sent to the "uhmics" account during Spring 2001 and Fall 2001 semesters. I will conduct a user survey to support the hypothesis that INCA improves the predictability and understandability of students. The operational definitions of predictability and understandability are stated in chapter One. To support my hypothesis that INCA provides valuable data to the departments that can be used for feedback and planning on their curriculum, I will analyze the INCA database and come up with the information useful to departments for planning on their curriculum. In the following sections, I will look into administrator, student, and department evaluation in detail. The term administrator refers to course administrator in the following sections.

4.1 Administrator evaluation

The primary goals of administrator evaluation are to show how INCA reduces the administrative overhead and what new kinds of problems it presents to the administrator. In order to do administrator evaluation, I will do an analysis of emails received at "uhmics" account during Spring 2001 and Fall 2001 semester. Students used uhmics account to email their course requests during Spring 2001 semester and to report the questions and problems with INCA during Fall 2001 semester. Students notified administrator to correct their records in case the student information listed in INCA was incorrect.

4.1.1 Email Analysis

The email analysis will help in validating the hypothesis regarding administrators. According to the hypothesis, INCA should reduce the administrative overhead involved in reading and responding to the emails. Another goal of email analysis is to look into student problems and improve INCA to minimize occurrence of those problems.

During Spring 2001, 497 students requested the approval codes. So, total number of expected emails was 497 with one email containing the course request from each student. However, a total of 1537 emails were received. For Fall 2001 semester, 429 different students emailed uhmics account and a total of 1316 emails were received. The expected and actual numbers of emails differ by a large amount. I will analyze all the emails in order to prove that INCA reduces the administrative overhead. I will look at the nature of problems before and after INCA. The different tasks in email analysis are as follows:

- Classify emails into different categories and see how the patterns of problems change
- Identify the student problems before and after INCA was used.
- Identify newer kind of problems that come with INCA

4.1.2 Methodology

All of the emails, sent to uhmics account since 14 November 2000, are residing on the UH UNIX servers. The administrator used to download emails on her personal computer, but kept a copy of all the emails on UH UNIX systems. These emails are source of the email data analysis. I also downloaded the emails from UHUNIX system to my personal computer using Netscape Messenger email client. Netscape Messenger stores the emails in a proprietary file format and lets the users to do email operations via client window.

I analyzed that Netscape file format and wrote a Netscape email file parser. For each message, I extracted the following information from the Netscape file: message number, date and time on which message was sent, and sender information (I assigned an ID to each sender). I imported the information in a Microsoft Excel spreadsheet. I started reading the emails from Netscape Messenger one-by-one. I categorized every email and then recorded the information in the Excel spreadsheet. At the end, I summed up each of the categories. We can see the sample of analysis being performed in the following table. The portion of Excel spreadsheet shows the message ID, date and time at which message was sent, sender ID, and the different problem categories. The table doesn't list the summary of the emails.

Table 7 - Sample email data analysis

M. ID	Date	S. ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	Mon, 30 Apr 2001 04:50:00 -1000	8	1													
9	Mon, 30 Apr 2001 05:47:51 -1000	9			1	1										
10	Mon, 30 Apr 2001 08:08:49 -1000	10			1		1									
11	Mon, 30 Apr 2001 08:51:45 -1000	11						1								
12	Mon, 30 Apr 2001 09:56:14 -1000	12							1							
13	Mon, 30 Apr 2001 10:00:53 -1000	13		1												
14	Mon, 30 Apr 2001 11:04:55 -1000	14							1							
15	Mon, 30 Apr 2001 11:32:38 -1000	15							1							
16	Mon, 30 Apr 2001 11:41:24 -1000	16							1							
17	Mon, 30 Apr 2001 11:53:47 -1000	17						1								
18	Mon, 30 Apr 2001 12:34:12 -1000	18								1						
19	Mon, 30 Apr 2001 12:36:05 -1000	18									1					
20	Mon, 30 Apr 2001 12:37:45 -1000	19							1							
21	Mon, 30 Apr 2001 12:46:00 -1000	20							1							
22	Mon, 30 Apr 2001 12:47:17 -1000	21		1			1					1				
23	Mon, 30 Apr 2001 12:51:03 -1000	22						1								
24	Mon, 30 Apr 2001 12:55:42 -1000	23						1								
25	Mon, 30 Apr 2001 12:56:29 -1000	24						1								
26	Mon, 30 Apr 2001 13:03:45 -1000	25							1							
27	Mon, 30 Apr 2001 13:17:43 -1000	26											1			
28	Mon, 30 Apr 2001 13:18:54 -1000	14														
29	Mon, 30 Apr 2001 13:28:15 -1000	27							1							
30	Mon, 30 Apr 2001 13:34:30 -1000	28							1							
31	Mon, 30 Apr 2001 13:35:18 -1000	29						1								
32	Mon, 30 Apr 2001 13:50:08 -1000	30								1						
33	Mon, 30 Apr 2001 13:52:35 -1000	23				1								1		
34	Mon, 30 Apr 2001 13:58:35 -1000	19							1							
35	Mon, 30 Apr 2001 14:05:11 -1000	31													1	
36	Mon, 30 Apr 2001 14:09:55 -1000	32														1
37	Mon, 30 Apr 2001 14:33:37 -1000	33														1
38	Mon, 30 Apr 2001 14:36:59 -1000	34							1							
39	Mon, 30 Apr 2001 14:48:53 -1000	35						1								

I kept the problem definitions in a separate table. For every email read, I identified the problem and put a "1" against the email and problem category in the Excel spreadsheet. At the end, I summed up all the numbers. The following table shows a sample of problem definitions. I kept revising the problem categories as I proceeded by introducing new categories and combining some of the categories.

Table 8 - Fall 2001 problem categories

1. Student doesn't know INCA URL
2. Student is requesting summer 2001 code
3. Student is requesting fall 2001 code by email
4. Student has a concern about graduation in fall 2001 semester
5. Student is not sure of the procedure for requesting approval codes
6. Basic info needs correction. Possibilities may be: Grades not in database, Change to the ICS major, Enter grade for equivalents, Enter transfer grades, change graduating senior
7. INCA registration problem
8. INCA password doesn't work
9. The message to ignore a previously sent message
10. A student with different major
11. When grade change of spring 2001 will be reflected in INCA
12. Priority question
13. Summer grades problem
14. INCA bug 1

This kind of email analysis will help me in finding following numbers and proving the hypothesis of administrators:

- What are different types of emails?
- What are the most common emails?
- Which problems existed before INCA was used?
- Which problems got vanished with INCA?
- What new kinds of problems got introduced?
- What is the time period during which most of the emails were sent?

4.1.3 Limitations

The emails received at uhmics don't describe all of the problems. If a student never emailed a problem or directly approached the department or course administrator to discuss the problem, then, that is not covered.

4.2 Student evaluation

The primary goal of student evaluation is to get the thoughts and feelings of students about INCA. I will conduct a user survey in order to do student evaluation. Currently, with INCA:

- Students can predict their chances of getting a course. INCA provides them with real-time information on total seats, total requests that have been made, and their ranking in the queue. Students can see their chances for getting a course based upon their ranking. There was no such concept as *predictability* before INCA.
- Students can get a good understanding of course qualifications. They can check pre-requisites, co-requisites, and concurrent course requirements. They can check why or why not they are qualified for a certain course. INCA is designed to improve the *understandability* of course requirements. Without INCA, manual explanations were needed.
- Students can clearly see the departmental policies in terms of approval code allocation. INCA is designed to improve this kind of *visibility*. The departments may favor students with good academic credentials or majors and advanced students of the department. Departments may not favor repeating attempts and

withdraws. INCA enforces the departmental policies and makes them visible to the students.

4.2.1 User survey

The goals of INCA user survey are to get the feedback from students regarding the improvements in INCA in terms of ease-of-use, INCA selection policy, and desired new features. The user survey will also help me in validating my hypothesis regarding predictability and understandability of approval code allocation process. The INCA user survey is presented in appendix A.

4.2.2 Methodology

I will conduct a web based user survey. The survey will be anonymous and participation in the survey will be voluntary. Students will not given any credit for participating in the survey.

I will post the survey over the web and ask all the students to fill out the survey by visiting the web-link. The web-link will be specified in the email sent to the student mailing list. All of the students are members of uhm-ics-undergrads and uhm-ics-grads mailing list.

Each time a student will fill out the survey, an automatic email will be sent to me containing the answers. I will look for the student responses for one week since the survey will be posted.

4.2.3 Limitations

The user survey was conducted after 3 months from the beginning of INCA operation. The survey should have been after 1 month of operation.

4.3 Department evaluation

The primary goal of the department evaluation is to come up with the data valuable to the departments for course and sections planning. I plan to look into the data residing on INCA server to see what information might be useful for the departments and how it can be presented in an elegant manner.

4.3.1 INCA database analysis

The INCA database analysis will help me in validating my hypothesis that INCA provides valuable data to the departments that can be used on feedback and planning on their curriculum.

The INCA database stores the data regarding students, their grades, their requests and the approval codes granted to them. It also maintains the course scheduling information along with prerequisites and co-requisite requirements for different courses.

4.3.2 Methodology

I will write SQL scripts to extract different types of information.

Chapter 5. Research Schedule

The thesis proposal describes the main ideas of the research. The thesis proposal contains almost final versions of chapters one, two, and four. The dates during which I plan to perform the evaluation and finish writing rest of the chapters are listed in the following table.

Table 9 – Research Schedule

Task	Dates
Finishing of the thesis proposal	24 September 2001
Presenting thesis proposal to the thesis committee and getting their feedback	25 - 26 September 2001
Finish writing INCA design and implementation chapter	24 - 30 September 2001
Conducting the user survey to get the feedback of students	28 - 5 October 2001
Analyzing the results of user surveys	6 - 10 October 2001
Analyzing the INCA database	6 - 10 October 2001
Analyzing the uhmics emails	25 September – 10 October 2001
Finish writing the Results chapter	11 - 15 October 2001
Finish writing the Conclusions chapter	11 - 15 October 2001
Present the final draft to the chair-person	16 October 2001
Presenting final draft to committee	20 October 2001
Thesis defense	26 October 2001

Appendix A. INCA user survey

As part of my M.S. thesis research, I am performing an evaluation of the INCA system in order to support future improvements. Please support this effort by taking a few minutes to answer the following six questions. Participation in this research is voluntary, no information about your identity will be collected or used in the results, and you will receive no course credit for participation. For information about the results of this research, please contact me, Jitender Miglani, at jitender@hawaii.edu.

1. Please rank the overall quality of INCA in terms of ease-of-use.

- ☐ Excellent
- ☐ Very Good
- ☐ Good
- ☐ Average
- ☐ Poor

2. INCA provides real-time information on total seats, total requests that have been made, and your ranking in the queue. This is to help you in predicting the chances of getting the approval codes. To what extent was this useful feature?

- ☐ It was extremely useful. Now I can predict my chances of getting the approval codes.
- ☐ It was somewhat useful. It is nice to know your rank in different courses, but I think I can get the courses I want without it.
- ☐ It was not useful at all.
- ☐ No opinion. I don't care about it.

3. Many ICS courses have prerequisites, co-requisites, and concurrent course requirements. INCA tells you why or why not you are qualified for a certain course. To what extent was this effective in improving your understanding of different course qualifications?

☐ It was extremely effective. Now I know why or why not I am qualified for certain courses.

☐ It was somewhat effective. I already knew course qualifications, but it is nice to know them through INCA.

☐ It was not effective at all.

☐ No opinion. I don't care about it.

4. INCA provides the "priority points" mechanism that enables students with a strong academic background to have higher priority for their future courses. To what extent do you support this approach to course allocation?

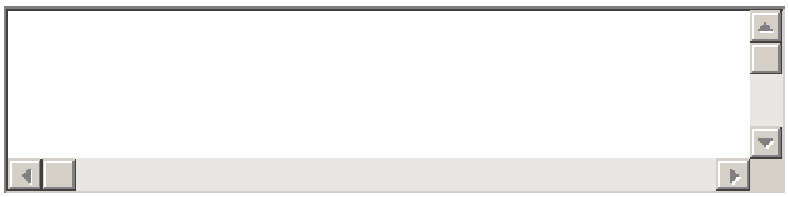
☐ I strongly support it. It should definitely be used next semester.

☐ I prefer it. But it doesn't matter if it is not used for next semester.

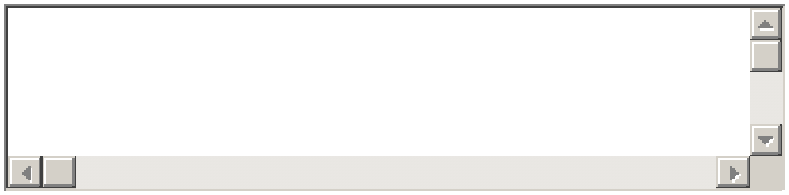
☐ I don't think it should be used.

☐ No opinion. I don't care.

INCA prioritizes the students using different rules such as: favor advanced students; favor students with high ICS GPA; favor first and second choices of students; and do not favor repeat attempts. Do you have suggestions for new point rules or changes to the current priority rules? Should point values be changed?



5. If you would like, please tell us about the problems you faced or give some suggestions for improvements in INCA.



6. Do you prefer INCA to the prior method of sending emails to uhmics?

- ☐ Yes, I prefer INCA
- ☐ No opinion, I don't care
- ☐ No, I would prefer sending emails to uhmics@hawaii.edu to request the courses

Bibliography

- Chad Vawter and Ed Roman. *J2EE vs. Microsoft .NET, A comparison of building XML-based web services.* The Middleware company. June 2001
- Charles M. Tudd, Eliot R. Smith, and Louise H. Kidder. *Research methods in social relations*. Harcourt Brace Jovanovich College Publishers. Sixth Edition
- Duane K. Fields, Mark A. Kolb. *Web development with Java Server Pages.* Manning publications, 2000
- Jon Elster. *Local Justice, How institutions allocate scarce goods and necessary burdens.* Russell Sage Foundation. 1992
- Lind E. and Taylor T. *The social psychology of procedural justice*. 1988
- Murray Laver. *Computers and social change*. Cambridge University Press. 1980
- Nicholas Kassem and the Enterprise team. *Designing Enterprise Application with the Java 2 Platform, Enterprise Edition.* Addison Wesley. 2000
- Roger Sessions. *Java 2 Enterprise Edition (J2EE) versus The .NET Platform. Two Visions for eBusiness*. ObjectWatch Inc. March 2001
- William T. Hartman and William Lowe Boyd. *Resource allocation and productivity in education. Theory and Practice.*
- Web - *educause website* <http://www.educause.edu/ >
- Web - *INCA Software requirement specifications* <http://csdl.ics.hawaii.edu/techreports/01-09/01-09.html>
- Web - *ITPapers website* <http://www.itpapers.com/ >
- Web – *MIS ITS website* <http://mis.its.hawaii.edu/projects.html >