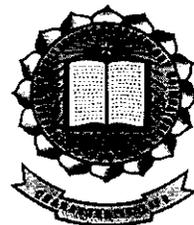




P-3105



EXPERIENCE BASED APPROACH TO SCHEDULING PROBLEMS WITH THE LEARNING EFFECT

A PROJECT REPORT

Submitted by

S.DASS

71206104009

K.B.PRABU

71206104033

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

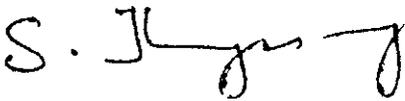
ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2010

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**EXPERIENCE BASED APPROACH TO SCHEDULING PROBLEMS WITH THE LEARNING EFFECT**” is the bonafide work of “**K.B.PRABU, S.DASS**” who carried out the project work under my supervision.



SIGNATURE

Prof. S.THANGASAMY, Ph.D

HEAD OF THE DEPARTMENT

Computer Science and Engineering
Department

Kumaraguru College of Technology,
Coimbatore-641 006.



SIGNATURE

Mr.V. SUBRAMANI M.Tech.

SUPERVISOR

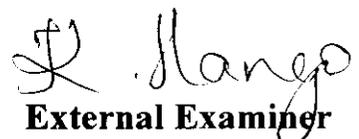
Associate Professor/CSE

Kumaraguru College of Technology,
Coimbatore- 641 006.

Submitted for the University Examination held on 15-04-10



Internal Examiner



External Examiner

DECLARATION

We

S.DASS

71206104009

K.B.PRABU

71206104033

Hereby declare that the project entitled”**EXPERIENCE BASED APPROACH TO SCHEDULING PROBLEMS WITH THE LEARNING EFFECT**” is a record of original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any Institutions, for fulfillment of the requirement of the course study. The report is submitted in partial fulfillment for the award of the degree of bachelor of computer science and engineering of Anna University, Chennai.

Place: Coimbatore

Date: 15-04-10



[S.DASS]



[K.B.PRABU]

ACKNOWLEDGEMENT

We extend our sincere thanks to our principal Dr. Ramachandran, Kumaraguru College of Technology, Coimbatore for being a constant source of inspiration and providing us with necessary facility to work on this project.

We would like to make a special acknowledgement and thanks to **Dr.S.THANGASAMY, Ph.D.**, Dean, Professor and Head of Department of Computer Science and Engineering, for his support and encouragement throughout the project.

We express deep gratitude and gratefulness to our guide, **Mr.V. SUBRAMANI M.Tech.**, Department of Computer Science and Engineering for his supervision, enduring patience, active involvement and guidance. We would also like to thank our project coordinator, **Mrs. P.DEVAKI M.S.**, Assistant Professor, Department of Computer Science and Engineering, for her support during the course of our project.

We would like to convey our honest thanks to Mr.V. Subramani for his consistent guidance throughout the project. We also thank all members of staff of the department and wealth of experience from which we greatly benefited.

ABSTRACT

Many large Medical centers and Hospitals stand to gain to operational efficiency and smart hospitality by effectively scheduling Patients. Early considerations on learning and its impact on the behavior of living beings revealed that the time required to perform a task (job) decreases as experience related to it increases. These psychology studies exposed the existence of the phenomenon called “learning effect” in the case of “learning-able” objects, i.e., such that living beings can increase their skills and efficiency. The learning effect has attracted particular attention in Medical industry and has perceived as worthwhile to be taken into consideration, during medical attention planning as it can affect general hospitality efficiency. The existence of the learning effect in many hospital systems like patient scheduling etc are undoubted; thus, it is worthwhile that it be taken into consideration during scheduling of patients and planning to increase hospitality efficiency.

Thus, a methodology that allows for modeling this effect and, on this basis, arranges efficient patient schedules is of a great importance in modern management. It reveals that most of the learning models in scheduling are based on the learning curve introduced by Wright. However, further study about learning itself pointed out that the curve may be an “S”-shaped function, which has not been considered in the scheduling domain. To fill this gap, we analyze a scheduling problem with a new experience-based learning model, where Patient scheduling processing times are described by “S”-shaped functions that are dependent on the experience of the processor, problems with other experience-based learning models are also taken into consideration.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1.	INTRODUCTION	1
2.	HARDWARE AND SOFTWARE REQUIREMENTS	3
3.	LITERATURE SURVEY	4
	3.1 LEARNING	5
	3.2 SCALE EFFECTS	6
	3.3 FORGETTING	6
	3.4 METHODOLOGY	8
4.	EMPIRICAL RESULTS	12
5.	SYSTEM DESIGN	15
	5.1 EXISTING SYSTEM	16
	5.2 PROPOSED SYSTEM	18
6	FEASIBILITY	19
7	DESIGN AND DEVELOPMENT	22
8	SOFTWARE SPECIFICATIONS	25
9	BLOCK DIAGRAM	29
10	TESTING	30
11	IMPLEMENTATION	37
12	SOURCE CODE	39
13	SCREENSHOTS	55
14	CONCLUSION	63
15	REFERENCES	65



INTRODUCTION:

PROBLEM DESCRIPTION

In hospitals when patient comes for a test the required equipment and other resources has to be arranged. The time required for the arrangement is called changeover time. It comprises of setup time and removal time. Setup time is the time span required to prepare machine for processing a job. The removal time is the time span needed to restore initial state of machine .In older patient scheduling systems this changeover is included with in processing time itself. In the proposed system the changeover time and processing time is considered separately. So the learning effect can be applied to reduce both setup time and processing time. When a patient arrives the Common Agent (CA) determines the tasks that have to take, which consists of consulting doctor, conduct tests etc.The patient agent then request for the resources .Each Resources Agent (RS) contains Learning Agent (LE) and they schedule the job according to the experience they possessed. The result is a PS (Patient Schedule) for each PA and a RS (Resource Schedule) for each resource.

HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

Processor	:	Pentium IV
Speed	:	Above 500 MHz
RAM capacity	:	2 GB
Floppy disk drive	:	1.44 MB
Hard disk drive	:	200 GB
Key Board	:	Samsung 108 keys
Mouse	:	Logitech
CD Writer	:	52x LG
Printer	:	DeskJet HP
Motherboard	:	Intel
Cabinet	:	ATX
Monitor	:	17" Samsung

SOFTWARE CONFIGURATION

Operating System	:	Windows XP and above
Front end used	:	JAVA
Back End	:	SQL Server

LITERATURE SURVEY

LITERATURE SURVEY:

SOURCES OF EXPERIENCE

Given that there is evidence to support the existence of learning curves as good descriptors of how learning occurs, the question arises as to why this effect appears. The answer appears to be that it has three major sources: learning, technological advances, and scale effects.

LEARNING

The efficiency of all aspects of labors input (production, maintenance, supervision etc.) increases as the people involved learn or gain experience in the activity in question. Learning also involves the discovery of better ways

- to organize work through improved methods and
- work specialization or
- getting better performance from production equipment as personnel become well acquainted with it.

TECHNOLOGICAL IMPROVEMENTS

New production processes, especially in capital-intensive industries, often contribute substantial economies to the learning effect.

Beevan noted as an example, Golden Wonder's introduction of

batch-frying mode. Changes in the resource mix, such as the replacement of labour by automation, also provide a technology-driven basis for the learning effect. Process and product changes that improve yield are yet another source. Product standardization and redesign has contributed to changes in learning rates also.

SCALE EFFECTS

Scale economies at the plant level result from savings on capital costs, an increased potential for division of labour, and better utilization of resources. Hollander, in a study of the sources of efficiency increases at DuPont rayon plants, concluded that only 10 to 15 percent of the efficiency gains were attributable to scale effects.

FORGETTING

Just as learning increases with experience, memory loss appears to increase as a function of a break from task performance. Steedman suggested that the residual memory of the participants after a break, is a function of the length of the break and performance time immediately before the break took place. Carlson and Rowe developed a learning-forgetting-learning model in which forgetting is modeled by a curve similar to the learning curve.

According to their model, forgetting is a function of the break length and the performance time prior to the break. Their forgetting curve

approach, assuming that the same learning curve may be used to model both learning and forgetting. However Bailey articulates that a measure of the forgetting rate is uncorrelated to that of the learning rate. He challenges that forgetting is “retrogression” along the original learning curve. Globerson and Levin presented a conceptual model hypothesizing that the forgetting process is a function of several factors, including turnover, communication and documentation.

Unlike learning models that are based on real data collected in industry, all relevant forgetting models are based either on conceptual theory or experimental data. This is because it is difficult to monitor and analyze the impact of interruptions. The issue of forgetting has attracted the attention of psychologists for sometime. Researchers in the area of memory decay for example, Hulse differentiated between short-term and long-term memory decay. Short-term memory has a limited capacity. If the information stored is not retrieved within 30 seconds or so, it will be totally forgotten.

Studies concerning the short-term memory involving experiments in which the subjects were asked to memorize a list of syllables and asked to repeat them after a specified time intervals of no more than a few moments. Long-term memory research is more applicable to forgetting in an organizational environment. Experiments in this area involve

length, typically up to few days. Klatzky reports the results of such a study showing that memory decay is a power function of the break length. For example, subjects forget 55 percent of the words within a six hour break time and 80 percent within 72 hours. While such break lengths between consecutive sessions may be relevant to organizational environments, the nature of the task is not.

Although the results of the psychological studies are not relevant to most practical situations, the research methodologies associated with these types of controlled experiments are advantageous for model development. Taking this approach, an experiment was designed to satisfy the following objectives.

- 1) Compare learning curves between the production line assembly and the single person assembly initially,
- 2) Evaluate the impact of forgetting on the length of break by using measures taken from Globberson and Levin.
- 3) To develop a lost time equation to be used in one of the measures.

The following section describes the experiment. It is followed by an analysis and a discussion of the results.

METHODOLOGY

The process concerned was, putting together a number of circulars and packaging them into one parcel. Each time the experiment took place

packages was used as a repetition. Four subjects who were involved in the assembly line were between 10 and 14 years old. In the forgetting experiment the time breaks between repetitions were 2, 4 and 8 days. Single person assembly was used, as this would be easier to control.

RESULTS AND ANALYSIS

The analysis is divided into two major parts. The first deals with the assembly line learning process versus the single person process. The second part analyses how forgetting is a function of break length. The power model described by equation was used as the basic learning model. Its validation was checked against the chart that the power model gives the two respective learning rates;

Single Person = 94%,

Assembly Line= 96%

These numbers were rounded to the nearest percentage. The percentage figures show that the single person assembly learns a little faster than the respective assembly line process. This would be expected as the single person has a larger ability to learn new techniques and change his or her process design. In this experiment the production line was in cramped conditions and therefore the ability to learn as a group was lowered, because not much experimentation occurred. What must be noted, though, is that there seems to be a cyclic trend. This can be explained by

seemed to get the participants motivated again. This signifies that the power model is correct and can be used for the main part of the experiment. To try to forecast the time it will take to do the task after a break period of D . The impact of forgetting may be evaluated on two levels: the initial level and the cumulative one.

INTRODUCTION

The modeling of human capital in economic growth is controversial. Aghion and Howitt distinguish two major frameworks within the endogenous growth literature, i.e. the 'Lucas' approach and the 'Nelson-Phelps' [NP] approach. The former, based on Lucas and shared by neo-classical growth theory, assumes that growth is driven by the accumulation of human capital. It treats human capital like an ordinary input in the production function. In particular, differences in growth rates across countries are assumed to be primarily due to differences in the rates of human capital accumulation.

The second approach, based on Nelson and Phelps (1966), relates growth to the stock of human capital which affects a country's ability to innovate and catch-up with more advanced countries. It is argued here that the characterization of Nelson and Phelps (1966) adopted by the NP approach is biased and driven by the controversy whether human capital

Nelson and Phelps' original hypothesis related to education speeding the process of technology diffusion and catch-up, not to the advance of the technological frontier (domestic innovation), which they assumed to occur at a constant exponential rate. However, they did conclude their paper by suggesting "the usual, straightforward insertion of some index of educational attainment in the production function may constitute a gross misspecification of the relation between education and the dynamics of production".

This statement seemed to have been confirmed by Benhabib and Spiegel and many others since then, i.e. economic growth seems to be unrelated to increases in educational attainment. Instead, BS found that their version of the Nelson and Phelps model, which included human capital in levels as well as human capital levels interacted with a technology lag factor was preferable and confirmed a positive role for human capital in growth. It seems fair to say that in their discussion, BS emphasized the 'first differences versus levels' controversy of the specification of human capital, rather than the technology diffusion/catch-up factor, thereby distracting from the arguably major contribution of Nelson and Phelps (1966). BS's results are suspect for a number of reasons.



EMPIRICAL RESULTS

The empirical analysis is conducted in two parts. First, BS's results for their Nelson- Phelps style model are re-examined. The focus is on their total sample of countries, as well as on their 'rich' group of economies. Second, by adding human capital stock data from Barro and Lee (2000) to the BS data set, it is possible to estimate their model. Throughout, the importance of correcting for data outliers is emphasized. This model for their complete data sample of developed and developing economies appears to favour the diffusion/catch-up component over the domestic innovation component.

It seems to indicate that, the diffusion/catch-up component is relatively unimportant, whereas the innovation component is positive and statistically significant. However, one has to be careful not to put too much weight on this finding. BS's rich country sample is determined by initial per capita income levels. It is not the same as the Organization for Economic Co-operation and Development(OECD) group of economies. All trimmed least squares regressions reported in this paper have been performed with the trimming proportion TRIM set at 0.005, 0.01, 0.025, 0.05, 0.1. Only the results for TRIM=0.01 are reported. All other results were very similar. Temple (2000) has emphasized the importance of differences between the developed countries, proxied by OECD

In general, the former groups of countries have very similar institutions, and the quality of their data is much higher. There are only 11 observations left once outliers are deleted. However, the trimmed least squares regression does give a first indication that BS's rich country results may not apply to the OECD group of economies. The innovation component remains statistically insignificant, whereas the diffusion component becomes statistically significant, which is the reverse of the BS results for rich economies. The reason why BS excludes many OECD economies from their data sample is lack of human capital stock data. This has since been remedied. Combining the latest Barro and Lee (2000) schooling data with the other data used by BS, one can estimate their models for a sample of 25 OECD economies. This again emphasizes the importance of robust estimation in growth regressions.

CONCLUDING COMMENTS

It is argued that the usual interpretation of the Nelson-Phelps approach to the modeling of human capital in growth regressions is biased due to the prominence of the 'first differences versus level' controversy. Emphasizing the technology diffusion/catch up component over the domestic innovation component, it is shown that at least for OECD economies interpretation of the Nelson-Phelps approach as well as the Lucas approach are supported, when outliers are deleted from the

higher schooling as the human capital proxy are reported. These are the only empirical growth studies, so far, that include both domestic and foreign R&D capital variables as well as a human capital variable side-by-side support. In common with many other cross-country growth studies, issues of simultaneity and data quality are not addressed. Moreover, the findings, obtained from only one cross-country data set, need to be confirmed with a wide variety of other data sets and model specifications. However, if they should stand up to further scrutiny, it would indicate that we have to refine the tests of the different approaches to modeling the relationship between human capital and growth, and have to focus more on the development of hybrid models.

SYSTEM DESIGN

EXISTING SYSTEM:

During recent decades, the learning effect has attracted particular attention in industry, business, and service sectors, since it was perceived as worthwhile to be taken into consideration during production planning as it can affect production profit. Although in industry, economy, management, and service sectors, numerous research works on the learning effect were done. Biskup and Cheng and Wang were among the pioneers that brought the concept of learning to the field of scheduling. Biskup investigated single processor scheduling problems, where the processing time of a job depends on the number of the previously performed jobs, i.e., on the job position in a sequence. Thus, the author modeled the learning effect by the job processing time being a non increasing function of a job position in a sequence, i.e., called further a position-based approach (model).

Further studies on problems with the same model were continued by Mosheiov. He investigated problems with this model for a single processor and identical parallel processor environments, respectively. Later, Mosheiov and Sidney extended Biskup's learning model by the introduction of different learning ratios a_j for jobs. Lee considered problems where the learning effect represented by the Biskup's function emerged in the presence of a deteriorating effect, i.e., the job processing

time can increase as the job waits to be processed. On the other hand, Koulamas and Kyparisis, besides considering problems with the learning effect on job processing times, also took into consideration that learning has an impact on setup times.

PROBLEMS OF EXISTING SYSTEM:

- 1) These approaches are not well discriminative and are rather crude simplifications of complex industrial problems.
- 2) Their major weakness is mainly due to an inaccurate modeling of a difference between the experience gained by performing easy and difficult jobs.
- 3) They can be applied only to the limited number of scenarios in which jobs are similar.
- 4) While the experience-based approach covers also settings with various jobs characterized by different grades of complexity, at the same time, it accommodates position or sum of normal processing-time-based models as special cases, consequently inheriting their properties. Furthermore, the flexibility of the experience-based approach enables modeling much more complex settings.

PROPOSED SYSTEM:

In this project, we focus on a problem with a new experience-based model that is described by an “S”- shaped curve. Moreover, scheduling problems with models and extended by will be also considered. The presented approach to the considered problems with the learning effect has to the best of our knowledge never been investigated in the scientific literature. The recent study about learning itself pointed out that, in practice, the learning curve is the “S”-shaped function. Nevertheless, it has not been considered in the scheduling context. It is easy to notice that the processing time of any job characterized by model is always positive. Therefore, no additional assumptions are required there. First, the three phases of the “S”-shaped learning curve are distinguished, where incipient and learning phases are described by a nonlinear function. Second, the providing of different experiences by jobs is enabled.

FEASIBILITY

FEASIBILITY

FEASIBILITY STUDY

A feasibility study is concerned to select the best system that meets performance requirements. These entities are an identification description, an evaluation of candidate systems and the selection of the best system for the job.

- Economic feasibility
- Technical feasibility
- Behavioural feasibility

ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs, and then the decision is made to design and implement the system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have an enhancement to approve.

TECHNICAL FEASIBILITY

Technical analysis centre on the existing computer system (Hardware, Software etc) and to what extend it can support the proposed addition. This involves financial considerations to accommodate technical enhancement. If the budget is a serious constraint, then the project is judged not feasible.

BEHAVIOURAL FEASIBILITY

An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do understandable that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of considering business.

DESIGN AND DEVELOPMENT

The design phase is a multi step process which focuses on system creation with the help of user specifications and information gathered in the above phases. It is the phase where the system requirements are translated to operational details.

SYSTEM DESIGN

System Design is the process of making the newly designed system fully operational and consistent in performance. The following steps have been followed in the implementation of the system.

- Implementation in planning
- User Training

As the part of implementation, the system is taken the site and loaded on to client's computer. These users are trained first and they run the system for a month. A detailed documentation is prepared for the employees and they trained to access the software. These users are trained first and they can run the system for a month.

After installation of software, the hardware specifications are checked. If hardware specifications are satisfactory, then the software is loaded for pilot run. User training starts at this time itself. Users will be given a user manual, which documents how to use the system and all the exception handling procedures.

INPUT DESIGN

Input design is the part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the number of people involved; it is the point of most contact for the users with the

computer system; and it is prone to error. If data going into the system are incorrect, then the processing and output will magnify these errors.

In this system inputs are given in two ways, the Existing users can directly enter into the system using login form, and new users have to register all their details in the registration form provided. Input design is the very important part in the project and should be concentrated well as it is prone to error. The data that are to be inserted are to be inserted with care as this plays a very important role. In order to get the meaningful output and to achieve good accuracy the input should be acceptable and understandable by the user.

OUTPUT DESIGN

Output design plays a very important role in a system. Getting a correct output is a task that has to be concentrated, as a system is validated as a correct one only if it gives the correct output according to the input. Here in this project in all the three days of inductions if the employee has completed all his/her input, then the output shows the status as completed or his status will be pending.

SOFTWARE SPECIFICATION:

JAVA

Java is related to C++, which is a direct descendant of C. The trouble with C and C++ is that they are designed to be compiled for a specific target. But Java is a portable, platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments. Java can be used to create two types of programs: applications and applets. An application is a program that runs on our computer, under the operating system of that computer. An applet is an application designed to be transmitted over the Internet and executed by a Java-compatible Web browser. Java is Simple, Secure, Portable, Object-oriented, Robust, Multithreaded, Architectural-neutral, Interpreted, High Performance, Distributed, and Dynamic.

SIMPLE

Java was designed to be easy for the professional programmer to learn and use effectively.

SECURITY:

When a java comparable web=server is used, the user can download applets without fear of virus infection. Java archives this protection by confining a java program to the java execution environment and not allowing it access to other parts of the computer.

PORTABILITY:

Many types of computers and operating systems are in use

programs to be dynamically downloaded to all various type of platform connected to the internet, some means of generating portable executable code is needed.

OBJECT-ORIENTED

The object model in java is simple and easy to extend, while simple types, such as integers are kept as high-performance nonobjective.

ROBUST

The ability to create robust programs was given a high priority in the design of java. To gain reliability, java restrict user in a few key areas, to force to find mistakes in early in program development. At the same time, java frees the user from having to worry about many of the most common causes of programming errors. Because java is strictly typed language, it checks the user code at compile time and it also checks the code at runtime.

MULTITHREADED

Java was designed to meet the real-world requirements of creating interactive, networked programs. To accomplish this, java supports multithreaded programming, which allows the user to write programs that do work simultaneously.

ARCHITECTURE-NEUTRAL

A central issue for java designers was that of code longevity and

guarantee exists that if you write a program today, it will run tomorrow, even in the same machine, operating system upgrades, processor upgrades, changes in core system resources can all combine to make program malfunction. But in java the goal was “write once, run anywhere, any time, forever”.

INTERPRETED AND HIGH PERFORMANCE

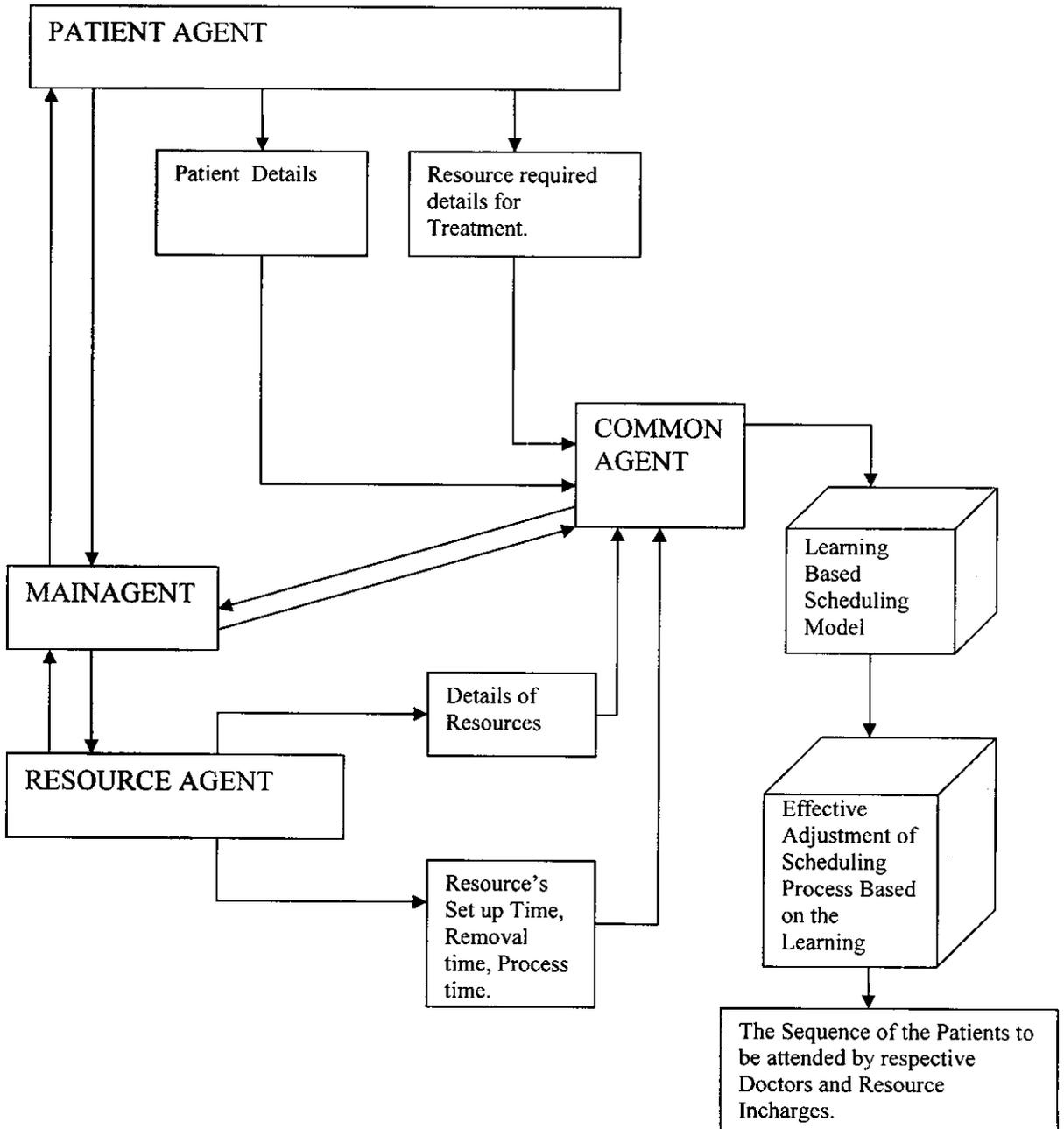
Java enables cross-platform program by compiling into a intermediate representation. The code can be interpreted on any system that provides java virtual machine. Cross-platform solution has done at the expense of performance. Java, however, was designed to perform well on very low-power CPU's. Java runtime machine that provides this feature lose none of the benefits of platform-independent code. “High performance cross-platform” is no longer an oxymoron.

DISTRIBUTED

Java is designed for distributed environment of the internet, because it handles TCP/IP protocols. The original version of java(Oak) include features for intra-address-space messaging. This allows objects on two computers to execute procedures remotely. Java has revived these interfaces in a package called remote method invocation (RMI).

BLOCK DIAGRAM

BLOCK DIAGRAM :



TESTING

TESTING

Testing is a process of checking whether the developed system is working according to the original objectives and requirements.

Testing is a set of activities that can be planned in advance and conducted systematically. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the global will be successfully achieved. In adequate testing if not testing leads to errors that may not appear even many months. This creates two problems,

- The time lag between the cause and the appearance of the problem.
- The effect of the system errors on the files and records within the system.

A small system error can conceivably explode into a much larger Problem. Effective testing early in the purpose translates directly into long term cost savings from a reduced number of errors. Another reason for system testing is its utility, as a user-oriented vehicle before implementation. The best programs are worthless if it produces the correct outputs. No other test can be more crucial. Following this step, a variety of tests are conducted.

- Unit testing
- Integration testing
- Validation testing

TEST PLAN

UNIT TESTING

A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. Achieving an error free program is the responsibility of the programmer. Program testing checks for two types of errors: syntax and logical. Syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. These errors are shown through error message generated by the computer. For Logic errors the programmer must examine the output carefully.

When a program is tested, the actual output is compared with the expected output. When there is a discrepancy the sequence of instructions must be traced to determine the problem. The process is facilitated by breaking the program into self-contained portions, each of which can be checked at certain key points .

Test case no.	Description	Expected result
1	Test for application window properties	All the properties of the windows are to be properly aligned and displayed
2	Test for mouse operations	All the mouse operations like click, drag, etc. must perform the necessary

FUNCTIONAL TESTING

Functional testing of an application is used to prove the application delivers correct results, using enough inputs to give an adequate level of confidence that will work correctly for all sets of inputs. The functional testing will need to prove that the application works for each client type and that personalization function work correctly.

Test case no	Description	Expected result
1	Test for all peers	All peers should communicate in the group.
3	Test for various peer in a distributed network framework as it display all users available in the group	The result after execution should give the accurate result.

NON-FUNCTIONAL TESTING

This testing used to check that an application will work in the operational environment.

Non-functional testing includes:

- Load testing
- Performance testing
- Usability testing
- Reliability testing

LOAD TESTING

Test case no.	Description	Expected result
1	It is necessary to ascertain that the application behaves correctly under loads when 'Server busy' response is received.	Should designate another active node as a Server.

PERFORMANCE TESTING

Test case no.	Description	Expected result
1	This is required to assure that an application perform adequately, having the capability to handle many peers, delivering its results in expected time and using an acceptable level of resource and it is an aspect of operational management.	Should handle large input values, and produce accurate result in a expected time

RELIABILITY TESTING

Test case no.	Description	Expected result
1	This is to check that the server is rugged and reliable and can handle the failure of any of the components involved in provide the application.	In case of failure of the server an alternate server should take over the job

SECURITY TESTING

It is necessary to check that the application's data is secured.

Test case no.	Description	Expected result
1	Checking that the user identification is authenticated	In case failure it should not be connected in the framework
2	Check whether group keys in a tree are shared by all peers	The peers should know group key in the same group

WHITE BOX TESTING

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases.

Using white box testing method, the software engineer can derive test cases.

Test case no.	Description	Expected result
1	Exercise all logical decisions on their true and false sides	All the logical decisions must be valid
2	Execute all loops at their boundaries and within their finite	All the loops must be

	operational bounds.	
3	Exercise internal data structures to ensure their validity.	All the data structures must be valid

BLACK BOX TESTING

Black box testing, also called behavioral testing, focuses on the functional requirements of the software. That is, black testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not alternative to white box techniques. Rather it is a complementary approach that is likely to uncover a different class of errors than white box methods. Black box testing attempts to find errors in the following categories.

Test case no	Description	Expected results
1	To check for incorrect or missing functions	All the functions must be valid
2	To check for interface errors	All the interface must function normally
3	To check for errors in a data structures or external data base access.	The database updation and retrieval must be done
4	To check for initialization and termination errors.	All the functions and data structures must be initialized properly and terminated normally

IMPLEMENTATIO

IMPLEMENTATION

System implementation is the process of making the newly designed system fully operational and consistent in performance. The following steps has been followed in the implementation of the system.

- Implementation in planning
- User Training

As the part of implementation, the system is taken the site and loaded on to client's computer. Some of the user's level, exposure to computer etc. These users are trained first and they run the system for a month. A detailed documentation is prepared for the employees and they trained to access the software. These users are trained first and they can run the system for a month.

After installation of software, the hardware specifications are checked. If hardware specifications are satisfactory, then the software is loaded for pilot run. User training starts at this time itself. Users will be given a user manual, which documents how to use the system and all the exception handling procedures.

SOURCE CODE :

COMMON AGENT :

```
package commanAgent;
import java.util.*;
import java.io.*;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.AudioFileFormat;
import jade.core.*;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.DFService;
import jade.domain.FIPAException;
public class CommanAgent extends Agent {
    Logger logFile;
    static int count = 0;
    class WaitPingAndReplyBehaviour extends SimpleBehaviour {
        private boolean finished = false;
        public WaitPingAndReplyBehaviour(Agent a) {
            super(a);}
        public void action() {
            ACLMessage msg = blockingReceive();
            String agents = "";
            String path="";
            String fname="";
            Object obj = msg.getSender();
            String sender = obj.toString();
            String[] senderName = sender.split("\\ :");
            senderName = senderName[1].split("\\ ");
            senderName = senderName[1].split("\\@");
            sender = senderName[0];
            System.out.println("Sender Name :\t"+sender);
            String c=msg.getContent();
            String cl[]=c.split("###");
            if(msg != null){
                if(msg.getPerformative()==
                    ACLMessage.NOT_UNDERSTOOD){
```

```

    IVED,ACLMessage.getPerformative(msg.getPerformative())
);
    logfile.log("No reply message sent");
    System.out.println("No reply message sent");
}
else{
    String content = msg.getContent();
    String file[] = content.split("###");
    if(CommanAgent.count == 0)
    {
        try{
            File out;
            path=newjava.io.File(".").getCanonicalPath();
            if(sender.equals("patient"))
            {
                fname=path+"\\Output\\patient.txt";
            }
            if(sender.equals("resource"))
            {
                fname=path+"\\Output\\resource.txt";
            }
            FileOutputStream wfile=new FileOutputStream(fname);
            wfile.write(file[2].getBytes());
            wfile.close();
            System.out.println("File
            "+file[3]+" Recieved and Stored into "+fname);
        }catch(Exception e){
            e.printStackTrace();
        }
        machine mn_alloc= new machine();
        int tardiness=0;
        Vector v = new Vector();
        int mcount=2;
        int mac=0;
        int schedule[][]= new int[6][1];
        String roww[]=null,st,rt,pt; //,finger-task[]=null,leg-
        task[]=null,head-task[]=null,eye-task[]=null,nose-
        task[]=null,kidney-task[]=null;
        int row[]=new int[3];
        try{
            FileInputStream fstream= new
            FileInputStream("Output\\res1.txt");
            DataInputStream in = new DataInputStream(fstream);
            BufferedReader br = new BufferedReader(new InputStreamReader(in));

```

```

    {if(mm==1)
    { mm=2;}
    if(strLine.contains("machine"))
    {mm=1; mac++; jj=0;
        if(mac>1) mn_alloc.add(mac-2,schedule); }
        if(!(strLine.contains("rid")|| strLine.contains("machine")))
        if(mm==2)
        {String tt[];
        tt=strLine.split(":");
        roww=tt[1].split("#");
        int jct=0,as=0;
        for(int i=0;i<roww.length;i++)
        {row[i]=Integer.parseInt(roww[i]);
        jct+=row[i]; }
        Calendar calendar = new GregorianCalendar();
        String am_pm;
        int hour = calendar.get(Calendar.HOUR);
        int minute = calendar.get(Calendar.MINUTE);
        int second = calendar.get(Calendar.SECOND);
        if(calendar.get(Calendar.AM_PM) == 0)
        am_pm = "AM";
        else
        am_pm = "PM";
        int ah=0,rm=0;
        if(jct>=60)
        { ah=jct/60;
        rm=jct-(ah*60); }
        ah=hour+ah;
        int mh,mx,tmin=minute+rm;
        if(tmin>=60)
        { mh=tmin/60;
        mx=tmin-(mh*60);
        ah+=mh; rm=mx;
        }else rm=tmin;
        schedule[jj++][0]=jct;
        v.add(jct);} }
        mn_alloc.add(mac-1,schedule);
        in.close();
    }catch (Exception e){

```

```

Vector vector2 = new Vector();

```

```

Vector vres = new Vector();

```

```

Hashtable machines = new Hashtable();

```

```

String      task[]=null;      //,finger-task[]=null,leg-task[]=null,head-
task[]=null,eye-task[]=null,nose-task[]=null,kidney-task[]=null;
    try{
        FileInputStream fstream = new
FileInputStream("Output\\pat.txt");
            DataInputStream in = new DataInputStream(fstream);
        BufferedReader br = new BufferedReader(new
InputStreamReader(in));
            String strLine; int mm=0; int jt=0; int cok=0;
            int ct10=0 ,t=0;//,tardiness=0;
            int zct=0;
            while ((strLine = br.readLine()) != null)
                {{ String tt[];
                    tt=strLine.split(":");
                    task=tt[1].split("#");
                    Hashtable ht =mn_alloc.get();
                    int mat[][]=(int[][])ht.get(jt);
                    for(int kb=0;kb<task.length;kb++)
                    {int zt=0;t=0; ct10=0;
                    if(task[kb].contains("finger-scan"))
                        { t=t+ct10; ct10= mat[0][0]; }
                    else if(task[kb].contains("leg-scan"))
                        { t=t+ct10; ct10=mat[1][0]; }
                    else if(task[kb].contains("head-scan"))
                        {t=t+ct10; ct10= mat[2][0]; }
                    else if(task[kb].contains("eye-scan"))
                        { t=t+ct10; ct10= mat[3][0];}
                    else if(task[kb].contains("nose-scan"))
                        { t=t+ct10; ct10= mat[4][0]; }
                    else if(task[kb].contains("kidney-scan"))
                        { t=t+ct10; ct10= mat[5][0]; }
                        zct+=ct10;      zt+=(t+zct);
                    vres.add(ct10+"=="+"0");
                    }}}
            in.close();
            int tak=2;
            while(tak<=vres.size())
            { String optimal[]= new String [tak];
            for(int i=0;i<tak;i++)
                { String gj=""+"vres.get(i);
                    optimal[i]=(gj);

```

```

        Vector vs= new Vector();
        int ty=tak,tez=tak;
        long fac=fact(tak);
        System.out.print("\n*****");
        while (vs.size()<fac)
        {
            int first_c1=0;int first_t1=0;
            String gg=""; Vector vst= new Vector(); int first_c=0; int
first_t=0;
            for(int u=0;u<tak;)
            {
                int ht=new Random().nextInt(tak);
                while(!vst.contains(ht))
                { vst.add(ht);
                    gg+=( "T"+ht+"-"); u++; int jk=0;
                    // System.out.print(" \n-----"+(jk++)+"-----
>" +optimal[ht]);
                    String fz[]=optimal[ht].split("==");
                    first_c=Integer.parseInt(fz[0]);
                    first_t=Integer.parseInt(fz[1]);
                    first_t1+=(first_t+first_c1);
                }
            }
            if(!vs.contains(gg))
            {
                first_c1=first_c;
                int sumtemp=first_c+first_t1;
                System.out.print("\n"+gg+" C+T=" +(sumtemp));

                if(sumtemp<small)
                {small=sumtemp; optsch=gg;}
                vs.add(gg);
            }
        }

        System.out.print("\noptimal schedule =" +optsch+" C+T="+small);
        tak++;
    }
} catch (Exception e){

```

PATIENT AGENT:

```
package patient;
import java.util.Date;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;
import java.io.FileInputStream;
import java.io.DataInputStream;
import java.io.File;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import jade.core.*;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.DFService;
import jade.domain.FIPAException;
public class PatientAgent extends Agent
{
    Logger logFile;
    public static Gui g = null;
    Thread T=new Thread(this);
    class WaitPingAndReplyBehaviour extends SimpleBehaviour
    {
        private boolean finished = false;
        File fileIn[];
        public WaitPingAndReplyBehaviour(Agent a)
        {
            super(a);}
        public void action()
        {
            ACLMessage msg = blockingReceive();
            String str = "", file = "";
            int training = 0;
            String sender = "", senderName[];
            Object obj = null;
            obj = msg.getSender();
            sender = obj.toString();
            senderName = sender.split("\\ :");
            senderName = senderName[1].split("\\ ");
            senderName = senderName[1].split("\\@");
```

```

String result = msg.getContent();
file = PatientAgent.g.testfile;
if(file.equals(""))
training = PatientAgent.g.trainfile.length;
else
training = 0;}
if(training > 0)
{fileIn = PatientAgent.g.trainfile;}
else{
fileIn = new File[1];
fileIn[0] = new File(file);}
String con = msg.getContent();
System.out.println(con );
for(int s = 0; s < fileIn.length ; s++)
{if(msg.getPerformative()

```

```

ACLMessage.NOT_UNDERSTOOD){

```

```

    logfile.log(msg.getSender().getName(),Logger.RECEIVED,ACL

```

```

Message.getPerformative(msg.getPerformative()));

```

```

    logfile.log("No reply message sent");

```

```

    System.out.println("No reply message sent");}

```

```

else{String split[] = con.split("###");

```

```

split = str.split("::");

```

```

msg.addReplyTo(aid);

```

```

if(fileIn.length == 1 || file.equals(""))

```

```

str = str+" Testing";

```

```

else

```

```

str = str+" Training";

```

```

str = str+" "+Gui.method;

```

```

ACLMessage reply = msg.createReply();

```

```

if(msg.getPerformative()== ACLMessage.QUERY_REF){

```

```

String content = con;

```

```

if ((content != null) && (content.indexOf("file") != -1)){

```

```

logfile.log(msg.getSender().getName(),Logger.RECEIVED,ACLMessage

```

```

e.getPerformative(msg.getPerformative()));

```

```

reply.setPerformative(ACLMessage.INFORM);

```

```

String format = "";

```

```

int size = 0;

```

```

try{System.out.println("Sending file...\t"+fileIn[s].getName());

```

```

FileInputStream fr=new FileInputStream(fileIn[s]);

```

```

size=fr.available();

```

```

byte[] b=new byte[size];

```

```

file = new String(b);
    System.out.println("\nFile Size :\t"+size);
} catch(Exception e)
{e.printStackTrace();}
if(s == fileIn.length-1){
reply.setContent(str+"###"+size+"###"+file+"###"+fileIn[s].getName()+
"###"+"last");
System.out.println("All Files Where Send to The Comman Agent!!!");
} else
reply.setContent(str+"###"+size+"###"+file+"###"+fileIn[s].getName()+
"###"+"cont");
logFile.log(((AID)reply.getAllReceiver().next()).getName(),
Logger.TRANSMITTED,ACLMessage.getPerformative(reply.getPerfor
mative()));
} else {
logFile.log(msg.getSender().getName(),Logger.RECEIVED,ACLMessag
e.getPerformative(msg.getPerformative()),msg.toString());
reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
reply.setContent("( UnexpectedContent (expected ping))");
logFile.log(((AID)reply.getAllReceiver().next()).getName(),Logger.TRA
NSMITTED,ACLMessage.getPerformative(reply.getPerformative()),repl
y.toString());} } else {
logFile.log(msg.getSender().getName(),Logger.RECEIVED,ACLMessag
e.getPerformative(msg.getPerformative()),msg.toString());
reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
reply.setContent("(
(Unexpected-act
"+ACLMessage.getPerformative(msg.getPerformative()+") ( expected
(query-ref :content ping))))");
logFile.log(((AID)(reply.getAllReceiver().next()).getName(),Logger.TR
ANSMITTED,ACLMessage.getPerformative(reply.getPerformative()),re
ply.toString());
send(reply);
try
{Thread.sleep(4000);
} catch(Exception e){
e.printStackTrace();
}} } else {
system.out.println("No message received");
}}
public boolean done() {
return finished;}}
protected void actv... () {

```

```

ServiceDescription sd = new ServiceDescription();
sd.setType("AgentcitiesPingAgent");
sd.setName(getName());
sd.setOwnership("TILAB");
dfd.setName(getAID());
dfd.addServices(sd);
try {
    DFService.register(this,dfd);
} catch (FIPAException e) {
    System.err.println(getLocalName()+" registration with DF
unsucceeded. Reason: "+e.getMessage());
    doDelete();}
logFile = new Logger(getLocalName());
logFile.log("Agent: " + getName() + " born");
System.out.println("\n\nHello Agent : "+getName());
WaitPingAndReplyBehaviour PingBehaviour = new
WaitPingAndReplyBehaviour(this);
addBehaviour(PingBehaviour); }
public static void main(String args[])
{try{
    PatientAgent.g = new Gui();
    PatientAgent.g.setVisible(false);
        InputForm i=new InputForm(PatientAgent.g);
        new CenterFrame(i);
        i.setVisible(true);
    }catch(Exception et){ System.err.print(""+et);}
}}
```

RESOURCE AGENT:

```

package resource;
import java.util.Date;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;
import java.io.FileInputStream;
import java.io.DataInputStream;
import java.io.File;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
```

```

import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.DFService;
import jade.domain.FIPAException;
public class ResourceAgent extends Agent {
    Logger logFile;
    public static Gui g = null;
    class WaitPingAndReplyBehaviour extends SimpleBehaviour {
        private boolean finished = false;
        File fileIn[];
        public WaitPingAndReplyBehaviour(Agent a) {
            super(a);}
        public void action()
        { ACLMessage msg = blockingReceive();
          String str = "", file = "";
          int training = 0;
          if(msg != null)
          { String sender = "", content = "", senderName[];
            Object obj = null;
            obj = msg.getSender();
            sender = obj.toString();
            senderName = sender.split("\\ :");
            senderName = senderName[1].split("\\ ");
            senderName = senderName[1].split("\\@");
            sender = senderName[0];
              String result = msg.getContent();
              file = ResourceAgent.g.testfile;
              if(file.equals(""))
                  training = ResourceAgent.g.trainfile.length;
              else
                  training = 0;
            if(training > 0)
                {fileIn = ResourceAgent.g.trainfile;}
            else
                {
                    fileIn = new File[1];
                    fileIn[0] = new File(file);
                }
            String con = msg.getContent();
            for(int s = 0; s < fileIn.length ; s++)

```

```

    {
        logFile.log(msg.getSender().getName(),Logger.RECEIVED,ACL
Message.getPerformative(msg.getPerformative()));
        logFile.log("No reply message sent");
        System.out.println("No reply message sent");
    }else{
        String split[] = con.split("###");
        split = str.split("::");
        AID aid = new AID("comman@"+Gui.host+":1099/JADE");
        if(fileIn.length == 1 || file.equals(""))
            str = str+" Testing";
        else
            str = str+" Training";
        str = str+" "+Gui.method;
        ACLMessage reply = msg.createReply();
        if(msg.getPerformative()== ACLMessage.QUERY_REF)
        {content = con;
        if ((content != null) && (content.indexOf("file") != -1))
        {
            logFile.log(msg.getSender().getName(),Logger.RECEIVED,ACLMessage
.getPerformative(msg.getPerformative()));
            reply.setPerformative(ACLMessage.INFORM);
            String format = "";
            int size = 0;
            try{System.out.println("Sending file...\t"+fileIn[s].getName());
            FileInputStream fr=new FileInputStream(fileIn[s]);
            size=fr.available();
            byte[] b=new byte[size];
            fr.read(b);
            file = new String(b);
            System.out.println("\nFile Size :\n"+size);
            }catch(Exception e)
            {e.printStackTrace();}
            if(s == fileIn.length-1)
            {reply.setContent(str+"###"+size+"###"+file+"###"+fileIn[s].getName()
+"###"+"last");
            } else
            reply.setContent(str+"###"+size+"###"+file+"###"+fileIn[s].getName()+
"###"+"cont");
            logFile.log(((AID)reply.getAllReceiver().next()).getName(),
Logger.TRANSMITTED,ACLMessage.getPerformative(reply.getPerfor

```

```

send(reply);
try{
Thread.sleep(4000);}
catch(Exception e)
{e.printStackTrace();}} }
    else
    {
        System.out.println("No message received");
    }
}
public boolean done()
{
    return finished;
} }
protected void setup() {
    DFAgentDescription dfd = new DFAgentDescription();
    ServiceDescription sd = new ServiceDescription();
    sd.setType("AgentcitiesPingAgent");
    sd.setName(getName());
    sd.setOwnership("TILAB");
    dfd.setName(getAID());
    dfd.addServices(sd);
    try {
        DFSservice.register(this,dfd);
    } catch (FIPAException e) {
        System.err.println(getLocalName()+" registration with DF
unsucceeded. Reason: "+e.getMessage());
        doDelete();
    }
    logFile = new Logger(getLocalName());
    logFile.log("Agent: " + getName() + " born");
    System.out.println("\n\nHello Agent : "+getName());
    WaitPingAndReplyBehaviour PingBehaviour = new
WaitPingAndReplyBehaviour(this);
    addBehaviour(PingBehaviour);}

public static void main(String args[])
{
    ResourceAgent.g = new Gui();
    new CenterFrame(ResourceAgent.g);
    ResourceAgent.g.setVisible(false);
    InputForm i=new InputForm(ResourceAgent.g);
    new CenterFrame(i); }}

```

FACILITATOR AGENT:

```
package main;
import java.io.*;
import java.util.*;
import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.AudioFileFormat;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import jade.core.*;
import jade.lang.acl.*;
import jade.gui.*;
import jade.tools.DummyAgent.*;
import jade.domain.FIPAAgentManagement.Envelope;
public class FacilitatorAgent extends JFrame implements ActionListener
{
    JMenuBar mb = null;
    JLabel lcomman = null, ldatabase = null, lresource = null, lpatient =
null;
    JTextField tcomman = null, tdatabase = null, tresource = null,
tpatient = null;
    JButton but = null;
    static String patient = null, comman = null, database = null,
resource = null;
    public static String agents = "";
    public FacilitatorAgent()
    {
        super("Facilitator Agent GUI");
        lcomman = new JLabel("Comman Agent");
        ldatabase= new JLabel("Database Agent");
        lresource = new JLabel("Resource Agent");
        lpatient = new JLabel("Patient Agent");
        tcomman = new JTextField();
        tdatabase = new JTextField();
        tresource = new JTextField();
        tpatient = new JTextField();
        tcomman.setText("comman");
        tdatabase.setText("database");
        tresource.setText("resource");
        tpatient.setText("patient");
    }
}
```



Facilitator Agent GUI



Patient Agent

Comman Agent

Database Agent

Resource Agent

```

getContentPane().add(lcomman);
getContentPane().add(ldatabase);
getContentPane().add(lresource);
getContentPane().add(lpatient);
getContentPane().add(tcomman);
getContentPane().add(tdatabase);
getContentPane().add(tresource);
getContentPane().add(tpatient);
getContentPane().add(but);
lpatient.setBounds(30,30,150,25);
tpatient.setBounds(180,30,100,25);
lcomman.setBounds(30,70,150,25);
tcomman.setBounds(180,70,100,25);
ldatabase.setBounds(30,110,150,25);
tdatabase.setBounds(180,110,100,25);
lresource.setBounds(30,150,150,25);
tresource.setBounds(180,150,100,25);
but.setBounds(120,200,100,25);
but.addActionListener(this);
setVisible(true);
setSize(400,400); }

```

```

public void actionPerformed(ActionEvent ae)
{ Object objButton=ae.getSource();
  if(objButton.equals(but))
  {
    patient = tpatient.getText();
    comman = tcomman.getText();
    database = tdatabase.getText();
    resource = tresource.getText();
    if(patient.equals("") || comman.equals("") ||
database.equals("") || resource.equals(""))
{ JOptionPane.showMessageDialog(this, "Fields can't be blank" );
return;
}else
{ agents = comman+"::"+database+"::"+resource+"::"+patient+"::da0";
String arg[] = {"-gui"};
jade.Boot.main(arg);
setVisible(false); } } }
public static void facilitateService(ACLMessage msg)
{ Agent agnt = new Agent();
  DummyAgent myAgent = null;

```

```

System.out.println("\n\nmsg :\n"+msg.getContent());
myAgent = new DummyAgent();
obj = msg.getSender();
sender = obj.toString();
senderName = sender.split("\\ :");
senderName = senderName[1].split("\\ ");
senderName = senderName[1].split("\\@");
sender = senderName[0];
ACLMessage toAgent = msg.createReply();
toAgent.clearAllReceiver();
if(sender.equals(patient))
    {System.out.println("\n\nContent Recieved from Patient Agent and
Send it to Comman Agent");
AID aid = new AID(comman.concat("@prabu2:1099/JADE"));
toAgent.addReceiver(aid);
aid = new AID("da0@prabu2:1099/JADE");
toAgent.setSender(aid);
System.out.println("Sender :"+toAgent.getSender());
}if(sender.equals(comman))
{
    String content1 = msg.getContent();
    String[] split = content1.split("###");
    split = split[0].split(" "); }
    if(sender.equals(resource))
        {System.out.println("Content Recieved from Resource Agent and
Send it to Comman Agent");
AID aid = new AID(comman.concat("@prabu2:1099/JADE"));
toAgent.addReceiver(aid);
aid = new AID("da0@prabu2:1099/JADE");
toAgent.setSender(aid);
System.out.println("Sender :"+toAgent.getSender());
}
    if(sender.indexOf("patient") != -1)
        {content = (msg.getContent()).trim();
}else
    System.out.println(""); }
public static void main(String args[])
{
    FacilitatorAgent obj = new FacilitatorAgent();
    new CenterFrame(obj);
    obj.setVisible(true); }}

```




File Actions Tools Remote Platforms Help



- AgentPlatforms
 - "prabu:1099/JADE"
 - Main-Container

name	addresses	state	owner
------	-----------	-------	-------

--	--	--	--

Enter the Patient Details

patient id

patient name

age

sex

address

phone/mobile

- finger-scan
- leg-scan
- head-scan
- eye-scan
- nose-scan
- kidney-scan



- AgentPlatforms
 - "prabu:1099/JADE"
 - Main-Container
 - Container-1

name	addresses	state	owner
NAME	ADDRES...	STATE	OWNER



res id:

machine id:

setup time

removal time

processing time

finger-scan:

leg-scan :

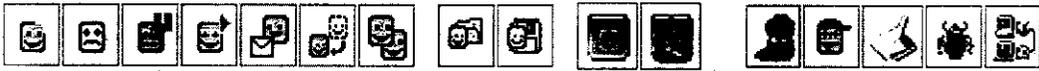
head-scan :

eye-scan :

nose-scan :

kidney-scan:

File Actions Tools Remote Platforms Help



- AgentPlatforms
 - "prabu:1099/JADE"
 - Main-Container
 - Container-1
 - Container-2
 - Container-3

name	addresses	state	owner
------	-----------	-------	-------

C:\WINDOWS\system32\java.exe

```
INFO: Service jade.core.messaging.Messaging initialized
Apr 1, 2010 7:20:12 AM jade.core.BaseService info
INFO: Service jade.core.mobility.AgentMobility initialized
Apr 1, 2010 7:20:12 AM jade.core.BaseService info
INFO: Service jade.core.event.Notification initialized
Apr 1, 2010 7:20:12 AM jade.core.AgentContainer info
INFO: -----
Agent container Container-81,java is ready.
```

```
Sender: patient
File patient.txt created and stored into file system in storage\reference\
c:\Output\patient.txt
```

```
*****
I1-I0- C+I-130
I0-I1- C+I-130
optimal schedule =I1-I0- C+I-130
```

```
*****
I2-I0-I1- C+I-150
I2-I1- I0- C+I-130
I1-I0-I2- C+I-80
I1-I2-I0- C+I-130
I0-I1-I2- C+I-80
I0-I2-I1- C+I-150
optimal schedule =I1-I0-I2- C+I-80
```


CONCLUSION:

In this paper, besides the study of scheduling problems with the well-known experience-based learning models, we also introduced and analyzed a problem with a new model of the learning effect that is based on the “S”-shaped learning curve. We proved that the make span minimization problem on a single processor is NP-hard or strongly NP-hard with most of the considered learning models. A number of polynomially solvable cases were also provided. Unfortunately, the complexity status of the problem $1|LEW|C_{\max}$ was not determined; hence, it will be the aim of further research, as well as a design of exact and fast approximation algorithms for NP-hard cases of the presented problems.

REFERENCES:

- [1] E. L. Thorndike, "Animal intelligence: An experimental study of the associative process in animals," *Psychol. Rev. Monogr.*, vol. 2, no. 4, pp. 1–109, 1898.
- [2] L. L. Thurstone, "The learning curve equation," *Psychol. Monogr.*, vol. 26, no. 3, pp. 1–51, 1919.
- [3] T. P. Wright, "Factors affecting the cost of airplanes," *J. Aeronaut. Sci.*, vol. 3, no. 2, pp. 122–128, Feb. 1936.
- [4] E. C. Keachie and R. J. Fontana, "Production lot sizing under a learning effect," *Manag. Sci.*, vol. 13, no. 2, pp. 102–108, Oct. 1966.
- [5] R. B. Jordan, "Learning how to use the learning curve," *N.A.A. Bull.*, vol. 39, no. 5, pp. 27–39, 1965.