On the Design and Integration of Education Simulation Games in Engineering Business Management

by

Siu-Wing Cheung
B.A., MSc., MHKIE, MBCS, C.Eng.

A thesis submitted for the Degree of

Doctor of Philosophy

Department of Engineering

University of Warwick

March 1996

ACKNOWLEDGMENTS

I would like to thank my supervisors, Prof. S.K. Bhattacharyya and Mr. Rajat Roy for their guidance and help in the last seven years. Also, many thanks to Mr. Jay Bal. who has spent a lot of his valuable time to advise me on my project. Thanks also go to all my colleagues in the simulation team especially Mr. John Waller who help me to write my first simulation program.

Also, I would like to thank my family and my friends, Miss Regina Tse, Miss Mandy Yam, Miss Lillian Wong, Miss Kitty Sit, Miss Sara Lee, Miss Teresa Lee, Miss Harvina Chow, Miss Dilys Tse, Miss Eliane Tang and Dr. W.H. Ip who have given their support all the way through.

Finally, I would express my sincere thanks to Prof. W.S. Lau who has spent a lot of his valuable time to proof-read my thesis and guide me through the thinking process.

DECLARATION

I declare that the work presented herein is my own work and has not been submitted for another award

SUMMARY

Manufacturing has evolved from production-oriented to customer-oriented in the last century. The modern management approach is 'total integration' which is the integration of 'functional optimization', 'internal integration' and 'external integration'. Nowadays, Engineering and Business can no longer be viewed as two separate entities. Engineering Business Management (EBM) becomes the norm of modern manufacturing management.

In universities, simulation games have been identified as one of the best teaching media in teaching EBM concept. Hundreds of good simulation games are thus required for different subject areas and education objectives. However, game design procedure is always a 'black box' and game design philosophy of EBM games is even a 'black box' within a 'black box'. This research attempts to develop a new generic approach in designing education simulation games in Engineering Business Management. The approach is broken into two phases: micro and macro level.

In micro level, game design philosophy is studied. The appropriate levels of formats and features of EBM games are determined and the weaknesses commonly found in games are examined and rectified. The findings are confirmed and evaluated by conducting three surveys between 1992 to 1994.

In macro level, a standard interface is established for all EBM games so that functional games which are built on this platform can be integrated together to form internal management games which can be further integrated into a supply chain network by hierarchical modular structure. The hierarchical structure allows the games to cover some EBM areas that have never been mentioned by any other game before.

Two functional games, a production game and a marketing game have been built as prototypes to demonstrate the design approach. In micro level, the established design criteria minimize the chance of developing a 'bad' game. In macro level, the formation of internal management games and supply chain networks by hierarchical modular modeling opens a new platform for game design.

Hence, the research is titled 'On the Design and Integration of Education Simulation Games in Engineering Business Management'.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

APPENDIXES

ABBREVIATIONS

EBM	Engineering Business Management
EOQ	Economic Order Quantity
JIT	Just-In-Time
JIT II	Just-In-Time II
MRP I	Material Requirement Planning
MRP II	Manufacturing Resource Planning
OPT	Optimal Production Technology
QC	Quality Circle
TQM	Total Quality Management

CHAPTER 1

INTRODUCTION

1.0 Trends in Manufacturing

1.1 Late 19th Century - Early 20th Century : Production Oriented

The saying 'the customer is always right' is probably as old as business itself but it was not always true. In the late nineteenth century and early part of the twentieth century the fundamental role of business was seen as production. Manufacturers were in a suppliers' market and faced with virtually insatiable demand for goods and services. Firms concentrated on production and productive efficiency in order to bring down costs. Product decisions were taken first with production implications in mind. Firms tended to manufacture and offer products that they were 'good at producing', with customers' requirements and satisfactions of secondary importance. Firms tended to be 'production oriented' and the production man was the most important person in the organisation. The principal planning tool employed was the Gantt Chart, developed in 1917 and 1918 to plan for war production. This production mentality was workable as long as a sellers' market existed. However, many firms had to change their attitudes as the world economy drifted into recession in the 1920s and 1930s and to produce was no longer enough. (Evans 1993)

1.2 1920s - 1960s : Sales Oriented

The world economic recession of the early twentieth century changed the minds of business people. Many firms failed and fortunes were lost.

Unemployment was high and effective demand slumped. Production capacity was under-utilised and there were many unsold goods. Gradually business people began to realise that it was not enough simply to produce goods as efficiently as possible. For profits to materialise such goods had to be sold.

The guiding business philosophy of many firms switched from production to sales orientation. The sales concept believed that effective demand could be created by sales techniques. Sales volume was the most important criterion, and planning horizons tended to be relatively short term. The actual customer, and how customers might perceive the value or utility of the goods being sold, were of secondary importance. Philip Kotler defined this selling concept as: "*A management orientation that assumes all customers will either not buy or not buy enough of the organisation's products unless the organisation makes a substantial effort to stimulate these interests in its products.*" (Jewell 1993)

In a sense, sales orientation was a conceptual step forward because although goods and services were still produced with little regard to customer requirements, at least it was realised that products did not sell themselves as a matter of course.

In order to achieve a competitive advantage greater importance was attached to product differentiation and branding. Advertising, sales promotion and other sales techniques were used to increase competitive edge. These techniques were used to sell the product rather than to communicate and inform or to increase customer satisfaction. This sales approach lasted beyond World War

II. Nevertheless, this period saw the development of a number of techniques that are still used today in modern marketing. (Krajewski 1990)

After World War II, the impact of technology on manufacturing was spreading at an accelerating pace, especially in electronics, information, communication and materials. More than 15 years of economic depressions and war, created an almost 20-year period of unbroken growth opportunities for business enterprises. There was no need to change any marketing techniques. The products were almost sold by themselves. So long as the external market opportunities continued to grow, this approach to business planning worked reasonable well.

At the same time, manufacturers were still trying to lower their production cost and increase productivity. Machines or labours were not supposed to idle in any circumstances. As a result, inventory level was sky level. Numerous mathematical models were constructed to optimise the production cost or time with operation research techniques introduced during World War II for military purpose. Shop floor scheduling, line balancing, machine layout and numerous inventory models were products of the period. However, these techniques never paid any attention to the inter-relationships between functional areas. Until mid-century, business planning in most firms was mainly cost control of functional departments which the planning horizon was seldom more than one year.

1.3 1970s - Present: Customer Oriented

Beginning from 1970, manufacturing changed rapidly because of recession, advance in technology, Japan's competition and customers become more knowledgeable. In autumn 1973, the lengthy economic recession originally brought by the oil crisis sparked off by the Middle East war. The dramatic increase in oil price affected the cost of energy as well as the cost of many oil-based products like synthetic fibres and plastics.

At the same time, due to the rapid development of communication, transportation and distribution, global competition took place. The competition from south East Asia countries especially Japan getting more and more seriously. The manufacturing firms began to realise that it is essential to produce goods according to the customers' needs. (Lancaster 1988)

Although this customer oriented marketing concept started as early as 1950s in United States, it did not come to serious practice until 1970s. Since then, customer approach has been adopted as the central business philosophy by many firms throughout the world. This concept suggests that in order for a firm to make a long term profit it must ascertain the genuine needs and wants of specifically defined target markets and then produce goods and services that satisfy customer requirement. (Jewell 1993)

In order to satisfy the customer requirement, high quality products have to be produced to increase the competitive edge. According to a survey in 1985, 'quality' was ranked as the most important competitive edge of UK manufacturing plant. Similar surveys from USA as well as from Europe showed similar results that quality has been ranked as the top competitive weapon. (New 1991)

-Internal Integration

Japan is probably the first country to supply high quality product at low cost. Beginning from 1970s, several large companies in the States like 'Ford Motors' and 'General Electric' sent their management teams to Japan to look for the secret of success. Suddenly, almost all world class manufactures were adopting some sort of 'Just In time' production system (JIT). However, they only copied the techniques, the system and the procedures of JIT but not the spirit. JIT is in fact a production philosophy more than a production control system. The key factor of JIT is the total commitment of the company, from top management to the workshop labours, from purchasing to delivery - a true internal integration. Other famous production techniques like 'Total Quality Management (TQM),' 'Quality Circle (QC)', 'Total Preventive Maintenance (TPM),' are also based on this philosophy. It may still be necessary to have the functional departments to perform the daily operations, but the management as well as the labour have to possess the same company objectives and view the company as a whole.

The traditional specialised departments which carry out different company tasks do not apply here. This management approach challenges every member of a company and ultimate satisfaction gained by the customer is a function of the totality of all company department acting in unison. By 1991, around one-fourth to one-third of companies have been effective in organising for functional integration. (Richardson, Trunick 1991)

-External Integration

In 1990s', the low cost, good quality production is only the admission to the arena. Customer wants more. They not only require the high quality product at low price but also at the right time. Time becomes the major competitive weapon in 1990s. In order to deliver the goods on time and have an immediate response to the changing needs of the customers, it is vital to keep close relationships with the suppliers. Companies begin to look at external integration including supply chain management, channel integration, purchasing responsibility, and partnership for strategic alliance.

Supply chain management is a loop which starts with the customer and it ends with the customer. Through the loop flows all material and finished goods, all information, even all financial transactions. The whole business will be viewed as one continuous process. The process absorbs such traditionally distinct functions as forecasting, purchasing, manufacturing, distribution, and sales and marketing into a continuous flow of business interactions. Functional departments are structured as a pipeline that stretches between a company's suppliers and its customers.

Although this theory has been around for more than 20 years but it is always been hard for companies to bring it into being. As the increase in technology in information systems and the cost of making information available to more people has steadily gone down while the physical costs of business such as facilities and inventory have steadily risen, supply chain strategy reached a point of maturity. Companies begin to realise that its most important concern was not the management of various functions, but the quality of the service. Management Tools such as JIT II are always mentioned in nowadays manufacturing planning for external integration. However, the most important factor of integration is not the system itself but the people working in the system. (Distribution, 1988; 1990)

2.0 Change For the Better

The following graph summaries the changes of manufacturing in the last century.

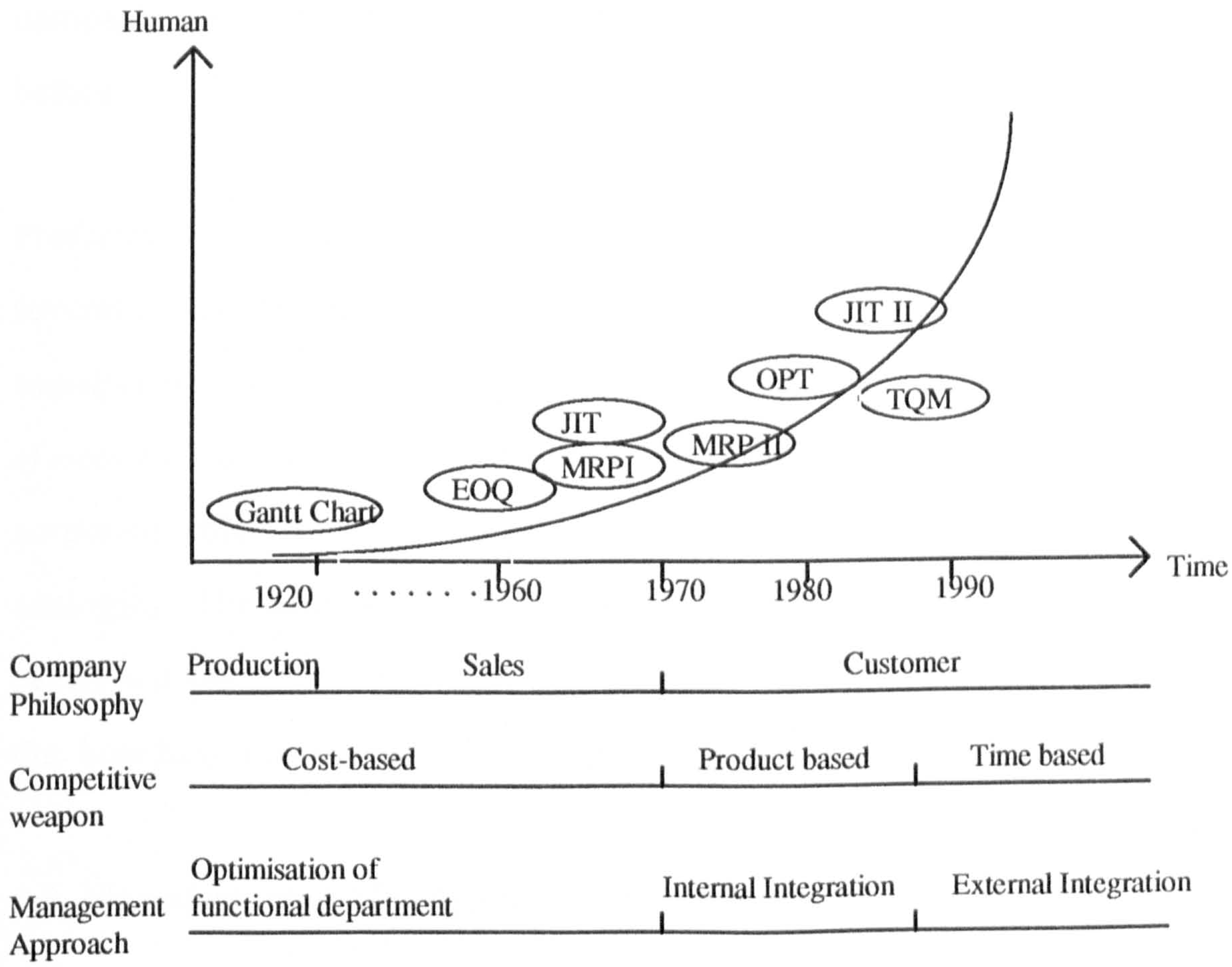


Fig. 1.1 Schematic diagram of changes of manufacturing in the last century

The main areas of change can be summarised as follows:

- Changes in the market
- Changes in competition
- Changes in product design
- Changes in manufacturing technologies
- Changes in management control methods

Manufacturing management changes from optimisation of functionalised departments to systems approach which integrates various departments, the whole company, the markets and the suppliers together to provide a better service to the customers. The achievement of these objectives is clearly dependent upon a series of company activities working in unison. The key company functions need to be inter-linked in a more responsive way than before.

Professor S.K. Bhattacharyya, Head of Warwick Manufacturing Group, on several occasions, has emphasised that "*if everyone works to the same goal and asks himself at every turn 'how will this decision affect the profitability of the firm? ' the chance of success is that much greater.*" He feels more emphasis should be placed on the corporate objective of profitability. He returns repeatedly to medicine for analogies. The surgeon must look at the body as the total system. "*What good is putting in a new heart if your kidney fails*?". So an integrated approach driven from the boardroom is essential. (Birmingham Post May 1, 1986)

3.0 Manufacturing Management Education

While the manufacturing management approach in the industry is changing, universities continue to improve and update their curricula to suit the needs of the industry. Until 1970, functional optimisation was still the key management approach of the manufacturers. The course scheme of manufacturing management in most universities was thus subject-based. A typical course scheme might include 'Production Planning and Control', 'Project Planning', 'Design for manufacturing', 'Manufacturing Technology', 'Metallic Materials Selection', etc. At this stage, manufacturing management education emphasised mainly on production area. Optimisation of the production cost was the key function of the production engineers. The true needs of the customers were hardly taken into account. Since product design was 'Design for manufacturing' and not 'Design for customers', there was hardly any subjects concerning customer needs.

Eventhough within the subject itself, each topic was taught separately. For example, in a typical 'Production control and management' course, topics like 'line balancing', 'facility layout', 'inventory control' and 'production scheduling' were taught by the lecturers individually assuming that there was no inter-relationships between each function. Operations research techniques were applied to each individual function looking for the optimal solution. Since manufacturers believed that the sum of individual optimal solutions result in optimal total production cost, that was how the course was taught.

In 1970s, while the world industry was busy to understand the concept of internal integration and adopt the changes in their production lines, the academics were changing their course schemes in manufacturing management. Academics began to realise that Business and Engineering courses can not be viewed as two separate entities. As a result, several business and finance courses were added in the curricula of manufacturing engineering degree courses. Engineering Business Management (EBM) became the norm of manufacturing management in which finance, technology and business management were integrated together. Also, courses like 'Management of Integration' and 'Total Quality Control' were introduced to integrate different functions together. (Lau 1993)

Functional subjects were still necessary to remain in the course scheme to help the students to understand the detail production operations. Since most of the optimisation techniques were represented by mathematical models, the techniques could be easily learnt from lectures and tutorials. However, internal integration is a management concept or a philosophy which could hardly be represented by a quantitative model. Other teaching media, like case studies, simulation games and projects were used by academics to convey this message. It is uncommon nowadays to find any degree course without project requirement in their curricula. Although project is one of the best ways to integrate different functions together, it is time consuming and most students can only work on one to two projects in a year of study. Projects also place heavy loading on the supervisor who have to guide the students step by step through out the projects. Case studies and simulation games are very common in tutorials. They are quite similar and sometimes, simulation games are referred as 'Dynamic case studies'. However, finding an appropriate case study or simulation game is always difficult as different case studies and games are required for different subject areas. Also, as the concept of engineering business management keeps on growing, the content of the case studies and simulation games are required to be updated from time to time.

When time-based production came in place in late 1980s', courses like 'Integrated logistic support' and ' Supply Chain management' are commonly found in universities. Although the importance of having good manufacturers-suppliers relationship is always highly emphasised in production literature, teaching this concept of external integration to the students is difficult. Telling them what happens is one thing and making them understanding is another thing. Even with the help of human management courses such as

'Human factors in industry', teaching human factors is always a headache to the lecturers. Role-play sometimes may be used but effect is hard to measure. Although numerous management games and case studies are produced around supply chain management, human factors are hardly covered in the games. (Logistics Today 1993)

Although universities try very hard to address each management approach, 'optimisation of functional department', 'internal integration' and 'external integration' in their course design, they lack total integration. It has to be emphasised that these approaches do not replace each other, they exist at the same time in a manufacturing company. In real life, a company has to co-ordinate different functional departments and at the same time, try to minimise the cost of each function and maintain good relationships with the suppliers. For education purpose, learning engineering business management bits and pieces can make the students understand the detail of the management gradually. However, a total integration of all these approaches is vital so that a true picture of the industry can be shown to the students.

4.0 Need for a Hierarchical Modular Integrated Game

In order to show the students a total picture of the industry, the existing teaching techniques need to be re-designed and re-structured. Instead of creating new teaching medium as the effect of which can hardly be measured, modification and improvement on the existing medium is more favourable. Although both case studies and simulation games are good effective teaching tool in engineering business management, however, games are definitely more enjoyable by students. In fact, game is just a dynamic form of case study with feedback. In addition, since games are used less frequently than case studies by lecturers in universities, the potential of simulation games have not been fully revealed in education and with the help of computer programming, new game structure can be evolved for better education effect.

There are thousands of educational games available in the market. Even in UK alone, there are more than 50 games currently used in universities or companies for in-house training in engineering business areas. They are either functional games which address in particular functional areas or total management games which try to integrate all the important internal aspects together. A functional area in a business is a specialised activity which contributes to the overall business objectives, and which are characterised by distinctive skills and culture. Although the existing games cover most topics in EBM, they lack integration in between. The students play a functional game in learning 'production planning and control' and another functional game in quality management. When integration is desired, total management games are played. Unfortunately, the functional games and the total management games are not compatible in either scenarios or parameter settings. They have no links in between. The idea of integration is in fact lost in this kind of integrated arrangement. In addition, a total management game is usually fairly complicated with 20 or more decisions in each round and the students will take a long time to get familiar with the game environment.

For the best education results, functional games under the same scenario should be used in different functional subjects. Then, integration of all these functional games together become an integrated game where students can understand the inter-relationship between the functional elements which have been considered before separately. Moreover, these set-ups save time for both students and game administrators because learning does not start until students are fully familiar with the scenarios. Since the scenario, input and output formats are similar in each game, the students take less time to enter into the learning process..(Cheung 1991)

However, to integrate all functional aspects together may seems impractical and even though the work may be done, it may actually provide adverse effect on the students since the game will be too complicated to be handled. Also, the main ideas that the instructor would like to get through to the students may be lost during the game. So, what functional areas should be integrated together? It may be true that every aspect of engineering business functions are important. Trying to compare the importance of different functions in engineering business is impractical if not impossible. Hence, there is no point in trying to justify what are the most appropriate engineering business elements in a game.

Each functional game has its own objective and a game instructor can only have limited control over the game scenario by changing the game parameters. For different subject areas, the instructor chooses different functional games. In order to satisfy the needs of the EBM education, game authors keep on writing new management games. It is not difficult to discover that a lot of effort have been duplicated as similar engineering business elements can be found in different games. The game authors try very hard to create different scenarios, select different elements in their games but they never consider integrating other authors' games for their own use.

One way to solve this problem is to establish a standard interface so that all functional games can be integrated together. Under this standard structure, each functional game will become a module and by putting different modules together will form an integrated game. As more and more functional games are formulated under this standard interface, a library of modules will be formed for future usage. The way of integrating different games together will become almost infinite and no more effort will be wasted on defining the same element by different game authors

Modular structure is not something new. It has been used in computer programming for many years. A module is a 'library routine' which can be used by any programmer in their programs. No one will waste any time to program a 'sort procedure' which can be easily called out from the library as long as the interface or the passing of parameters are compatible. With the same token, a simulation game can also be structured in this way.

This concept can also be further elaborated into a marco level in which each total management game is an integrated module which is composed of functional modules. Integration of all these integrated modules form a supply chain network in which the characteristics of supply chain can be studied. This hierarchical modular structure enables the reusability of functional modules as well as integrated modules. A instructor can create his own unique simulation game by integrating modules from the library and this user defined simulation game will be 'correct by construction'. Since the library function games will be fully characterised and tested, the overall time to create an integrated games will be drastically reduced. This approach makes use of the concept of modularity endorsed by Zeigler (1987, 1990) and the idea was further elaborated by Cheung in 1992 and 1994. (Cheung 1992; Cheung 1993; Cheung 1994)

Another major advantage of building simulation games with hierarchical structure is the capability of incorporating human factors in a game. For traditional game authors, human factors are always difficult to be included because they are restricted by the game structure.

The establishment of a standard interface can help out the construction of an integrated game which is composed of functional games. However, if the functional games are 'bad', integrating rubbish together ends up with rubbish. Also, game design is a personal task and the author's bias will always be incorporated in an education game. Unfortunately, there exists little help in game design procedure especially in engineering business management area. Game design philosopy is always a 'black box' and game design in EBM is a 'black box' within a 'black box'. A set of general guideline should be provided to the game authors in EBM to minimise the risk of obtaining a bad education simulation game.

5.0 Objectives

After all, the goal of this research is to provide an effective integrated teaching environment in engineering business management and simulation game has been identified as the teaching medium. Instead of developing one good simulation game which cannot solve the problem as hundreds of games are required for different education objectives, generic approach of constructing education simulation games in EBM is investigated. The approach is broken down into two levels: micro and macro. The objectives of this research are:

1. In micro level, the game design philosophy of an education simulation game in engineering business management is investigated.

2. In macro level, a standard platform for all games in EBM is established so that functional games built on this platform can be integrated together to form total integrated games which can be further integrated together to form a supply chain network.

In short, the title of the research is 'On the Design and Integration of Education Simulation Games in Engineering Business Management'.

6.0 Thesis Plan

Chapter 2

Simulation game - An Experiential Learning Method

Game is chosen as the teaching medium because game is an experiential learning method which is proved to be one of the most effective teaching media for integrated knowledge. The effectiveness of games and other related teaching media like case studies are also examined from an education point of view.

Chapter 3

History and Usage of Simulation Games in Education

The history and usage of simulation games in education are studied in detail. Also, the characteristics, the benefits and sources of simulation games, etc. are given in this chapter.

Chapter 4

Game Design Philosophy of an Education Simulation Game in Engineering Business Management

The existing game design approach are reviewed: systematic approach and inspirational approach. However, these game approaches are for social science games only. Although, some basic principles are also applicable in the game design in EBM, three main areas are missing:

1) Determination of appropriate level of features and formats (Chapter 5)

1) Identification and rectification of weaknesses commonly found in simulation games in EBM (Chapter 6)

3) Evaluation - Users' and Administrators' opinions (Chapter 7)

Chapter 5:

Determination of Appropriate Level of Features and Formats

Formats refer to the physical environment when running a game such as 'location', 'number of teams', 'team size' and 'with/without administrators', etc. Features, on the other hand, form a general structure or framework of a game such as 'degree of realism', 'level of complexities' and 'degree of interaction'. These no doubt will affect the effectiveness of an education simulation game.

Chapter 6

Identification and Rectification of Weaknesses Commonly Found in Engineering Business Management Games.

Simulation games have a lot of advantages but also possess a list of drawbacks. During the review of the existing simulation games, considerable weaknesses have been discovered. However, most literature only list out the benefits of using games for teaching and overlook the drawbacks. A game can never be good if the weaknesses are not rectified. In this chapter, the weaknesses of simulation games are identified and methods of rectification are suggested.

Chapter 7

Evaluation of the Game Design- Users' and Administrators' Opinion

If the users and administrators do not like the game for whatever reasons, a good simulation game and a bad one does not make any difference. During 92-93, three surveys were carried out to collect the users' and administrators' opinion on using simulation games in education and their preference on the appropriate levels of the formats and features of education simulationg games in EBM. This chapter is the evaluation and confirmation of the previous findings on game design.

Chapter 8

Integration of Education Simulation Games

In order to make use of the functional games as part of an integrated game, a standard interface is established. With this standard interface, different engineering business structures can be formed. As the size of the database (of games) increases, the choice of games is more and since the number of functional games and their relationships in the network are totally flexible, the number of different structures that can be formed is infinite. This principle is a modification of Zeiger's hierarchical modelling principle.

Chapter 9

Implementation and Evaluation

In order to demonstrate the generic approach, two functional games - a production game and a marketing games are produced. The design of the game elements follows the game design approach at micro level. At macro level, these two games can be integrated into different structure for different education purposes. This principle is further confirmed by using Dynamic Data Exchange under Windows environment to build the standard interface and a base unit becomes a machine instead of a functional game.

Chapter 10

Conclusion

A lot of existing EBM games can be modified to run on the standard platform with only minimum effort. Further study is recommended on using Windows environment and its multi-tasking ability to communicate between different functions and modules. Also, expert system may be developed on game design to help the game authors to build the framework of a game by inputting the education objectives into the system.

CHAPTER 2

SIMULATION GAME - AN EXPERIENTIAL LEARNING METHOD

2.0 Common Teaching Media

More than 10 teaching media are commonly used in tertiary education (Joyce, Weil and Showers, 1992; Unwin, 1969; Ely, D.P., Gerach, V.S. 1980; Craig, R.L. 1976; Megarry, J. 1979). They are:

- Lecture
- Demonstrations
- Group Discussions
- Tutorial
- Role Playing
- Case Studies
- Games
- Brain Storming
- Programmed Learning
- Computer Assisted Instruction
- Independent Study
- Leaderless groups
- Sensitivity training
- Project work

Project work, games and case studies are always chosen by university lecturers in teaching of engineering business management. They are not random choices from numerous existing techniques. They are a type of 'experiential learning methods' and in particular, falling into the category of 'learning by doing'.

2.1 Experiential Learning

Experiential learning is a broad church encompassing a number of different traditions. Different types of learning experiences were categorised and summarised by Henry 1989 as shown in the following figure:

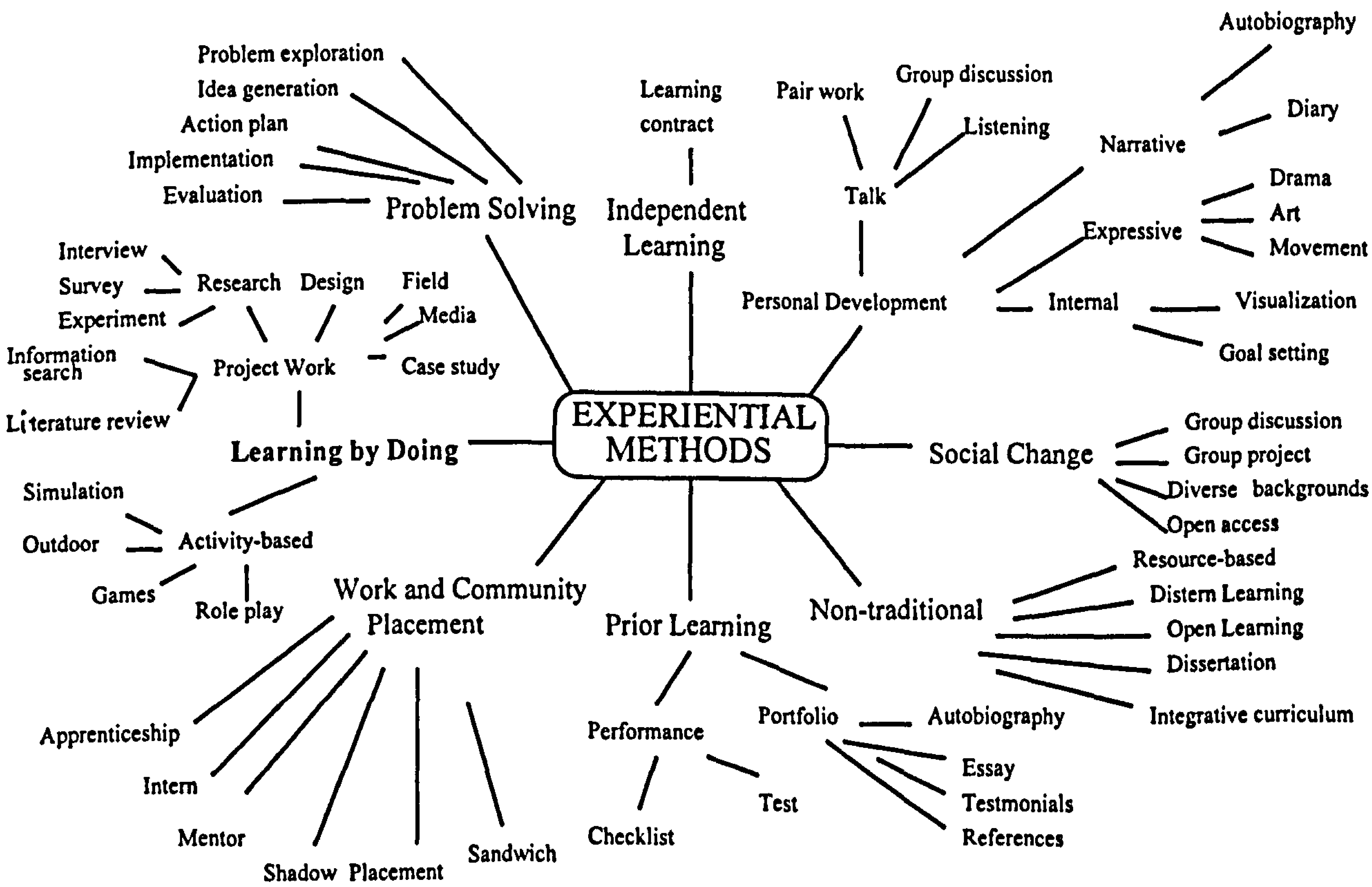


Fig. 2.1 Experiential learning methods

(Extracted from Henry in Weil & McGill (1989))

There is no argument that most university courses are still predominantly taught by 'lecture' (Expository Approach). The significance and the potential of 'Experiential Learning' (Discovery Approach) has not been fully recognised until relatively recently. (Weil, McGill 1989). Although experiential, or experience-based, learning can be regarded as the earliest approach to learning for the human race, it had been mainly overlooked in the formal education system. In fact, it is common to find a university course purely delivered by lectures but it is hard to locate any course that purely adapts discovery approach.

Fortunately, increasing numbers of teachers and practitioners of all kinds are realising that the polarity between the intellectual and the practical is an absurdity which can no longer be supported and that we can only progress if we accept that thinking and action are entirely complementary. (Weil & McGill 1989). This is exactly the difference between knowing all the engineering business theories and knowing how to run an engineering business firm.

2.2 Models of Instruction

Experiential Learning or Discovery learning was first mentioned by Jerome Bruner in 1966. Bruner argued that the teacher's role must be to create situations in which students can learn on their own, rather than to provide pre-packaged information to students. Bruner states:

"We teach a subject not to produce little living libraries on that subject, but rather to get a student to think ...... for himself, to consider matters as an historian does, to take part in the process of knowledge-getting. Knowing is a process, not a product." (1966, p.72)

Bruner suggested that students should learn through their own active involvement with concepts and principles, that they should be encouraged to have experiences and conduct experiments that permit them to discover principles for themselves.

On the contrary, David Ausubel (1968) argued that students do not always know what is important or relevant, and that many students need external motivation to do the cognitive work and it is necessary to learn what is taught in school. Ausubel described an alternative model of instruction, called **reception learning (Expository Approach).** Reception theorists suggest that the job of the teacher is to structure the learning situation, to select materials that are appropriate for students, and then present them in well-organised lessons that progress from general ideas to specific details. At the core of Ausubel's approach is what he called expository teaching, which is teacher-planned systematic instruction on meaningful information. Expository teaching consists of three principle stages of lesson presentation: 'Presentation of

'Advance Organiser', 'Presentation of Learning Task or Material', 'Strengthening Cognitive Organisation'.

Fenton 1967, however, realised that both approaches are too extreme and few teachers are purely 'expository' or purely 'discovery' oriented. He interpreted two different approaches as two extreme poles on a teaching continuum and the teacher must decide which point on the continuum will help to reach the education objectives.

A TEACHING CONTINUUM

Exposition ———————————————— Discover

Fig. 2.2 A Teaching Continuum

In fact, pure discover approach may not be suitable in education as students always need some sort of guidance in their learning process to make sure that they are on the right track. For projects, there are always project supervisors available and for simulation games, the most important session is the after-game session in which the game administrator discusses with the students on the problems that they have faced during the game.

2.3 Definition of Experiential Learning

The most popular theories of experiential learning was offered by Kolb (1975) who presented four stages of experiential learning -- concrete experience, observation and reflection, abstract conceptualisation and generalisation, and active experimentation.

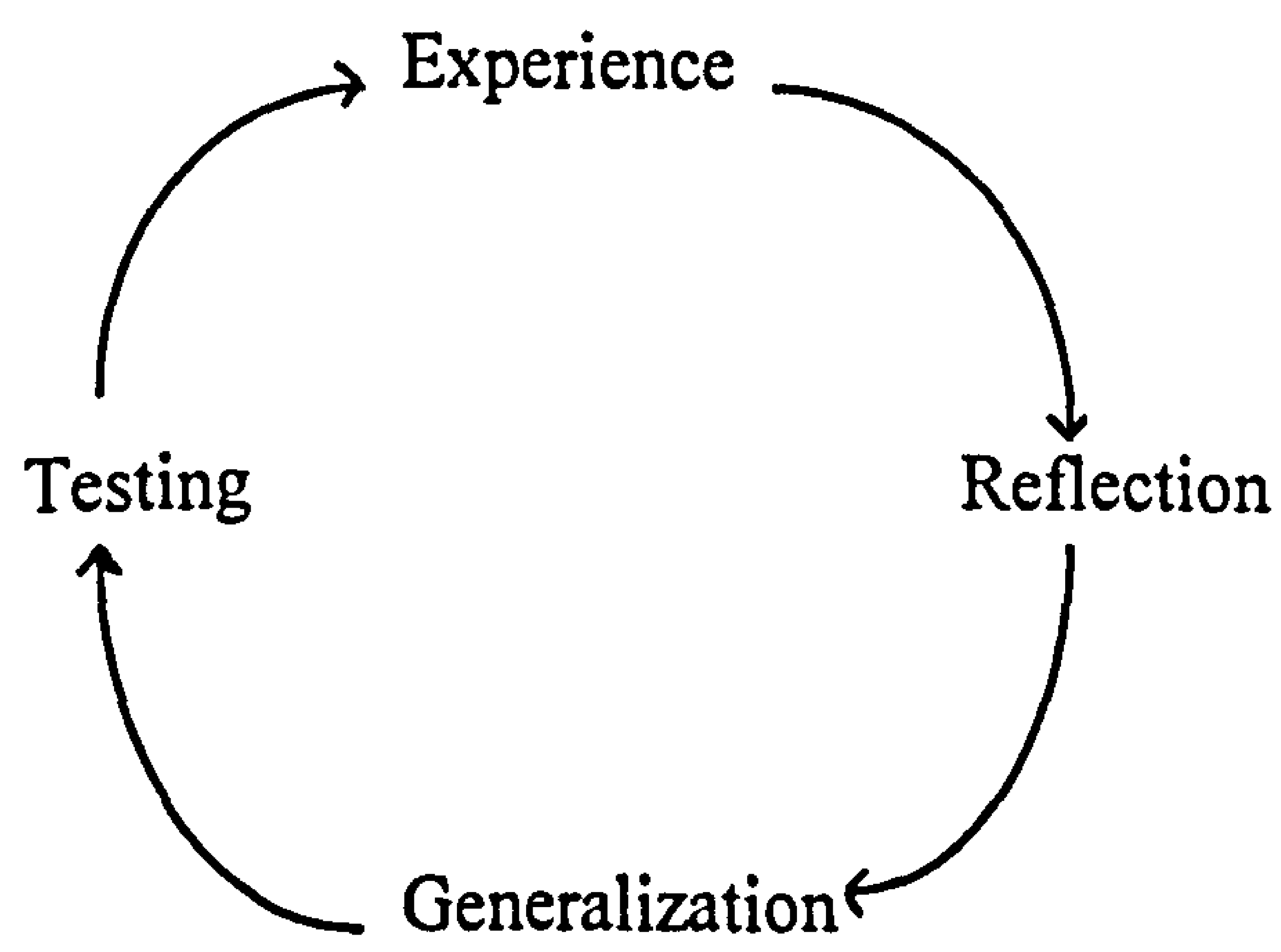


Fig. 2.3 The Kolb Learning Cycle

(Kolb and Fry, 1975)

However, Kolb's model has attracted a lot of criticism as too general and abstract and could be used to describe any learning process. Further confusion arises as proponents of experiential learning often use different keywords to summarise their understanding of the Kolb cycle and experiential learning. Although several educators had done research to modify Kolb learning cycles. (Gibbs, 1987; Boud and Pascoe, 1978; Boydell, 1976; Kilty, 1982), nevertheless, Kolb's definition on experiential learning is often the main or the only theorist quoted in papers on experiential learning.

Another way of classifying experiential learning is not to describe the sequence of learning that is entailed, but to pull out the dimensions that are the characteristics of this particular approach. Boud and Pascoe (1978) suggested students involvement, learner control and the correspondence of the learning task to activities outside the classroom are central to any experiential learning activity. Although there is no single definition for experiential learning, both the experiential theorist and educational practitioner seem to agree on what experiential learning is not. It is definitely not the mere memorising of abstract theoretical knowledge, especially if taught by traditional formal methods of instruction such as lecturing and reading from books. That is, people seem to agree that experiential learning is about ensuring that people can 'do' rather than merely 'know', but differ in their emphasis on what skills enable the desire quality of 'do-ability'.

2.3.1 Key Dimensions of Experiential Learning

Henry further provided a model (shown below) on Experiential Learning Approach to show the key dimensions on which these different approaches are focused. The horizontal axis stresses applications in the 'real world' (especially work and community) versus a concern with meaning. The vertical axis polarises a focus on student autonomy and choice versus the needs of the environment in which the student finds herself.(Henry 1989)

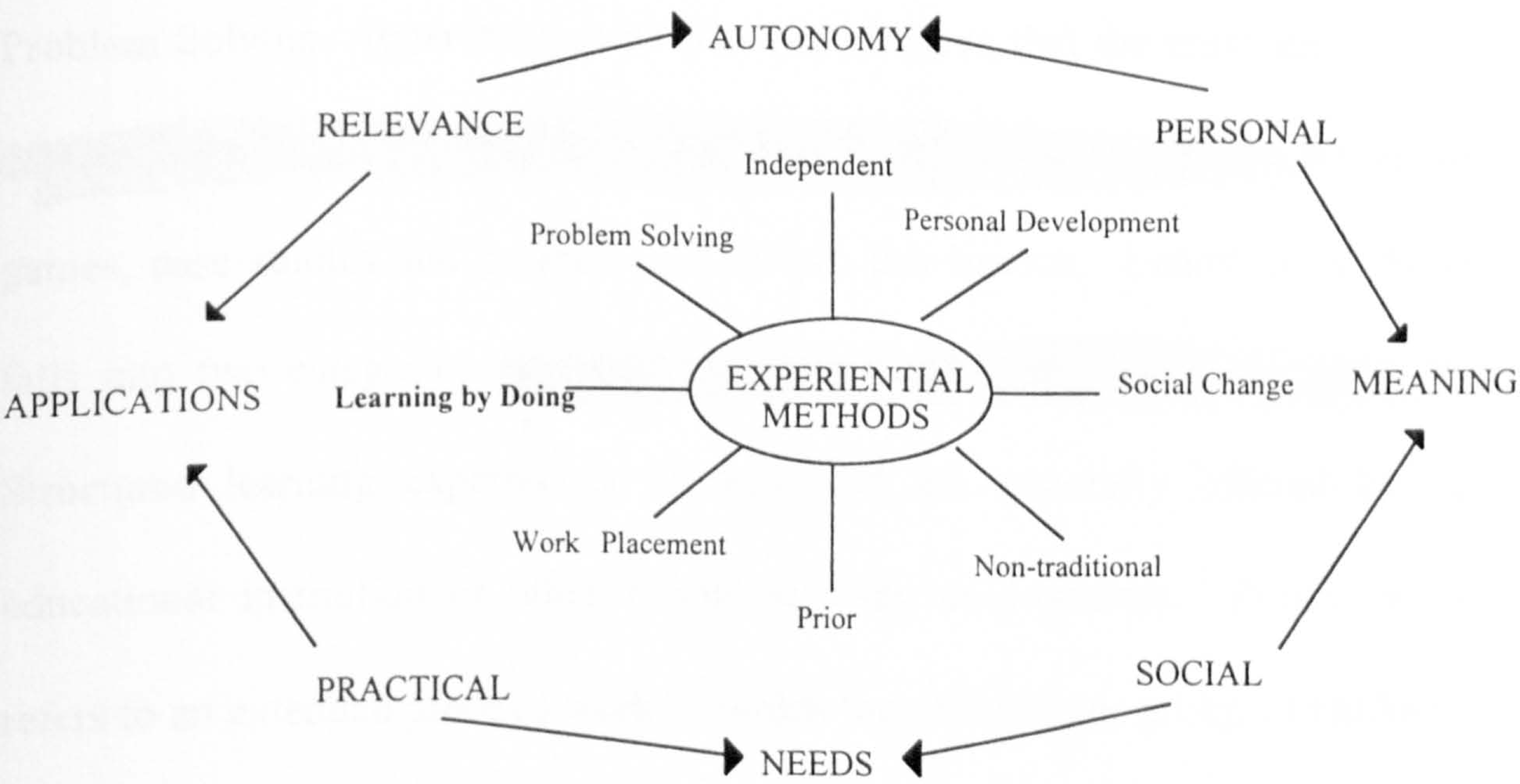


Fig. 2.4 Experiential Learning Approaches

(Extracted from Henry in Weil & McGill)

Games and case studies are simulation of business and engineering environments in which the participants are allowed to practice their management theories and ideas obtained from lectures and textbooks. They provide laboratories for the students to manage companies without taking the risks of the real life. As suggested in the above model, games and case studies are applications in the real world.

2.3.2 Learning by Doing

Experiential learning methods can be categorised into eight branches: Independent Learning, Personal Development, Social Change, Non-traditional, Prior Learning, Work and Community Placement, Learning by Doing, and Problem Solving. Apparently, most educators believe that the most appropriate way of learning engineering business management is 'Learning by doing' as games, case studies and projects all fall into this branch. Learning by doing falls into two categories: *project-based* learning and *activity-based* learning. Structured learning experiences of this kind are generally offered by an educational institution or other formal learning environment. Project work refers to an extended piece of work in which the students (or group of students) is required to select a topic, collect relevant information and organise this material into a presentation. This term covers diverse activities such as research, an information search or design. Research projects may be based around quality of quantitative methods involving surveys, interviews or experiments. Information searches include literature reviews, and work with primary documents or secondary sources. Design projects may involve building a tangible product or just producing plans which would enable one to do so (Henry 1977). The case study approach is an allied method. Activity-based learning includes practicals, simulations, games, role-play or expressive approaches like drama, art, and imaginative activities (e.g. visualisation).

2.4 Learning Process

Learning is a complicated process. Learning is a process of continual adaptation to environment, assimilation of new information, and accommodation of new knowledge to fit the old knowledge.

There is no single definition of learning, Hamachek, 1990 defines that "*learning is a process by which behaviour is either modified or changed through experience or training.*" and Slavin's (1991) definition of learning is: "*Learning involves the acquisition of abilities that are not innate. It depends on experience, part of which is feedback from the environment.*"

Whether an experiential learning method is appropriate in a subject area depends on what type of knowledge that instructor would like to conduct to the students. However, there is no unique classification on knowledge. Mayer (1985) classified knowledge into 'Linguistic knowledge', 'Factual knowledge', 'Schema knowledge', 'Strategic knowledge' and 'Algorithmic knowledge'.

However, most 'knowledge-based system' educators classify knowledge into 'deep' and 'surface' knowledge.

2.4.1 Deep and Surface Knowledge

Deep knowledge is abstracted from books regarding concepts, principles and techniques and is applicable to a large number of situations. Surface knowledge is highly job oriented. It is usually obtained from actually doing the task and is accumulated from experience. Surface knowledge is difficult to obtain through learning from books. For example, a person may be very knowledgeable in driving a car and he knows every part of the highway code. However, he may not be a skillful driver. Because despite his knowledge about car, the only way of learning how to drive is to drive a car. This is a major difference between skills and (factual) knowledge. (it must be noted that 'skill' is also a kind of knowledge.) In this example, knowledge about car and highway code is 'deep' knowledge but driving a car skilfully is 'surface' knowledge. Similarly, a manager may understand every business principles in the textbook but it does not mean that he can make a rational decision. On the contrary, an experience manager may make a decision according to his experience although he may not be able to explain his decision with business theories.

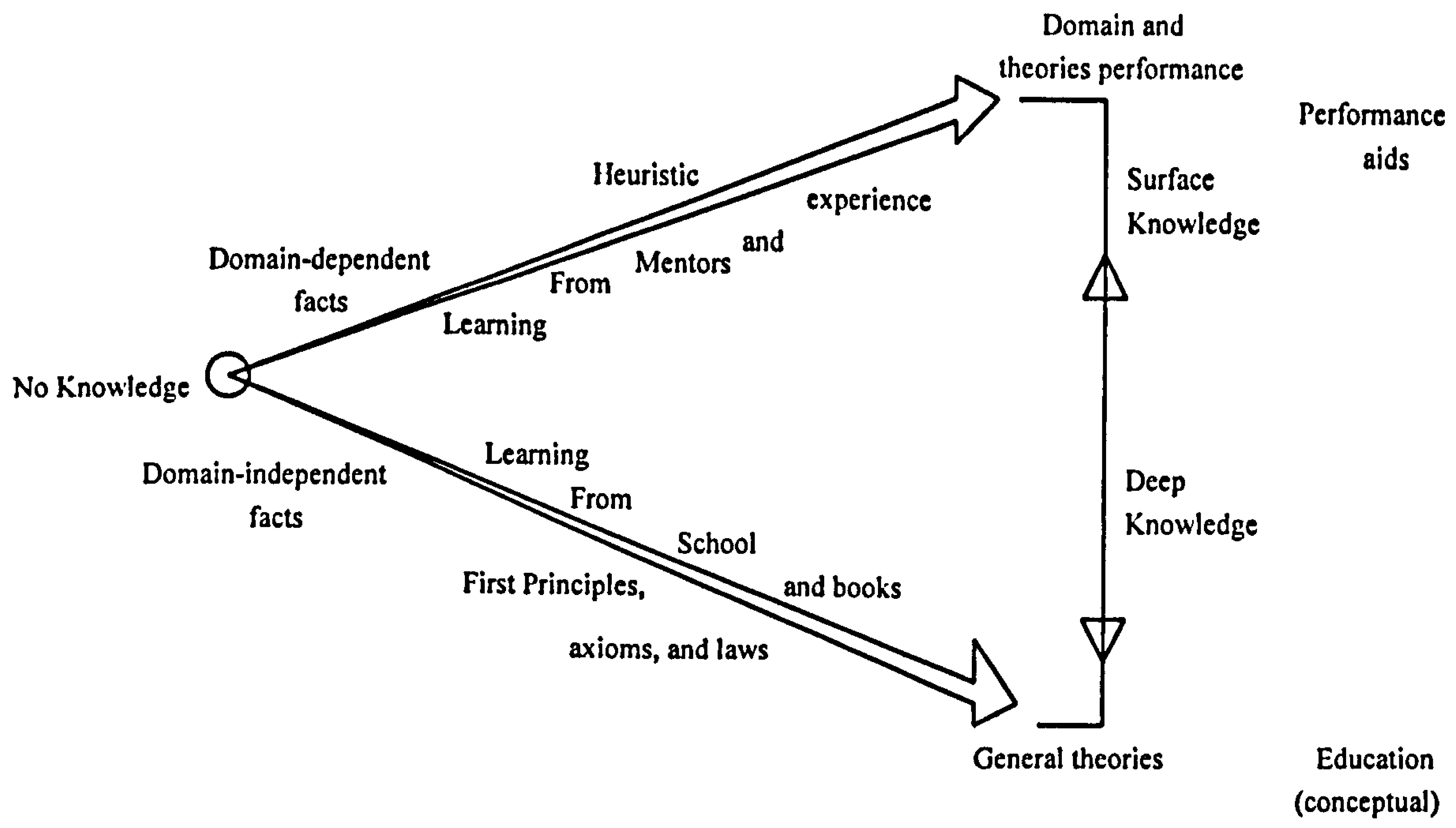


Fig. 2.5 Varieties of knowledge and their relationships to instructional strategies

(extracted from Horman, 1983 p.236)

2.5 Games and Case Studies

Although more than ten different teaching methods are attached to the 'Learning by Doing' strategy: Media, Field, Case Study, Design, Research (interview, survey, experiment) , Literature review, Information search, Simulation, Games, Role play and Outdoor, due to the physical and resource constraints of a classroom environment in a university and the nature of the subjects in manufacturing management, most of the methods are not viable. The only two feasible choices are game and case study.

2.5.1 Simulation Games

Although simulation games can cover almost any subject in the world and the application areas range from survival in nuclear war to cattle breeding (Somogyi & Kisimre, 1983; Hartley, Johnson & Fitzsimons 1979). Due to the scope of this project, only business games are discussed. A game, which simulates the business and engineering environment for education and training purpose, is usually called a 'business game' (Elgood 1993; Greenblat 1988). It contains a model which is designed to simulate a certain portion of an organisation and its environment. It simulates certain portion of the real world for participants to play a simulated role in the model. Business games attempt to capture the main features and dynamics of key situations though ignoring the lower orders of complexity of 'real' business situations.

Many definitions of business games exist and two of them are offered below:

"A business game is a contrived situation which embeds players in a simulated business environment, where they must make management-type decisions from time to time, and their choices at one time generally affect the environmental conditions under which the subsequent decisions must be made." (Shubik 1975)

"Business games are case studies with feedback and a time dimension added." (Carson 1969)

In a typical business game, a business/production scenario is first set up by a game administrator and the history or the background of a fictitious company is provided to the participants who usually work in teams. The participants assume appointed management positions and make appropriate decisions which are then handed to the game administrator for processing. In a non-interactive game, the feedback (results) depends purely on the pre-set scenario, in an interactive game, the feedback is based on the pre-set scenario as well as the decisions made by other groups. Then, the participants analyse the results and come up with a new set of decisions for the next round which is usually a quarter or a year later. This process may continue for any number of decision periods. In general, business game is a sequential decision-making exercise structured around a model of a business operation, in which participants assume the role of managing the simulated operation.

2.5.2 Case Studies

Case studies are often used as a way of speeding up the learning process and focusing attention on the specific issues. Case studies are often descriptions of specific business situations which have occurred in real life. Uses of this techniques helps the learner to see that a problem has more than one solution. It can also helps develop supervisors' and managers' analytical and problem solving skills. Some educators believe that case studies can even help to develop social skills and group identity. (Easton 1992).

The Harvard Business School is a pioneer in the use of case studies for presenting business and economics activities in the classroom. Nowadays, almost every business school uses case studies for similar purpose (Robinson, 1985). In fact, case studies which are based on real world examples, were once thought to bridge the gap between 'surface' and 'deep' knowledge.

2.5.2.2 Drawbacks of Case Study

However, the practical application of case studies can present many problems to students and teachers. There is:

"...... increasing awareness that case studies are inefficient teaching vehicles. The participants both students and teachers have been at once excited and frustrated: excited over the possibilities and frustrated because they are unsure how the learning process should be managed and have too little time for training or experimentation."(Easton 1992 p.ix)

Cases are normally written, but this is time consuming and can lead to all sorts of errors. Inadventently long or short case analysis are not favourable as Easton described:

"... Short cases prove to be very taxing. ..All of the significant detail has been stripped away....... long cases provide challenges in organising material rather than practice in solving complex problem." (Easton p.2-3)

Moreover, the student should be doing enough case study to enable learning. However, time constraints in the classroom often prevent the instructor from providing too many similar cases to the students. Another problem with case studies is a lack of interaction. The students are often unable to measure their effectiveness in decision making. The only feedback available is from the case instructor and that is subject to personal style of teaching.

"One of the major criticism of the case study method has always been that the student could never measure the effect of his decision. He has only the subjective, qualitative judgement of the instructor and his fellow students to measure the correctness of his answers to problems" (Carson 1969, pp 35 -45)

Also, it may not be possible to use a case more than once. The second group participants may not judge the case on its own merits but to obtain the suggested solution from the first group.

The list may be lengthy and Easton 1992 has listed out ten drawbacks of case studies. Although the above comments for case studies may be true , it is also possible to develop a similar list on disadvantages of simulation game used as a teaching medium. Instead of listing out drawbacks of simulation games which may lead us to no where, a comparison of 'case study versus gaming' and the effectiveness of simulation game in education are further studied.

2.5.3 Simulation Game Vs Case Studies

Early work on the evaluation of games mainly involved the sampling of student opinions after participation in these games, and these include Jackson, 1959, Dill and Dopplet, 1963, Dobles and Zimmerman, 1966. The next research phase involved the use of controlled 'before and after' evaluation examinations, which compared the use of games to the use of cases for teaching the same concepts.

McKenney, 1963, used a series of cases on one group and a combination of game-and-case on another, and concluded that those who played the game understood planning concepts better. However, his conclusion could not be supported statistically. In another evaluation attempt, Raia, 1966, found that the complexity of a game was not related to its benefits, and that students who participated in the game scored higher in all final examinations with the exception of the business case analysis examination. In yet another evaluation, Moore 1967, used games and cases in an introduction to a production management course, and concluded that "*The findings did not support a general proposition that games are more effective than case method from the standard point of learning*". In another attempt, Wolfe and Guth, 1975 concluded that "*both teaching aids were effective, but it appears that the gaming environment was the more* *effective of the two"*. This, to some extent, appears to contradict Moore's findings. However, in a later attempt, Partridge and Sculli, 1979 supported the use of games in preference to case when teaching basic business policy but he also pointed out that since the experiment was undertaken with no involvement of teachers, it was very likely that a case course where students were motivated by teacher involvement would be more valuable than a game run in the same way as was done in the experiment. He emphasised that the cases and games should be seen as merely tools of the teaching trade but their effectiveness could be considerably increased by the enthusiasm of teachers using them.

Nowadays, educators admit that 'good' games and case studies are comparatively efficient in teaching and the success of a teaching method depends on a lot of other factors such as the involvement of the instructor, motivation of the students and the relevancy of the case study/game to the subject.

2.6 Effectiveness of Games in Teaching

Besides the above specific studies between case studies and games, several studies have been conducted to evaluate the effectiveness of games in teaching (or training) compared with traditional methods in different fields, especially in business oriented areas (Robinson, 1985; Klein 1984; Wolfe 1975; Horn, Cleaves 1980). The most common ways to measure the effectiveness of games are to compare the results of the before-game examination and the after-game examination or to compare the examination results of two groups of students in which one group is taught exclusively with games and the other group is taught with lectures or case studies. However, none of them gives definite conclusion. It is of little surprise that no evaluation report on using games in the teaching of engineering business subjects could be found. Although there is no hard evidence to prove the superiority of game against the other means of teaching methods, games are definitely more enjoyable by students and most of them prefer games to the other communication media. (Greenblat, 1988). The effectiveness of games in teaching is still argued since it is extremely difficult to measure the knowledge acquired in a game. This may be because the success of an educational game depends on many factors such as the contents of the game, backgrounds of the participants, the role of the administrators and the time available for each decision. In addition, the quality of a game depends largely on the quality of the simulation in representing the behaviour of the real world and the way the game administrator helps the participants applying the knowledge to real world.

Although there is no proved result on the effectiveness of games in education and teaching, successful examples can be found everywhere. For example, LOOKING GLASS is one of the oldest and most widely used simulation. It was developed in a three-year $300,000 project beginning in 1976 by social scientists at the Centre for Creative Leadership, a non-profit think tank in Greensboro, North Carolina. LOOKING GLASS was originally intended to be a research tool, but its creators added debriefing sessions after managers in test runs clamoured to know how well they had performed and how to improve their business skills. It thus became a training device. Larry Kahn, the training director of SmithKline Beckman, has tried to gauge Looking Glass's effects on managers. He surveyed senior managers at his company, asking whether subordinates who went through Looking Glass were better managers six months after they returned to their jobs. Nearly 80% of the bosses said yes, but, says Kahn, *"I can't prove it to myself. It gives me numbers to show my own boss, but evaluating a training program is like trying to nail Jell-O to a wall."* (Peter 1984)

In fact, the research into the general effectiveness of the gaming method suffers from inherent weaknesses. The first of these is that any particular game is different from every other game, so that the investigations into the effectiveness of 'the gaming method' simply lost in an infinity of differences. Educators end up in a position that is similar to attempting to average the taste of apples, oranges, pears, bananas and so on. Related to this is the lack of sound measurement techniques in the realms of attitudes. However, the fact that quantification in these areas is currently very difficult does not mean that the gaming method is not effective. It just means that at present we have no tools to rigorously quantify the things that are regularly reported by people who use the method, but then, teaching has always been much more an art than a science.

Moreover, where the evidence does not reveal benefits of gaming techniques over other modes of teaching, neither does it show the reverse; that is, those taught with games do not prove to have learnt less than those taught in traditional ways. Though the arguments continue, gaming has been accepted as a comparatively effective teaching media. Unfortunately, although physical simulation is used extensively in manufacturing industry for training or problem-solving, the using of gaming is often overlooked in the teaching of engineering business courses at universities.

Although there is no definite conclusion on the effectiveness of games in teaching, research does reveal the advantages of mixing traditional methods and games in teaching. In 1970, Kitchen performed a study to assess the effectiveness of using games at British European Airways, designed to improve the skills necessary in handling the extreme complex business of managing aircraft turnaround at an international airport. He concluded that the group which was taught by game plus lecture yielded the best result followed closely by the group that was taught by games only. The group taught by lectures only did not perform satisfactorily in the examination. This reinforces the view that games are comparatively effective as other teaching media if it is not better. Most importantly, a mix of game and traditional teaching media may yield a better outcome.

In short , games can be made to embody all phases of Kolb's learning cycle (Fripp 1984). It is precisely in this respect that they gain their added reality over case studies. The participants of case studies do not have the opportunity of living the consequences of their decisions. Using the terminology of Kolb's model, their focus is purely on reflective observation and abstract conceptualisation.

Simulation games, on the other hand, offer very real opportunity for the full range of learning experiences to occur. Participants have to make a variety of decisions and actually put them in practice. These decisions affect the future of the company. In a very real sense that they have to live with the results of their own actions.

In addition, case studies have been used in teaching management subjects in the last few decades but the potential benefits of using simulation games have not been fully revealed. Since games are more enjoyable by students plus there are more than thousands of case studies already existed but 'good' simulation games are relatively scarce, research on game design and integration is carried out in this thesis.

CHAPTER 3

HISTORY AND USAGE OF SIMULATION GAMES IN EDUCATION

3.0 Simulation

Simulation originates from the application of the principles of 'cybernetics', a branch of psychology. There is a general lack of consistency in the literature as to what simulation is. Depending on the orientation of the author simulation has been described in many different and sometimes contrasting ways. Pritsker 1979 has collected twenty-two definitions of simulation from different authors. Although only five of them are listed here, this is not to say that these definitions are more precise than the others.

- *This is precisely what simulation is: the representation of the dynamic behaviour of* a *system by moving it from state to state in accordance with well-defined operating rules (Pritsker and Pegden 1979)*

- *Simulation denotes a computer-based numerical technique for the experimental study of a stochastic or deterministic process over time (Scheruben, 1978)*

- *Simulation is the technique of constructing and running a model of a real system in order to study the behaviour of that system, without disrupting the environment of the real system (Bobillier el at, 1976)*

- *The phrase 'modelling and simulation' designates the complex of activities associated with constructing models of real world systems and simulating them on a computer. In particular, modelling deals primarily with the relationships between real systems and models; simulation refers primarily to the relationships between computers and models (Zeigler, 1976)*

- *Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies (within the limit imposed by a criterion or set of criteria) for the operating system (Shannon, 1975)*

From the above definitions, it can be said that simulation modelling is a means of carrying out experiments specifically designed to study the behaviour of a 'real world' system as it operates over time by gathering and structuring of data of its various components in such a way that the values of parameters, the initial values of the variables, and their interrelations with each other are formalised. These experiments are usually manipulated on the computer since it may involve a large amount of data and lengthy processing time.

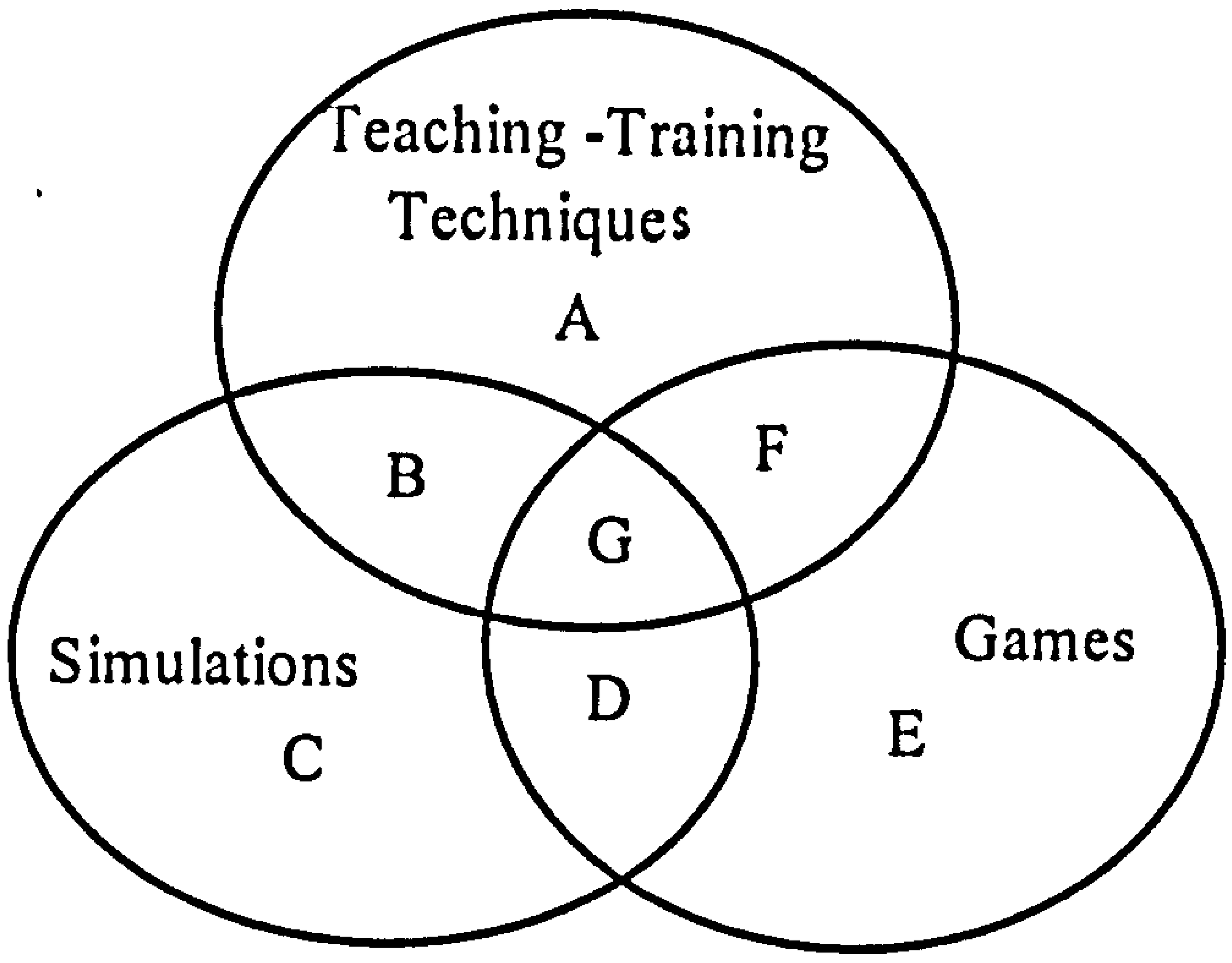
Simulation is in fact one of the most powerful analysis tools for designing and analysing complex systems. However, good computer simulations of complex systems are usually not cheap because a lot of training is required on model building, statistical analysis and familiarisation with simulation languages, etc.

Simulation is a famous means for education and training too. The best-known training simulator is perhaps the flight simulators used by most of the major airlines to train pilots. Computer control has brought this type of simulator to a point where it is now possible to train an airline pilot to a high state of proficiency without the pilot leaving the ground. In Hong Kong, the Mass

Transit Railway Corporation's training employs simulators for training drivers and traffic controllers.

In 'The Right Stuff' (Butterworth 1989), a book describing the early development in America's manned space programme, the author stressed that training simulators were used extensively to train the surprise out of the missions long before the astronauts were allowed near a space craft. They repeated the procedures, standards and emergencies, thousand of times in simulators until they reacted instinctively to any situation. The instructor knew that an astronaut was fully trained when he responded to an emergency with no increase in the physiological indicators of stress.

Another class of simulation in education and training is simulation games. The relation between simulation and game is a subtle one. Greenblat describes the relationship between game, simulation and teaching-training techniques with the following diagram:



A: Teaching -Training Techniques
B: Simulations used for teaching and training
C: Simulations
D: Simulation games
E: Games
F: Games used for teaching and training
G: Simulation games used for teaching and training

Fig. 3.1 Simulation, Games and Teaching Techniques
(Extracted from Greenblat 1988)

It is clear from the above diagram that not all games are simulations and not all simulations are games either. Simulation can be used as a training device, like the flight simulator and it is not a game. Also there are several games used for training purpose like 'Equations' and 'On-wards' and they are not simulations. Category G is the simulation game employed for teaching and training . Some researchers suggest that a simulation is a real time activity and a game compresses time. Another defines competition as an essential element in a game but not in a simulation. (Ellington, Addinall & Percival, 1982). Jones, on the other hand, considers the main difference between games and simulations can be observed in the thoughts and the behaviour of the participants. For example, in running a business game, if the participants think they are only playing a game, then it is a game. On the contrary, if the participants are really putting themselves in the management positions and try to run a company, then the business game is a simulation. Most game or simulation authors spent pages and pages to make distinctions between them.(Jones 1985; Greenblat, 1988) They assign different characteristics on games and simulations, sometimes even in role-play as well. For examples, Coleman 1967 imposed six characteristics to games and five characteristics to simulation games and three rules to a game model.

This type of research is a total waste of time as arguing on the name is meaningless and the effectiveness of the teaching media will not be improved. Also, whether a particular teaching method is a simulation, a role play or a simulation-game, a simulated case studies, a game used as a case study or a simulation-game used as a case study, at the end of the day, there always remain room for disagreement. More effort and considerations should be placed on the objectives of the teaching medium, the ways to accomplish the objectives, the design process, the implementation method and a lot of other extras that should be considered but definitely not the name. In this paper, the name, 'simulation game' and 'business game' are used interchangably.

3.1 History of Simulation Games

Simulation games have been used widely in education and training for a long time. Everyone plays games from kindergarten upward. There is no hard evidence relating to where and when games were invented (Cohen, Rhenman 1961). Many people believe that the first use of a game to represent real world problems began with the Chinese game of Wei-Hai (Japanese 'Go') and the game of chess is the oldest form of war game (Rediffusion 1986; Sayre 1961). Business decision simulation exercises may be considered as an outgrowth of military war gaming (Greenlaw, Herron & Rawdon 1962; Meier, Newell & Pazer 1969; Ginter, Rucks 1983). Perhaps the best known and ambitious war games were those at the Total War Research institute and the Naval War College of Japan as part of the Japanese preparations for the Second World War. War games have also long been used by the British and Americans to test battle strategies and to prepare troops. The available simulation games in the market range from very simple to very complex; computerised to non-computerised; single participant to multiple participants; interactive to non-interactive. They can cover all major business decisions areas (total management game) or be specific in a single decision area (functional game) etc.

The applications of gaming can be broadly subdivided into six categories: teaching, training, operation, research, therapy, or entertainment (Shubik 1975). Whatever the form of game involved, the players play simulated roles. Undoubtedly, the games for entertainment have the longest history and the largest variety, from 'Tic-Tac-Toe' and 'Monopoly' to interactive animated computer adventure games found nowadays. However, only the use of simulation games for teaching and training is studied in detail in this paper.

Nevertheless, some findings from research games are helpful in designing and building of a good simulation game. Examples of research using simulation games includes the areas of 'International Relations', 'Organisational Research', 'Leadership and Team Effectiveness', 'Introduction of New Technology' and 'Management/Union Communication' (Fripp 1984; Guetzkow, Jensen 1966; Hogarth, Makridakis 1981; Rowland, Gardner 1973; Sloman 1977; Mintzberg, Raisinghani & Theoret 1976). In particular, when gaming is used seriously to determine the optimum solutions for strategies and to determine optimum structure for systems, this may be termed 'operational gaming' Thomas and Deemer define it as " *the serious use of playing as a primary device to formulate a game, to solve a game, or to impart something of the solution of a game*" (Thomas, Deemer 1957). In fact, the oldest known form of gaming, the military gaming has long had this purpose.

3.2 Types of Simulation Games in Engineering Business Management

3.2.1 Functional Games

In engineering business, there are generally two main categories of simulation games: functional games and total management games.

Functional games cover a particular function of a company, like shop floor scheduling, marketing or finance. The participants make decisions to optimise the production or sales without concern the effect of their decisions on the other departments. Since only a specific area is covered, the game usually involves a greater depth of detail.

Functional management games are appropriate for training lower and middle levels of managers to make better decisions on their relevant functional activities. They (except marketing games) are usually non-interactive between teams which compete against some standards such as cost and time.

3.2.2 Total Management Games

Total management game gives an overview of the whole company operations which usually comprise of three areas: production, finance and marketing. The decision made in one area will affect operations in the other areas. Since the game covers several areas, the number of decisions or the level of detail in each area will generally be less than in a functional game.

Total management games are mostly used for training of managers in upper and middle levels. The objectives of each team are either set by the game administrator or the participants themselves. They are usually organisational objectives on 'profit', 'return on investment', 'market share' and 'total asset', etc. Since functional areas in a total management game are inter-related, the participants have to well balance the decisions made in each functional area to achieve a 'global optimisation'. An integrated approach is required. Therefore, a total management game is more appropriate to convey the integrative concept to the participants.

3.3 Use of Simulation Games in Education

The traditional educational approach using exclusively readings and lectures provides an incomplete coverage of the learning process for action-oriented disciplines. As a result, only a fraction of the knowledge acquired has an effective impact on performance. Most of the remaining knowledge is not translated into action and is progressively forgotten. At the other extreme, one can try to learn by doing, without having first developed an adequate knowledge base. While this approach is probably perceived to be effective by those who have survived the experience, it is almost certainly the most expensive one and clearly a luxury in today's highly competitive world.

In between these two extreme educational approaches, two methods have been increasingly used in action-oriented disciplines: case studies and simulation games. They have attempted to bridge the gap between knowledge and action.

Learning Stage	Method	Approach
KNOWLEDGE	Readings Lectures	Traditional
↓	Case Studies Simulation games	Action-oriented
ACTION	Doing	Practice

Fig. 3.2 Educational Approach

(Extracted from Larreche 1987)

The relative effectiveness of games and case studies has been discussed in the previous chapter although the overall effectiveness of games in education is a controversial issue. Educators fall into different categories, from vigorous opponents to strong advocates of the simulation approach. The opponents doubt that learning of concepts can actually take place in the context of a simulation in which they believe the 'gaming' aspects dominate. Advocates are convinced that simulations provide a valuable environment for the dynamic learning and application of concepts (Lareche 1987).

This divergence of opinion is due to the different expectation of educators when using simulation as a teaching tool in business. Also, 'good' simulation games are hard to recognize. Good simulation games are simulations that can successfully represent realistic environments and demonstrate relevant theories. Some educators may have bad experience with some 'bad' games and conclude that all games are bad. On the other hand, some educators expect too much from game and end up with disappointment.

Educators need to understand the advantages and disadvantages of games before using it in teaching. Inappropriateness of applying games in education can bring adverse effect to the teaching objectives. And correct use of 'good' simulation games are effective teaching tool for development of management skills. Simulation games do not only embody all phases of Kolb's experiential cycle but provide a 'dynamic feedback' feature that hardly any other teaching medium can accomplish. In general, the objective of a business game is to develop business skills to effectively apply concepts through making decisions and taking appropriate courses of action. It ***complements*** other pedagogical approaches that focus simply on the transmission of knowledge as an objective.

The suitability of using simulation games in education can also be demonstrated from an interesting research which showed that successful simulation game players possessed some of the attributes which other research had found common to top managers (Vance, Gray 1967). The studies with university students indicated that 'All-Business Grade-point-average', 'dominance', 'intelligence', 'self-assurance', 'decision-making ability', 'perceived occupational level' , and 'initiative' were attributes which were statistically significant with correlations ranging from r=0.203 to 0.369. The results of the study with upper-middle management personnel indicated that 'vocabulary', 'supervisory ability', 'self assurance', 'decision-making ability', 'perceived occupational level' and 'initiative' were also statistically significant. These results supported those who believed that university students who turned in high game performance scores also possessed some of the traits found in those who desired or aspired to higher business positions. The results also suggested that those businessmen who desired or aspired higher business positions turned in higher game performance scores.

In today's increasingly competitive environment, quality of action is crucial, and simulation is the approach that can best develop the ability to act effectively, short of real-life experience. Simulation games, therefore, is one of the best teaching tools in management education.

3.3.1 History of Using of Simulation Games in Education

Simulation games have been widely used in the teaching of business oriented subjects in the last 30 years. The concept of using gaming for teaching and training can be traced back to the mid-19th century from a German, Kriegspiel. The early simulation games were developed in the 1950s as one of the contributions of operation research. The American Management Association produced the Top Management Simulation in 1957 and Andlinger published a description of his manually operated simulation game in 1958. (Andlinger 1958; Ricciardi 1957). In 1961, Cohen and Rhenman were already discussing in Management Science the role of management games in education and research (Cohen 1961). In 1962, the first survey of marketing games was published in the *Journal of Marketing* (McRaith 1962). During the early 1960s, some simulation games, such as the Carnegie Tech Management Game and International Operations Simulation (INTOP) (Cohen 1964; Thorelli 1962) became widely available. By 1968, almost all business schools were using some form of game in their teaching program (Graham 1969). By 1970, it was estimated that over 200 games were in existence and over 100,000 executives had been exposed to them (Shim 1978). The trend is still growing. According to a survey in U.S.A. (Faria, 1987), approximately 1914 universities were using simulation games in some part of their business programs, a minimum of

3287 separate business courses were using simulation exercises and there were between 6100 and 7200 business firms were currently using simulation exercises as part of their training programs. In 1989, Faria performed a more extensive survey on the use of simulation games in education and training and the results are shown below (Faria 1989):

Table 3.1 Use of simulation games in education & training

Survey Area	Total Sample	Response Rate	Usage Rate
Training and Department Managers	500	44.6%	54.7%
Management Consulting Firm	33	69.7%	50.0%
Business School Dean	671	61.1%	86.1%
Business School Faculty	500	56.6%	16.9%

In recent years, a lot of large companies like AT&T, IBM, Union Carbide and Monsanto have included simulation games in their training programs. Popular games like 'Top Management Experience', 'Intop and Intopia', 'Management Decision Laboratory', 'Carnegie-Mellon Game', 'Markstrat' and 'Airline Pricing Game' can always be found in universities or company training programs. Some business schools, such as those at Carnegie-Mellon, the University of Pennsylvania, and New York University, use total management games in final-year integrative courses, while other schools such as the University of

California at Los Angeles favour games centred on particular managerial functions. Two Leading European business schools, Centre European d'Education Permanente and the European Institute for Business Education, led recent testing and updating of the top marketing game, Markstrat (Shlomo, Klim 1988).

3.3.2 Use of Games in Teaching Engineering Business Management

Though no surveys on the use of games in engineering business education have been located, the usage of simulation games in engineering business education can be estimated by the level of availability.

In 1977, Couger listed out the 20 most popular games and none of them were related to production (Couger 1977). It is impossible to estimate the total number of gaming-simulation but at the time of writing, the most complete listing is in the latest edition of 'The Guide to Simulation/Games for Education and Training', edited by Robert Horn and Anne Cleaves in 1980. Over 1000 games are listed and only 21 of those are classified under 'Production, Logistics, Operations'. They are mainly small functional games on inventory control or distribution management which only covers a small portion of manufacturing engineering.

Ten years later, the situation has not improved. There is difficulty in locating a topic related to production or manufacturing engineering on the resource lists of SAGSET (Society for the Advancement of Gaming and Simulation in Education and Training) in which topics like music or health education can be easily found.

According to Elgood who had listed more than 200 management games available in Britain in 1988, there were only 13 games devoted to production but 11 of those were not computer based models. They were either discussion games or progressive games using charts or cards. Only one game (The MRP games) out of 200 covered manufacturing/production issues e.g. MRP and MRPII.

In 1993, the situation has changed significantly. Elgood in his fifth edition Handbook of management games lists 297 games currently in use in the UK. 18 of them are functional Manufacturing games and 37 are total management games. That is, around 20% of games in the handbook can be applied to manufacturing courses. The increase in use of simulation games in manufacturing may be attributed to the increase in popularity of personal computers, the increasing complexity in engineering business management. Different teaching methodology is required and perhaps, most importantly, the educators finally admit that game can be as effective as the other teaching media (Elgood, 1993).

3.4 Sources of Simulation Games

The high interest in simulation games can also be seen in the growth of organisations and journals devoted to gaming. Unlike finding a book in the library, hunting for a game suitable for a course is a difficult task. The following information is definitely helpful to designers and users in the search for or design of a game:

-Gaming Society and Centres

- Association for Business Simulation and Experiential Learning (ABSEL)
- Centre of Multi-disciplinary Education Exercises
- Extension Gaming Service
- International Simulation and Gaming Association (ISAGA)
- National Gaming Council
- North American Simulation and Gaming Association (NASAGA)
- Society for the advancement of games and simulations in education and training (SAGSET)

-Journal

- Journal of Experiential Learning and Simulation,
- Simages, NASAGA Washington, USA
- Simjeux/Simgames, Published and Edited by Pierre Corbeil, Quebec, Canada
- Simulation and Games: An International Journal of Theory, Design and Research, Sage Publications, California, USA
- Simulation/Games for Learning , SAGSET, University of Loughborough, UK
- The Gamer, AHC Publications, Luton, UK

-Directory of Games

- Stadskev, R., 1975 'Handbook of simulation in social education', Volume 1 & 2, , Tuscaloosa, Alabama: Institute of Higher Education Research and Services (Vol. 2. is a directory of social education games, centres and bibliographies)

- Horn, R.E. and Cleaves, A., 1980, 'The Guide to Simulation/Games for Education and Training', Sage, Beverley Hills, California.(Directory of more than 1000 games in all areas)

- Elgood, Chris 1993, 'Handbook of Management Games', 5th Ed, Gower (Directory of more than 230 games in all areas)

- Kirby, Andy, 1993, 'Games for Trainers', Volume 1 & 2, Gower (Directory of 75 games for a wide variety of training programmes)

3.5 Benefits of Using Simulation Games

The dynamic feedback ability is perhaps the key advantage of using simulation games in education and learning. It provides the opportunity for the participant to evaluate their decisions and live with their consequences. This is not found in conventional learning tools such as lectures, textbooks, and case studies.

Also, a simulation game is a simplified model of reality. It provides a laboratory environment to integrate and test out concepts from different subjects and theories under a risk-free environment with the benefit of compressed time . It allows practice in all areas of business management, ranging from objectives setting, strategy formation, and control decisions to tactical operations. Kossack 1961 has observed that '*Within this admitted artificial environment, gaming gives participants an opportunity to compare their decision -making assumptions with those of the game model, to discuss and evaluate both and compare them* *critically. In other words, the game serves as a part of catalyst to critical self-analysis and introspection.'*

In addition, participants are actively involved in learning rather than passively receiving knowledge from lectures. They are interested and involved in a game. The instructor is not required to stimulate the interest in learning. Game itself is excellent motivation. Studies consistently find that simulation increase student's interest, motivation and affective learning. Also, since students always work in team in a game, they learn to work with and through others. This enables participants to develop management behavioural skills such as communication skill.

In short, the benefits of using simulation games can be summarised as follows:

- to integrate material from several subjects
- to promote student participation in learning
- to provide a risk-free environment for experimentation
- to give students a broad view of the necessity of the subject at the start of the course
- to give students a broad view of the subject at the end of the course
- to help students learn how to implement theory
- to help students respond to a changing environment
- to help students develop professional competence
- to expose students to problem-solving under confusing and ill-defined circumstances

- to help students discover relationships between concepts instead of being told them
- to arouse student interest in the subject or profession
- to teach students how to compete alone or in teams
- to teach students to communicate or negotiate with partners or rivals

3.6 Misconceptions on Games

However, games are not always welcomed by people. Firstly, people always regard games as a kind of entertainment which is an activity to have fun. Parents and politicians may not like their children to play games in schools especially in universities. For example, a successful sociology course in the University of California, Berkeley had to be discontinued because gaming was the only teaching media used in the course (Jones 1985). Sometimes, even the participants have trouble understanding the objectives of a game; they play to win instead of to learn. Without any ideas of knowing what happens, they make the same set of decisions as last round only because it has yielded good result which can make them win.

Also, gaming has been used in teaching business and social science subjects in universities for more than 30 years since the introduction of the first widely known simulation game, 'Top Management Decision Simulation', which was developed by the American Management Association in 1956 (Meier 1969; Greenlaw, Herron, Rawdon 1962). The name 'business game' has existed in the academic and training fields for so long that whenever one mentions an educational game, people always relate it to a business game, giving the impression that games can only be used in teaching of business subjects.

3.7 Skills Developed by Simulation Games

There is a Chinese proverb saying that **wisdom cannot be told.** The mere act of listening to wise statement and sound advice does little for anyone. Many skills must be developed by the individuals themselves. This section will examine the most important skills exercised by simulation games.

- Knowledge

Acquiring new or specific knowledge is not a central objective of the simulation game. What simulation games contribute to is the development of the user's own framework of knowledge. This type of framework provides the foundation for students to apply concepts and theoretical ideas when faced with a similar situation in future.

"Learning 'theorists' have proven that memorisation doesn't work. Students must be learning how to think for themselves. They need to know how to do critical thinking." (Davis1966)

- Analytical Skills

Data is not of any value until it is processed to become useful information. Simulation games can equip the student with the ability to classify, organise, analyse and evaluate information. This promotes logical thinking in new situations.

"The ability to abstract, organise, and use information from a complex and diffuse environment is stressed, as is the ability to forecast and plan." (Shubik 1975)

- Skill Application

Concepts and techniques abstracted from books or lectures cannot provide solutions by themselves. We need to make appropriate judgements concerning when to use them and how to fit them into the analysis. With the assistance of simulation games, users can apply the knowledge they posses and test their applicability when facing different business environments.

- Creative Skill

Creativity is particularly important in generating alternative approaches when logical analysis has failed. Creative thinking prepares students to analyse the problem from a different angle which originally seemed to be impossible. Simulation games by their very nature encourages creative thinking and ideas, by helping people to overcome some of the well-known barriers to creativity such as the pressure to conform and the fear of looking foolish. (Richards 1985)

- Decision Making Skill

Simulation games demand the skill of reasoning and a deep understanding of the nature of problem before making any decision. This is action-oriented and requires analysis and creativity.

" [Simulation games] allow the executive to sharpen decision skills and experiment with strategies that executive may fear using without a trial..." (Faria 1989)

- Communication and Negotiation

Communication and negotiation are vital elements in business. This is because a manager not only needs to maintain good working relationships inside the organisation with his subordinate or superior, he also needs to keep a close relationship with the suppliers or vendors. Simulation games are usually played in teams. Communication among team members and negotiation between competitors are common.

- Forecasting and Planning Ahead

Decision making in simulation games may affect current and future performance. Simulation games provide the opportunities for players to estimate the pros and cons of all the decisions taken.

Simulation games demand that the participant recognises that their decisions not only have immediate effects, but also cumulative and long term consequences.

"The complexity of modern business operations and the time lags that occur before the effects of decisions are realised put a premium on the managers' ability to look ahead....... He should predict the consequences of his decisions so that he can measure what he accomplishes against what he planed to achieved." (Cohen 1960)

It is obvious that if a game is properly constructed, it can be a good teaching and training tool for engineering business management. Finally, this chapter is concluded with the research findings published by the British Audio-Visual Society. (Sherwool, Vervest 1991)

We remember about:

10% of what we read

20% of what we hear

30% of what we see

50% of what we see and hear

80% of what we say

90% of what we say & do at the same time

CHAPTER 4

GAME DESIGN PHILOSOPHY OF AN EDUCATION SIMULATION GAME IN ENGINEERING BUSINESS MANAGEMENT

4.0 Design Philosophy

Research has been conducted on the learning effect of a game's complexity (Butler, Pray, Strang 1979; Wolfe 1978), the instructor's degree of involvement (Wolfe 1975); prior student associations (Cohen, Rhenman 1961; Mckenney, Dill 1966); the game pacing (Aplin, Cosier 1979; Sampson, Sotiriou 1980; Walker 1978); learning objectives encouraged (McKenney 1963; Mckenney, Dill 1966) students preparation, aptitude ; cognitive structure (Wolfe 1978; Wolfe, Chacko 1980); group dynamics (Greenblat 1980), and team-size effects (Wolfe, Chacko 1983). Unfortunately, except complexity, the findings do not contribute much on the design of a game, instead, most researchers concentrate their studies on the role of administrators or the attributes of the users. Nevertheless, these findings are useful to provide a global picture of a successful simulation game.

Although numerous games have been developed in various fields, little attention has been paid by the game authors in designing of games (Jones 1985; Greenblat 1988). The designing procedures of a game are described as a 'black box' by Ellington, Addinall and Percival 1982. Only four books devoted to design have been published since 1970, namely, 'Simulation and Gaming in Social Science' by Michael Inbar and Clarice Stoll (1972), 'Handbook of Game Design' by Ellington, Addinall and Percival (1982). The third one is Ken Jones's 'Designing Your Own Simulations '(1985) and lastly, the 'Designing games and Simulations' by Cathy Greenblat in 1988. Unfortunately, none of them are concerned with designing games for engineering or business courses. They either concentrate on design of simple simulation games such as card games and board games or complex social sciences games. The designing procedures of a simulation-game in engineering business are in fact a 'black box' within a 'black box' .

It is quite obvious that the design procedures of a social science game is quite different from an engineering or business game. In designing a social science games, human behaviour is always a major concern but in design of a engineering business management game, a lot of engineering and business theories have to be considered and incorporated as well.

4.0.1 Systematic Approach

Basically, there are two different approaches in game design, '***Systematic***' and '***Inspirational***' (Ellington, Henry, Addinall 1982). 'Systematic approach' refers to a mechanical step by step methods of designing and tackling problems in a systematic way. In the 1979 conference of the International Simulation and Gaming Association (ISAGA), on the theme 'How to build a simulation/game', Dr. R. D. Duke presented 'Nine steps to game design':

" *(1) develop written specifications for game design;*
(2) develop a comprehensive schematic representation of the problem;
(3) select components of the problem to be gamed;
(4) plan the game with the Systems Component/Gaming Element Matrix;
(5) describe the content of each cell (above, 4) in writing;
(6) search my "repertoire of games" for ideas to represent each cell;
(7) build the game;
(8) evaluate the game (against the criteria of a, above);
(9) test the game in the field, and modify. "

Similar ideas can be seen in Greenblat's books, 'Principles and Practices of Gaming-Simulation (1981)' and 'Designing Games and Simulations (1988)'. This approach of designing games also received supports from Ellington, Addinall and Percival (A Handbook of Game Design) who wrote:

"If you feel that you are capable of designing a game by means of the first approach (inspirational), then neither we, nor anybody else, can probably be of further help to you. Either you are the type of genius who can dream up a successor to RUBIK'S CUBE in a moment of creative ingenuity, or you are not. If, on the other hand, you feel that you need to adopt a systematic approach to game design, then we believe that we can give you some useful advice on how to set about the task"

The following diagram (fig. 4.1) developed by Ellington and Addinall (1976) further elaborated the systematic approach in game design:

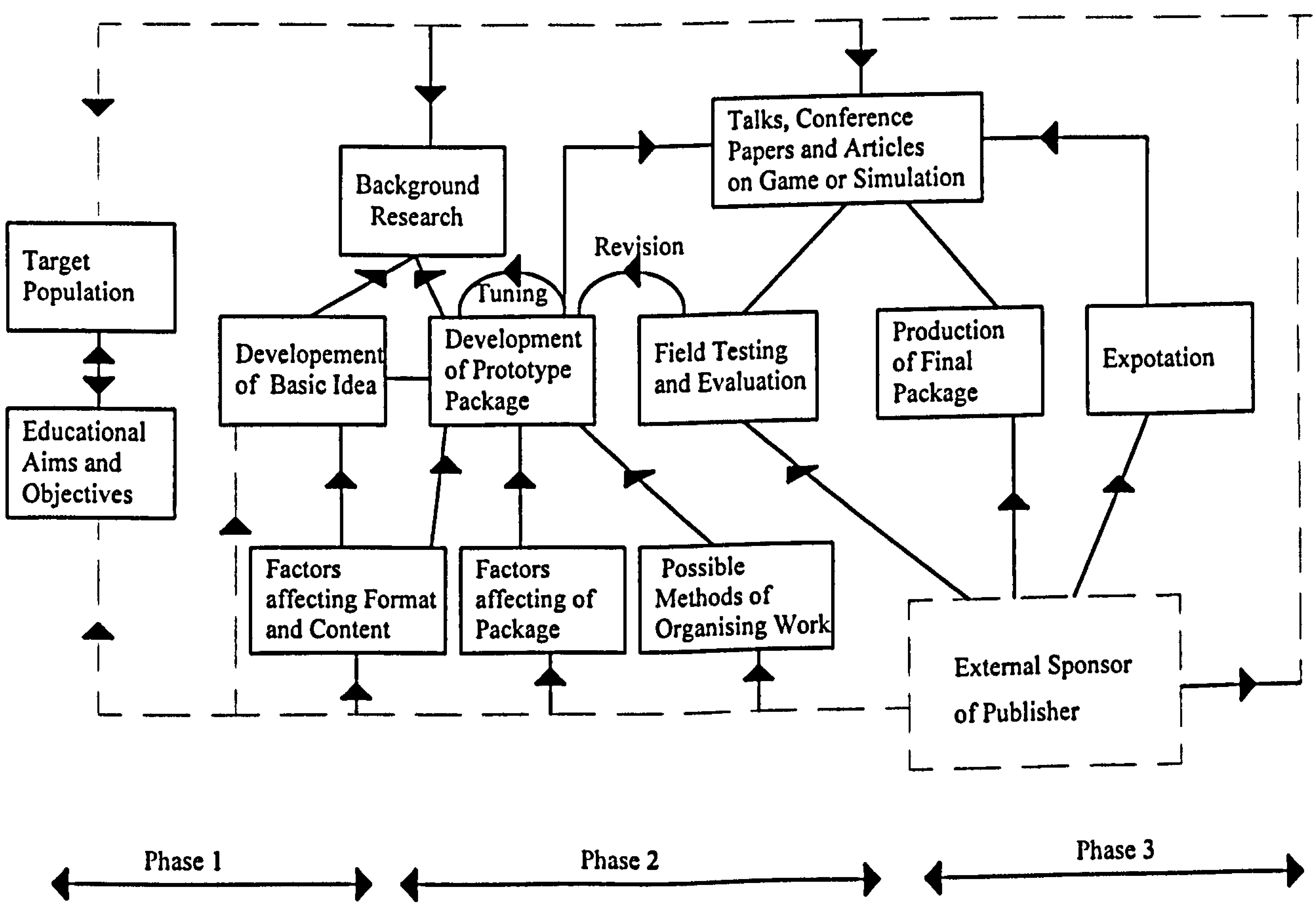


Fig. 4.1 A systems approach to the development of an educational game or education

(Extracted from Eddlington & Addinall, 1976)

4.0.2 Inspirational Approach

On the contrary, Ken Jones attacked this systematic approach vigorously in his book 'Design Your Own Simulations'. He named it 'Assembly Line' approach which operates like a motor-car assembly. He used the words 'ridiculous', 'unrealistic' and 'inappropriate' to criticise the approach which he believed would lead to bad simulations. In fact, he was not alone; Garry Shirts, in an article, 'Ten mistakes commonly made by persons designing educational simulation and games', wrote:

"The designing process, in my experience, is not sequential at all - new ideas 'F' requires an adjustment or rethinking of ideas 'A', 'B', 'C' and 'D'. And such adjustment in turn may suggest changes in idea 'F'. One moves back and forth among ideas...much as a performer who keeps a dozen or so plates of china spinning simultaneously on slender poles."

Ken Jones called this iteration (inspirational) approach 'cooking pot' which works like a chef who stirs the pot and tastes the mixture, and adds or reduces ingredients. He emphasised "*the actual starting point of the whole mental process need not be an aim, but an idea about the contents*". Jones' ideas were further supported by Morry van Ments in an article, 'Principles of Design and Use of Simulation and Games' in 1985.

Jones believed that designing itself is a personal task. Sometimes, one can design a game by instinct. On the contrary, one can waste a whole week without any achievement even though he tries to attack the problems step by step. Different types of simulation games have different focuses. Human minds work so strangely that ten people may tackle the same problem in ten different ways. A good simulation is the one which can achieve its objectives effectively and the designing approach should not be taken as a criteria of evaluation. As long as a simulation game serve its purposes, it is a good one.

The standard procedures of designing simulations and games have become the most controversial topic in the last decade and the arguments still continue. The existence of a general principle of designing all games is questionable . However, a set of general guidelines for design of a manufacturing management game is definitely useful. Both 'cooking pot' and 'assembly line' approaches are too extreme and mixing up these two approaches will be suitable and helpful in the design of a manufacturing management game.

It is true that a good simulation is the one which can achieve its objectives effectively and the design approach should not be taken as a criteria of evaluation. However, research has shown that there is no scientific way in evaluation of a game which can provide a conclusive result. Most importantly, the success of a game depends on too many factors (as mentioned in Chapter 2) which the game designer may not have full control on. At the end of the day, whether a game is effective or not is predominantly subjective.

Since there exists no conclusive way of evaluation, the justification should be moved back to the design stage. If the formats, features and each element of a game are carefully selected and justified in the design process, the chance of obtaining a 'good' simulation game is definitely higher. Similar instance can be found in the employment selection criteria in AT&T Bell laboratories which only employs staff graduated from the top 20 universities. The personnel manager in AT&T explained that it is impossible to judge a person in a two day interview. Thus, they only considered the applicants from the top 20 universities to minimise the chance of recruiting a 'bad' staff. They believed that a 'good' university has a higher chance of producing a 'good' candidate.

So, although Jones' philosophy is perfectly acceptable that a good simulation is the one which can achieve the objectives effectively, in designing a complex engineering business game, 'cooking pot' approach may not be appropriate because the probability that a 'good' engineering business simulation can be created by instinct is questionable. On the other hand, an 'assembly line' approach does provide a single word in the designing process. It is particularly useful for the game designers to have a check-list or a set of guidelines which reminds them some of the basic factors they may have forgotten. Going through the checklist each time can help the game designers to organise the thought and stimulate some new ideas as well. Nevertheless, human being is not a robot. One cannot program a human being to design or think by following a sequence of instructions. Moving back and forth among phases to make changes are necessary. It is also true that idea does not originate in a systematic manner. Today, one may come out an idea on how to run the game more effectively. Tomorrow, one may think about the appropriate level of complexity in a game. The designing sequence is uncontrollable. However, at the end of the day, all these thinking have to be organised to fit into the right place to form a game and the systematic approach does provide the help.

4.1 Design of a Engineering Business Management Game

The systematic approach suggested by Greenblat does provide a general guideline for game designers of all games. However, the guideline does not provide any concrete help to the game authors especially in EBM area. It is probably because different types of games (social science, manufacturing, team working, etc.) have different objectives (education, entertainment, etc.) and thus have different emphasis in the design approach. As a result, a game author has to design a game from his own experience and no concrete help can be obtained from anywhere.

In design of an EBM game, 'education' is obviously the key objective. So, in designing the feature, formats and the elements in the game, the education objectives should be always borne in mind.

4.1.1 What Contributes a 'Good' Engineering Business Management Game?

In order to have a 'good' engineering business management game, five areas have to be addressed:

-Formats and Features

A simulation game may take on different formats and features which will definitely affect the effectiveness of a game. Typical features and formats include 'complexity', 'realism', 'interaction', 'focus', 'manual or computerised', and 'entertainment', etc. The appropriate levels of these factors have to be determined according to the game objectives. Although the features and formats themselves do not direct contribute to the education objectives, without the appropriate levels of these factors, the education objectives of an EBM game can never be well achieved. The appropriate levels of formats and features are critically examined in detail in the next chapter (Chapter 5).

-Rectification of Weaknesses

The advantages and benefits of using games in education can be found in every game literature with detail description. However, they all avoid to discuss the weaknesses of games. In particular, some common weaknesses are found mostly in EBM games. Although some weaknesses of games originate from the nature of games, some are actually design flaws and can be rectified from game design. A game can never be good if it still keeps the same drawbacks as before. A list of common weaknesses found in games especially in EBM games have been identified and critically examined in Chapter 6. Although the list may not be complete, however, it can stimulate the game authors to consider a game element from different perspective before the element is incorporated in the game.

-Teaching Material

There are numerous areas in EBM. It is understandably that a game cannot cover all the important issues in it. The elements or functional areas that should be integrated together in a game depends on the messages or ideas that the game designer would like to convey through these areas? Too much material covered in a game sometimes bring averse effect to the educational objectives.

Although it is impossible to provide any concrete suggestion in selection of teaching materials as in the above two areas, however, the following considerations should be borne in mind when selecting an element:

(1) The game is for education purpose. Only engineering business theories should be incorporated in the game.

(2) If the examined issue is popular in other existing games? If yes, the issue may not be as important as some issue that has been overlooked.

(3) Whether the degree of complexity involved and the message getting across are well balanced? Participants have limited time to learn, the author has to make sure the time spent is worthwhile.

(4) Whether the issue can be incorporated in the game in a sensible way? Since only a certain number of selected issues can be incorporated into the game, so

if the examined issue does not fit into the picture well (even though it is important one), the issue has to be reconsidered.

After all, the game author should ask himself a question for every element that he puts in: 'How this element can help to achieve the education objective of the game?'

-Running the Game

How can a game be run most effectively? The role of the administrator, pre-game preparation and the use of decision support aids, etc. all affects the effectiveness of a game. Although some of these issues are in fact out of the control of the game designers, however, if the game administrators knows the game well and understand the logic of the game, the chance of having a successful run will be increased. In fact, some of the common weaknesses may be rectified by the administrators. Although the game administrator cannot change the game logic, however, the game scenario, the team size, number of teams, number of rounds and numerous game factors are all controlled by a game administrator who in fact controls the success of a game. Wolfe 1975 had confirmed the importance of the existence of the game administrator in a game. The instructor guidance in a game greatly improve the learning process.

In fact, a game without a game administrator showed no measurable knowledge obtained by the participants.

Numerous research have been conducted on the role of a game administrator and several conclusive findings have been published on how a game can be run more effectively. For example, Thorelli' a professor at the University of Indiana at Bloomington, had an interesting observation about which student teams succeed best in the game. *'Teams with students widely differing scholastic ability do better, than homogenous teams made up solely of good students. If all team members are top students, everyone wants to be top banana'.*

So, a game administrator cannot just leave the game to the students and wishing that they can discover all the messages and ideas from the game. Besides understanding the logic and the education objectives of a game, the game administrator should understand his role in a game and prepare for it just like preparing for a lecture. After all, a game is not a self-teaching aid. Since the role of a game administrator is out of the scope of game design, it will not be discussed in detail in this paper.

- **Administrators' and Users' Opinion**

Design is a personal task and one can never see his own bias in a design. At the end of the day, a game is used by the administrators and the users. If they do not like the game for whatever reasons, even a 'good' game is useless. So, the design has to be customers oriented. During 1992 to 1994, three surveys have been performed to obtain the views of the administrators and users on manufacturing games to confirm the findings in the last two Chapters and the results are discussed in Chapter 7. Although some of the suggestions are subjective due to their different backgrounds and experience, their views are always important. The game design approach in EBM can be described by the following diagram:

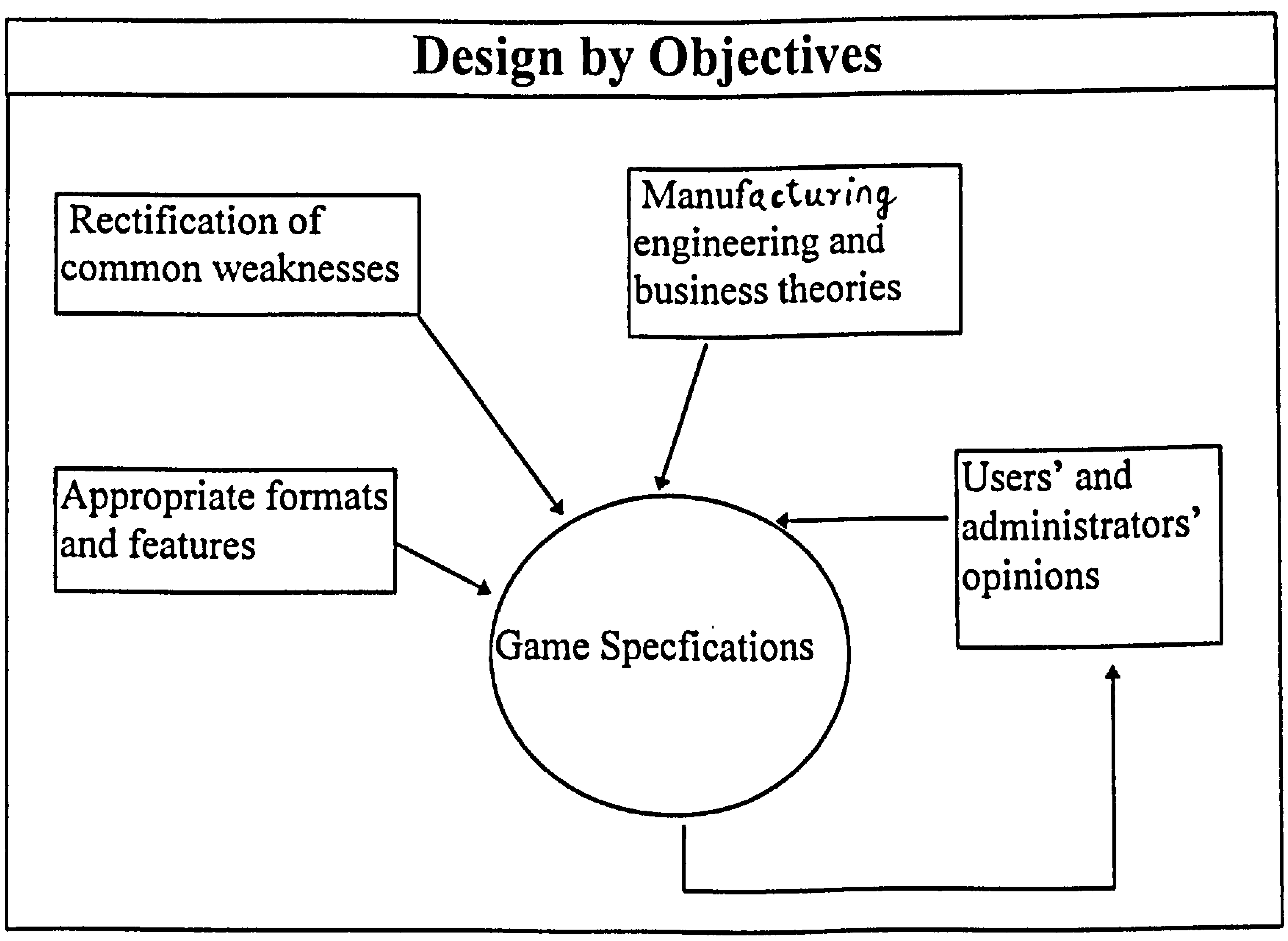


Fig. 4.2 Game design approach of an education simulation game in EBM

Game design is an iterative process. Since the users and the administrators are the end users of the games, their views are in fact the evaluation of the game especially there is no conclusive way on game evaluation.

It has to be emphasised that the game design approach described above does not intend to replace the systematic approach but to supplement the areas that have been looked and provide more concrete suggestions to the game designers. The systematic approach checklist is still necessary in design of games.

4.2 Systematic Approach Checklist

One of the objectives of this paper is to establish a standard interface for manufacturing games so that functional games built on this platform can be integrated together to form a integrated manufacturing game. The idea is originated from the need of a total integrated manufacturing game in manufacturing education. However, developing one 'good' total integrated manufacturing game cannot solve the problems. Hundred of good integrated manufacturing games are required in manufacturing education. So instead of building a particular total integrated manufacturing game, a standard interface is constructed to form a platform for all engineering business games.

A systematic checklist is gone through below to demonstrate the organisation of the above thinking process.

(1) Why is the game necessary ?

(2) What is/are the objective(s) ?

(3) Does a suitable simulation-game already exist ?

(4) Is there a simulation-game that could be modified ?

(5) Who are the likely participants ?

(6) Who are the likely operators ?

(7) What is the probable context of use ?

(8) What are the available resources for development?

(9) How is the game structured to meet the requirements ?

(1) Why is the game necessary ?

Since the trends in manufacturing industry have been evolved from production oriented to customer oriented, an integrated approach in education of manufacturing management is needed to cope with the changes in the industry.

(2) What is the objective ?

The objective of the integrated games is to show an integrated picture of manufacturing management to the participants. The integrated management approach is composed of 'functional optimisation', 'internal integration' and 'external integration'.

(3) Does a suitable simulation-game already exist ? and

(4) Is there a simulation-game that could be modified ?

Location of a suitable existing game to achieve the objectives effectively is a failure because a lot of weaknesses in the existing total management games are discovered. Also, the existing total management games are only good at 'internal integration'. They fail to give a total integrated picture to the participants.

(5) Who are the likely participants ?

There are two types of targeted participants:

- Final year of undergraduate students and master students studying in business or engineering departments

- Middle or senior company executives

They are both assumed to possess basic knowledge on business and engineering areas. In order to understand the behavioural difference between these groups of participants which may affect the game design, their behaviour in playing games have been studied:

(1) In graduate school, Intelligent, elegant things are done in games that executives don't do -- like trying to figure out precisely how much of a product to produce at the lowest possible transfer cost. For executives, the name of the game is get the product out the door and to get people to accept it.

(2) Executives are very aware that in some areas, they have more information than they care to while in others, they don't have nearly enough. They know that is the way life is, and take it with equanimity. Students, on the other hand, want closure. They want complete rationality, they want to know precisely what the situation looks like, in a world full of free and complete information. Neither the real world nor the simulated one looks that way.

(3) Executives usually perform better in a general management game partly because they seem to understand the fine line between co-operating among themselves and competing with other teams. They are more customed to

working together than students. On the other hand, college has been individualised, always has been, so students spend more time to co-operate within teams.

(4) Executives may be more co-operative, but they are more competitive. Their strength is that they know when to be which. The same things that makes them good mentors make them good players. What executives want to do very much is to destroy other team. They go to great lengths to set up a situation where they can push the other team's product off the market. Undergraduates or graduates are hardly found as fiercely competitive.

A good simulation game is in fact an art form. One can play out what's in his mind. If executives get into it and play it as a form of combat, that is good. If graduate students play it as some form of intellectual exercise, that is also acceptable. The above findings do not affect the design of the game directly but the game administrators may find the results useful to adjust their roles when the games are run.

(6) Who are the likely operators ?

Since the targeted participants are master students or company executives, the operators or the administrators of the game are expected to be the lecturers in universities or the training managers in companies. In order to run the games effectively, the administrators have to know the education objectives and understand the games thoroughly.

(7) What is the probable context of use ?

The modular integrated manufacturing game is designed to run parallel with lectures in engineering or business classes as tutorial or laboratory sessions in undergraduate or postgraduate courses. It may also be used in a company training program. Generally, courses on 'Production', 'Marketing', 'Finance' and 'Communication' are expected to make use of the game. Ideally, an 'integrated' course can be included in the curriculum to run the integrated game at the final year of a degree course or at the year end of a one year master courses.

(8) What are the available resources for development?

Since the integrated manufacturing game is an educational package used in universities, normal university resources are available. For example, computer facilities are sufficiently available. However, 'portability' and 'hardware independence' are always preferred.

(9) How is the game structured to meet the requirements ?

Instead of developing a particular total integrated game which may be only good at a designated subject area. A standard interface is established for all EBM games so that functional games built on this platform can be integrated together to form a total integrated manufacturing game. Integrating different function games together will form different total integrated manufacturing games which can address different manufacturing areas. Hierarchical structure is applied to integrate different games together. Two functional games, production and marketing are built as prototype to demonstrate the generic structure.

4.3 Conclusion

Design is a personal task. No one can program another person's thinking process of how to design an EBM game. However, without a systematic guideline to organise the ideas and thinking process, the chance of designing a 'good' simulation game is slim. Since EBM games are not for fun but for education purpose, the game authors should minimise the risk of designing a bad game. The three missing areas in design of EBM games are discussed in detail in the next three chapters.

CHAPTER 5

DETERMINATION OF THE APPROPRIATE LEVELS OF FEATURES AND FORMATS

5.0 Introduction

Games can take on different formats and levels of features. Formats refer to the physical environment when running a game such as 'location', 'number of teams', 'team size' and 'with/without administrators', etc. Features, on the other hand, form a general structure or framework of a game such as 'degree of realism', 'level of complexities' and 'degree of interaction'. These no doubt will affect the effectiveness of business games. The statistics on the formats and features of existing EBM games in UK are presented for reference. It is believed that if majority games take on a particular format, there must be reasons behind it. The statistics are calculated based on 'Handbook of Management Games' (Elgood 1993). Fifty four EBM games have been chosen for further study from two hundred and ninety seven games which are currently used in the UK.

Table 5.1: Types of Business Games

Production Games	13%
Marketing Games	28%
Total Management Games	42%
Others	17%

Table 5.2: Time Required

1 to 2 hrs	6%
2 to 4 hrs	23%
4 to 8 hrs	18%
1 to 5 days	48%
>5 days	3%
Flexible	2%

Table 5.3: Process Format

Centralised Computer Interactive	44%
Centralised Computer Non-interactive	6%
Direct Access Computer Non-interactive	20%
Direct Access Computer Interactive	4%
Manual	20%
Other Computer Games	6%

Table 5.4: Number of Teams

1 to 6	45%
8 to 10	18%
>10	9%
Flexible	28%

Table 5.5: Recommended Team Size

1 to 2	8%
3 to 5	59%
6 to 10	15%
>10	2%
Flexible	16%

5.1 Appropriate Formats of an Engineering Business Management Game

5.1.1 Location

An EBM game is designed to run in Universities to complement traditional teaching methods in engineering business education. Classroom environment is assumed and enough syndicate rooms are expected for group discussions.

5.1.2 Number of Teams and Team Size

Number of players assigned to a team and number of teams in a game are more often a function of administrative concerns rather than an objectively derived evidence regarding optimal individual or group learning. Those administrative concerns include class size, the number of companies modelled, processing time and the amount of coaching time the instructor wants to spend per firm.

The learning effects of different group sizes are hardly located in Education Psychology. More evidence, however, came from decision science literature. Size-related research includes the peculiarities of odd and even numbered groups (Berelson 1964), optimal interacting group sizes (Dale 1952; Filley 1970; Holloman 1971) and the internal dynamics of size increases (Thomas 1963). Research also shows that group decisions are superior to individual derived ones. (Allenm, Marquis 1964; Kelley, Thibaut 1969; Shaw 1932). Overall, an odd-numbered group of *five* participants seems to strike a balance for group creativity, decisiveness, and accountability with minimum internal administrative problems.

The effects of various decision-making team sizes in a complex business games were also studied. Wolfe and Chacko 1983 concluded that three and four member teams made more decisions and they also obtained higher profit and lower bankruptcy rates. Two member firms experienced only marginally significant knowledge increases while single member firms experienced the most bankruptcies and dropouts. These findings were consistent with the group-size/dynamics literature that stated that multi-member groups are more effective than individuals when the task requires creativity plus implementation.

It is understandable that team games are usually better than individual games because they provide an opportunity for team-mates to help and learn from one another, and avoid one problem of individual games, which is that more able students consistently win. If all students are put on mixed-ability teams, all have a good chance of success. (Slavin, 1991)

These findings agree with the statistic presented that 59% (Table 5.5) of current business games recommend 3-5 members/team. Nevertheless, the 'ideal' team size is a function of the group's task and time constraints as well as the participants' individual abilities and motivations.

Theoretically, the number of teams can be quite flexible if the game administrators can spend enough time to guide the discussion for each group. Also, there exists no research on the optimum number of groups that a game administrator can handle. However, a game administrator may not be able to coach more than four teams effectively. In fact, coaching for three teams are enough work for one game administrator. There are always unpredict problems faced by in some teams and the losing teams always need more than average time from the game administrators. Although the appropriate number of teams are also affected by the class-size, the levels of complexities, degree of interaction between teams and several physical resource constraints (number of syndicate rooms, number of computers, etc.), in the real situation, the dominant factor is the number of administrators available in a game. Taking into account that most games are administrated by one to two instructors in universities, three to six teams are recommended for a typical EBM game. In fact, the finding also matches quite well with the statistics (table 5.4) that only a small amount (27%) of existing manufacturing games recommend 8 or more teams in the game.

5.1.3 Time Allowed and Number of Rounds

Time allocated, number of rounds and complexities in business games are clearly inter-related. However, research results on these areas are contradictory and it is still the most controversial issues in business games. (Wolfe 1978; Butler, Pray & Stang 1979). While Wolfe concluded that most students were bored and learning stopped after the 6th rounds in a medium complex game, Butler and his colleagues argued that less than 25% of students were bored and learning was still on going even after the 8th round in a 'similar' environment.

Since the outcomes of a game depends on too many factors and different games were used by different researchers under different environments, the chance of obtaining the same results from different instances are slim. 'Similar' environments are almost impossible to set up for the same experiment. That is why social science findings are mostly subjective. However, three common findings are spotted:

(1) Even a simple game is educational for at least three rounds
(2) A complex game can last longer than a simple game (before the students get bored and learning stops)
(3) Games longer than 8 periods are not recommended

Clearly, the time allowed for each round must be related to the level of complexities in a game. Unfortunately, no research can be located on the relationship between the complexity and time allocated in a game. It is because many other factors, like the abilities of the students, team size as well as the degree of involvement of the administrators will also affect the time required per decision round. Where there are too many dependent variables around in the area, scientific research can be hardly performed. However, three interesting behaviours are observed:

(1) Students need longer time in the first two rounds and then time required in each round gradually decreases. This can be explained that the students take time to study the company backgrounds, etc. in the first round and they may need even longer time to analyse their first feedback.

(2) Some teams are always late. Deadlines are hardly met by all teams.

(3) When the times are tight, something usually goes wrong. Either the computer models do not run , the printers do not work or some teams have entry errors on their decision sheets. This may be explained by the fact that people make more mistakes (both the users and administrators) under pressure. From the statistics, the lengths of business games are quite evenly distributed between 2 hours and 5 days. Specifically, games that run less than 2 hours or more than 5 days are rare, and games last between 1 to 5 days are most popular. (48%).

In fact, there is a warm up period which may last up to several hours before the students can start learning from the game. So, short games may not be efficient. That is, students may have to spend more time to get familiar with the games than learning from the games. However, the students get boring if game is too long.

5.1.4 With/Without Administrators

It has been discovered that the instructor's role in guiding participants' learning in business games is almost as important as in a lecture class format. Numerous research can be cited on this (Wolfe 1975; Wolfe 1978, Mckenney 1967). Although some people still believe that game administrators are less important in non-interactive games, however, the role of administrators are much more than collecting the inputs and handing out the reports. Whether a game is interactive or not, it still requires a game administrator. Game is not a self teaching aid. In fact, playing a game without an administrator is like handling a gun without any guidance. Games do not explicitly tell the participants what are right or wrong as in lectures. Administrators' role can be divided into three stages of a game:

In pre-game briefings, administrators need to explain the objectives of the games and the situations that the participants are going to face. The good lead-in for the game enables the participants to concentrate more on the issues raised by the game.

During the game, the administrators become consultants who give advice to companies and solve any problems. Based on the progression of the participants, the administrators may need to trigger some of the random crises, increase the level of complexities and swap participants between teams to increase realism according to their teaching objectives. Finally, to avoid literal transfer, participants should be debriefed at the end of the game play to limit any unsuitable application of game results. In fact, some weaknesses of games, which will be discussed in the next chapter, are the results of absenteeism of administrators.

Therefore, the role of administrators in games can never be overlooked. As Pray 1988 mentioned "*A crucial role in the success of a simulation is played by the training and development officers*".

5.1.5 Manual or Computerised

Statistics show that only 20% of manufacturing related games currently used in UK are manual. This is understandable because computer games do have a lot of advantages over manual games:

- provide fast and highly accurate processing
- can store, organise and retrieve large amount of game-relevant data in a short time
- can motivate and entertain participants by use of graphics and cartoons.
- can reduce the administrative load of the administrators (e.g. on-line help and controllable environment)

However, a closer look at the statistics will discover that most computer games are under centralised control which means the users in fact do not interact with the computers directly. Besides the problems of having enough computer resource, there exists two major disadvantages for the participants to input their decisions to the computers directly:

(1) User knowledge

Time is limited and different computer software have different function keys and screen formats. In direct access games much time is wasted while participants learn how to use the machine.

(2) Computers are divisive

Computers split the world into those who can handle them and those who cannot. Team members who are computer literate often dominate the discussions and hence de-motivate the participants who are not familiar with computers.

A further study will also detect that an extremely small portion of direct access games are interactive games. Two reasons behind this:

1) Although Local Area Network (LAN) can be easily found in universities, the idea of using this technology to implement business game is fairly new.
2) Although LAN allows teams to communicate through the network, the portability of a game is highly reduced.

Taking advantages of manual user interface and the speed of computers, a EBM game should be centralised control computer game. So, while students are enjoying the team working environment which encourages high participation, swift processing can also be provided. Another major advantage of manual user interface is the flexibility provided to the administrators. Computer interface restricts the formats of inputs and the number of decisions entries per round. On the other hand, manual user interface allows the game administrators to have total control on the subject areas and the number of decisions that the students need to address. This idea can in fact apply to all existing games with centralise control. For example, a marketing game may require 12 decisions/round. However, by re-designing the decision forms, the administrators can cut down the number of decisions to 10 and pre-define the 2 decisions which may obstruct the learning objectives. That means, the administrators can simplify any existing business games according to his need. Also, printing out a new data form is much easier than messing around with the computer codes. Furthermore, manual interface provides opportunity for the administrators to glance through students' decisions which may provide hints to the administrators that whether the students need further guidance or not before any serious mistakes are made. So, computers not only divide team members, they also separate participants from the game administrators. Strictly speaking, it does not matter how user friendly or colourful is the computer interface, manual interface can achieve the education objectives more effectively.

5.2 Appropriate Features of a Engineering Business Management Game

The most important feature in a business game is possibly the ability to provide 'dynamic feedback'. In fact, this may be the most distinguishable feature between a game and other teaching media, like case studies. The processed results are given back to the participants based upon their decisions after each round so that they can learn from their mistakes or success and then make decisions for the next period. This dynamic feedback feature can hardly be found in any other teaching methods.

Besides dynamic feedback, there are five important and controversial features in business games: realism, complexity, interaction, focus, and entertainment.

5.2.1 Realism

> *" ...... if the realism of business games could be increased, a more effective educational and research tool than previously existed would be created." (Cohen 1960)*

This presents the most important constraint in developing a game. Realism means all the major success factors, both controllable and uncontrollable, must be integrated into a single entity so that the game provides a better real world representation.

Many game authors agree on the effectiveness of an education game will depend largely on the quality of the simulation in representing the behaviour of the real world and a realistic simulation is by definition good. However, this conceals a number of fallacies:

Firstly, although realism and complexity are clearly related, it does not mean that all business have the same level of complexity. Secondly, realism is not necessarily good. This idea relates to the question of objectives. For educational purpose, Kibbee 1961 already pointed out that it is sometimes necessary to make certain parts of the model intentionally unrealistic in order for the whole to seem realistic to the players. A true representation may be far less convincing than an exaggerated or simplified representation. Thirdly, what is realistic today is unrealistic tomorrow. Striving for complete realism is, itself an unrealistic goal. Fourthly, The substantial realism is not essential from the player's point of view. Truemper and Dean 1974 quoted a player's reaction to their game, OPRAD: "*Compared with game X ...., I felt that there was less realism this time, but I learned more in the OPRAD game*".

However, although the attainment of complete realism in a game is impossible due to the multitude of real world ambiguities, verisimilitude is absolutely essential. Verisimilitude is the appearance of reality as perceived by game participants (Gooding, Zimmerer 1980). In fact, if game designers can define functional relationships which are at least basically realistic, then verisimilitude will not be difficult to obtain.

5.2.1.1 How to Determine the Right Level of Realism?

The question arises as to how much fidelity is necessary without degrading the value of the training received by a student. The game designers must design what elements in the operational situation must be represented and with how much fidelity. Biel (1965) listed out two aspects of simulation must be considered to determine the degree of fidelity required:

1) What equations and functions must be simulated?
2) How accurately must be the stimulus situations on which training is being given represent real life?

Full fidelity may not also be cost effective. The following diagram illustrates the relationship between cost, fidelity and transfer of training:

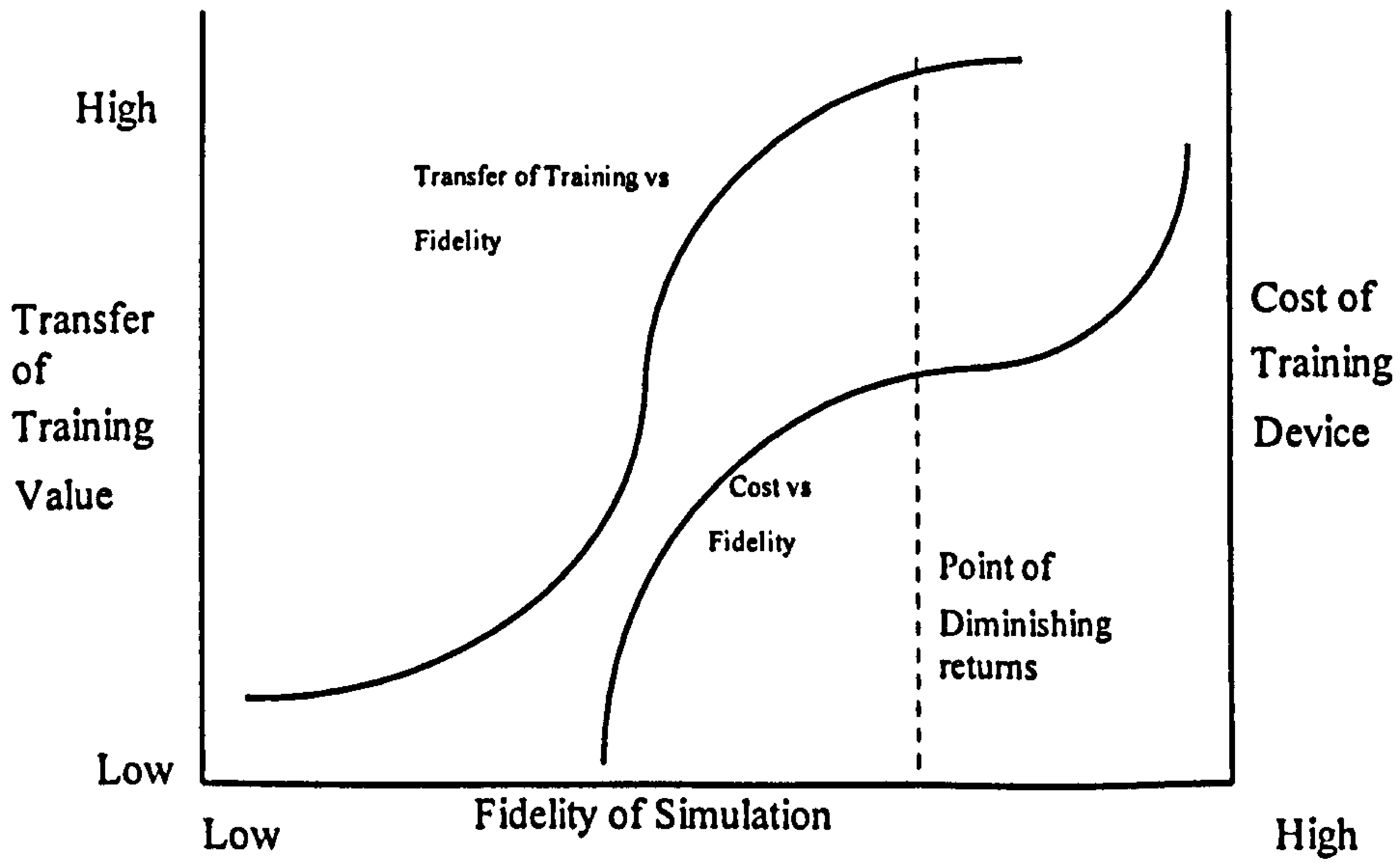


Fig. 5.1 The relationship between Cost, Fidelity and Transfer of Training

(Adapted from Miller, 1954)

Thus, the aim of the simulation game design should be to get the most effective and economic conditions for learning rather than to aim at high fidelity for the sake of it. However, high fidelity may become a necessity in a research game.

5.2.1.2 How Real Should an Engineering Business Management Game Be?

An EBM game is used as a simulated environment laboratory to test different textbook ideas. So, the design should be based on the integration of diversity of known 'micro-theories' on market and production. It has to be possible to trace each feedback or outcome back to the decisions. A good business game should offer a realistic representation of the business world and reality may be exaggerated or simplified to high-light the purpose but one may not distort or redefine it. Random factors should be cut to a minimum unless it is necessary. Generally there are two types of random factors:

(1) Random result - some game designers in order to emphasis on 'equal pay-off' is not necessarily true in the real world, random factors are introduced in the sales. This type of random factors in fact obstruct the learning process. A team may make all the appropriate decisions and still receive a bad result due to bad luck. This may be true in the real world but it hiddens the education objectives.

(2) Random events- random crises like fire, government price control and minimum wages can increase the realism of a business game. These random events should be fully controlled by the administrators who decide the right time to trigger the events. In fact, most of these random events can apply to most existing centralised control business games without any modification in

the programs. For example, the instructors can enforce a 'minimum wages' policy to all team during the middle of the games. The administrators can make sure they have complied with the policy before typing in their decisions in the games. That means, this type of random events can be introduced to almost all existing games without changing the computer codes.

Certainly, random events like fire cannot happen to all teams at the same time. However, the games must provide some options to avoid the events or to compensate for the lost. For example, if fire is a possible random event, the teams should have given the options to insure the companies or to install the automatic fire extinguishing facilities. By doing this, a team will not lose the game only because of bad luck and most importantly, the participants can learn from the events. In short, a random event should be introduced to the game not only to increase realism but to carry a specific message to the participants.

There are several other ways to increase realism. For example, when one team expands its plant, workers cannot just be installed like machinery, but must be trained and paid for six months before they become profitable labour. Although realism is important in a game, a well balance must be maintained between realism and education objectives in design of an EBM game.

5.2.2 Complexity

Realism and complexity are clearly connected. The complexity of a game is generally measured with three different indicators: 1)Number of decisions to be made in each time period 2)Length of instructions/manuals/data provided

3)Time requirement. (Elgood 1993) The suitable complexity of a business game is always a controversial issue.

Raia 1966 concluded that added complexity was not profitable beyond a certain point in a game. Kibbee 1961 suggested that moderate simplicity in a game is important for many reasons. Firstly, a game must be kept simple (not too many rules) if it is to be played after a reasonably short briefing. Also, the mechanics of submitting a decision and of receiving reports must be as simple as possible. Additionally, complex games can be costly in both administrator's time and computer modelling.

However, Joseph Wolfe in 1978 concluded in his research that the most complex simulation produced the greatest effect in teaching facts and concepts of business policy knowledge (Wolfe, 1978). He studied the amount of learning acquired by groups of people playing three games of different complexities. Game complexity was measured in three ways ; decisions per play (ranging form four in the simple game, through eight in the intermediate game to 53 in the complex game), size of game manual (622, 3601, 15086 words respectively) and program size (100, 245, and 1118 executable statements). Wolfe showed that while the complex game produced most factual and conceptual learning, the simplest game produced the second highest amount. Also the complex game produced the highest numbers of drop-outs and was obviously not therefore as highly motivating to those who took part. His findings were immediately challenged by many game authors (Butler, Strang 1979). The argument is still continuing. Since compatible environments can hardly be found to carry out the same test, the arguments probably will never end. Nevertheless, most game authors prefer simple games which they think will teach best although if a game design is too simple there will be little motivation for the participants.

5.2.2.1 How Complex Should an EBM Game Be ?

Thus the relationship between the game complexity and the effect of the game on those who play it is far from clear-cut. Also, one must be cautious that game complexity increases not only with the number of decision variables, but also with the interrelationships among variables. The suitable complexity of a simulation game depends on various factors such as the backgrounds of the users, the time allowed for each decision round, the experience of the users on the simulation game and the amount of help form the administrators. There is no single standard for complexity.

Obviously, a balance of consideration is needed. Since there is no so called 'standard level of complexity', various levels of abstraction (level of complexities) should be introduced in a game so that the administrator can select the desired level according to his needs. A level is a combination of selected elements or functions. For example, level 1 is production. Level two is marketing. Level 3 is production and marketing and level 4 is supply chain and so on. The complexity within each level may be further simplified because the administrators can take out (pre-define) some decisions in the game under a manual user interface environment. For example, in a marketing game, the administrator can pre-define the price of the product so that all teams have to sell the product at the same price. They can only vary other decision variables, like advertising, quality and after-sales support to increase the sales. In this case, the administrators (lecturers or training managers) can start the game in a simplified version and add on material gradually to match the pace of the class. Furthermore, number of firms, team size and time allocated can also be adjusted according to the selection of complexity level.

5.2.3 Interaction

Not all games are interactive, for it is still a competition if teams compare their performance against a common standard. In a non-interactive environment, participants play against a set of rules (scenario) pre-set by the game administrators. On the other hand, in an interactive game, groups' decisions are interrelated. The feedback of a team's decisions not only depend on the scenario but on the other groups' decisions as well. There is no pre-set optimal solution or winning strategy. That is, if one plays the game twice, the set of decisions which may have led him to win the first game may end up in a complete failure in the second trial since he faces different competitors.

The great advantages of the non-interactive simulation game are convenience and clarity. There is no need for different teams to play in a synchronised manner which makes a significant administrative difference. Also, there is no minimum on the class size to start the game. And the absence of uncontrollable competitors makes it much easier to understand the logical connections between the decisions a team has made and the consequences they subsequently experience. In a business context this can be viewed as unrealistic and therefore a weakness, but only in relation to certain objectives. When examining systems that do not feature direct competition, non-interactive simulations are specially valuable. For example, a simulation game of the shop-floor production operation is by nature non-interactive.

5.2.3.1 Should an EBM Game be Interactive or Non-Interactive?

Various levels of abstractions should be included in the game and the degree of interaction should depend on the nature of the functional areas being covered. If the examined system such as marketing requires interaction to bring out the reality, then, the game should be interactive in this level. On the contrary, if interaction may in fact obstruct the learning process in the examined system such as shop floor scheduling, then the game should be non-interactive.

5.2.4 Focus

Firstly, the breadth of business management is very wide. It is important to narrow down the field to a manageable size and concentrate on areas that the lectures wish to address. Better focus on the objectives of the game allows quicker system development and enhances the learning experience. It is impossible to teach everything a manger should know about. A clear, attainable education objective is required. In this project, the primary objective is to construct and design an effective game environment to teach the integrated business and engineering knowledge in manufacturing management. Any other achievements are in fact a bonus to the development.

Certainly, it will be nice to have a game that serves multi-purpose but it is already difficult enough to construct a game for a particular usage. Some game designers try to design a game which can fulfil both the education purpose and the research purpose. Unfortunately, these two purposes have different emphasis although not necessarily contradicting to each other. Persons trying to develop a game for both research and education will find themselves constantly having to choose between the rigorous demands of accuracy and detail required by the research and the constraints imposed by the educational and managerial needs of the teacher and students. Few people are able to satisfy the demands of both. Also, a game trying to fulfil both purposes may at the end achieve neither one of those (Shirts 1985). For example, two types of focus are meant in this research:

> *Subject Area*: Integrated Approach to Engineering Business Management
> *Use*: Educational

5.2.5 Entertainment (or Motivation)

Business games work well as a learning tool because they try to get rid of classroom boredom.

> *"...... enjoyment is one of the strongest arguments in favour of business simulation as a training technique." (Carson 1969)*

Business games are by nature entertaining and thus can motivate the participants. Enjoyable factors in a game include 'Freedom', 'Optimum challenge', Activity and influence', 'A decision point', 'Feedback', 'Competition', 'Opportunity to succeed' and 'Uncertainty and Excitement'. (Elgood 1993). Hence, introduction of random factors in business games to increase excitement is not necessary.

Since these enjoyable features seem to be a result of natural selection rather than conscious planning, there is no need to plan entertainment features in any EBM game. Although some 'interesting' elements may bring more fun to the games, however, they may also bring in unnecessary complexity to obstruct the learning process. Moreover, the participants should have enough 'fun' from the game itself.

5.3 Conclusion

Although the above discussion on formats and features of an EBM game may not be perfectly true as design should be changed with environment and objective, the discussion can definitely be served as a guideline for the game authors in this area. Most importantly, in design of an EBM game, the game author should ask themselves: 'Does the format or feature that has been adapted in the game help out in the achievement of the education objectives?'. In short, the findings can be summarised as follows:

Recommended levels of formats of an EBM game:

- Location: Classroom environment
- Number of teams: 3 - 6
- Team size: 3-5 member/team
- Number of rounds: 4 - 6 rounds
- Time allowed: 2 hours to 5 days
- Administrators: 2 people
- Game control: Computer centralised control with manual interface

Recommended levels of features of an EBM game:

- Realism: A good balance must be maintained between realism and education objectives. Random factors should not be incorporated unless necessary
- Complexity: Various levels of abstractions
- Interaction: The degree of interaction should depend on the nature of the functional areas being covered.
- Focus: Subject area: Engineering Business Management
  Use: Educational
- Entertainment: Business games are by nature entertaining and there is no need to add in any motivation factor.

CHAPTER 6

IDENTIFICATION AND RECTIFICATION OF WEAKNESSES COMMONLY FOUND IN ENGINEERING BUSINESS MANAGEMENT GAMES.

6.0 Introduction

Business games have a lot of advantages but also possess a list of drawbacks. During the review of the existing business games, a lot of weaknesses have been discovered. However, most literature only list out the benefits of using games for teaching and overlook the drawbacks. For example, in the most up-to-date comprehensive handbook of game, 'Handbook of Management Games' (Elgood 1993), drawbacks of games have never been mentioned. While opponents of games are yelling at these drawbacks, many game designers try to avoid the issues and hopping that they will be cured by themselves. The weaknesses can never be rectified if the game designers do not face the problems directly. Subsequently, the potential of business games can never be fully discovered. From the past experience and literature finding, a list of weaknesses are compiled and most importantly, possible ways of rectification are also examined and discussed. In general, there are two types of weaknesses:

1) Nature

Every teaching medium by nature has its own strong points and weaknesses. These weaknesses may never be rectified completely by any means though a good design may minimise the consequence.

2) Design Flaws

Most weaknesses in existing business games are design flaws. Either some important issues have been overlooked or overemphasised which may bring adverse effect to the education objectives.

However, in this chapter, the weaknesses are categorised in two different ways: 'Common weaknesses found in simulation games' and Common weaknesses found in engineering business management games'. It is because some weaknesses are more general and some are more specific.

6.1 Common Weaknesses Found in Simulation Games

6.1.1 It is Not Real!

The obvious weakness in simulation games is that everyone knows nothing is really at stake. Though much effort is put in to make a simulation game to look real, it is only a 'simulation' of the business or engineering environment. Compared, for example, to Russian Roulette with its real pistol and bullet, playing a business game is like putting a toy gun to one's head (although one may still get the feel of the fingers on the trigger). This weakness obviously originates from the nature of business games.

6.1.2 Availability and Suitability

One of the reasons why business game is not that popular compared with other teaching media, like lectures and case studies, is because it is very difficult to get hold of a suitable game for the intended purpose.

First of all, unlike preparing for a lecture that the lecturer has the total flexibility to choose or edit the material that he wants to teach in a course. Most commercial available games do not provide the flexibility on choice of functions and the logic of computerised games are usually 'black boxes' because commercial products always have to protect their copy rights. Due to the unavailability of the complete logic, the game administrators sometimes have the difficulty of understanding how the games work. For example, the total management game 'Stratplan' (Hinton, Smith and Daniel 1985) had been run several times in Warwick Manufacturing Group, Warwick University in teaching their MSc. students. However, the element, 'Shareholders value', was never been understood by the tutors. So, at the end of the day, the students were told to ignore the element which, however, was still shown on the report.

Secondly, a lecturer can easily walk into a library to select and preview a case study for a course but it is rather difficult for him to judge the appropriateness of a game though game handbooks are available. Accessibility to the existing games is a serious problem.

Suggestion:

These problems have de-motivated a lot of lecturers to use games for teaching if they have not used them anywhere before. However, the following recommendations may solve the problems.

Firstly, the university libraries should investigate on the possibility of setting up a game section so that lecturers can have free access to the existing games and thus increase the chance of locating a suitable one.

Secondly, a lot of 'good' games are still stored on the designers' shelves and never published. The game societies should encourage the game authors to publish their games through conferences and journals so that more good games are available for education.

Thirdly, it is true that every business game has its own objectives and the scenario needs to be pre-set. However, within the scope of the game itself, game designers may consider providing games with various levels of complexity.

Finally, copy right problem is in fact a dilemma. In order to protect the game logic from being copied by others, the potential benefit of most games cannot be fully revealed. However, business games are built for educational purpose. Fortunately, the development of a standard interface will solve part of the problems as the lecturers will have the freedom to integrate different functional games together to achieve a better educational effect.

6.1.3 Time and Money

Business games can be expensive in terms of development time, time taken to play and learn. Moreover, the cost used to train managers can be enormous when the objectives of the game are not met. Without any firm justification on the effectiveness of games, to be on a safe side, educators and training managers may use a more traditional teaching method.

Suggestion:

Education is always expensive and time consuming. The issue is whether the time and money spent, and the messages getting across to the participants are well balanced. In fact, a large portion of game time are spent in 'warm up'. According to Watson and Blackstone, 1989 "*Significant learning does not occur in games until all game rules and descriptive facts are mastered and underlying game concepts emerge*". This warm up period may last up to several hours. A game with various level of abstractions should be designed so that

(1) Development time and cost can be averaged out.

(2) Time to familiarise the package by both the users and administrators can also be shortened.

6.1.4 Literal Transfer

There is a danger that there will be literal transfer of unrealistic aspects to business practice. This is especially true for participants with limited training. Stewart (1961) reported that even experienced participants had come away from game playing sessions with distorted views of reality. There was the case of the manager who believed in a certain approach to marketing irrespective of market conditions. By chance, the game he played happened to suit his tactics, and his team won. The executive left the course with all his prejudices further hardened, convinced that the success of him in the game showed that he had been right along, while all that had really happened was that his tactics had worked in that isolated case (Tait 1970).

On the other extreme, players may have the impression that the experience obtained is not applicable to the real environment since they are only playing in a simulated environment. Therefore, they may not pay any attention to what they have learned.

Suggestion:

To avoid literal transfer, it is essential that games are used only in conjunctions with conventional teaching devices. According to Stewart 1961: *"Lectures, discussion sessions, and critiques can alert the game player to the artificiality of the assumptions in the model and help him to discriminate wisely between what can and cannot safely be applied to real-life situations"*.

In fact, Mckenney 1967, recommended that games should not be used without an integrative effort. He stressed the importance of administrator involvement and intelligent counselling: *"Without these two ingredients, the game may be fun and absorb a good quantity of time, but will not be a worthwhile experience to the participants"*. Participants should be debriefed at the end of the game play to limit any unsuitable application of game results.

6.1.5 To Win Not to Learn

While games are used to arouse interest, there is some evidence to indicate that students become more interested in the game itself and not in the subject matter that the game represents. In addition, competitiveness involved in the game may obstruct the learning process. That is, they play to win, to have fun instead of to learn.

Suggestion:

Although competition is necessary in a game because it is the nature of a business world and it definitely serves as a good motivator, however, it should not be over-emphasised. Two elements can be incorporated in the design to minimise the degree of competition:

(1) A game is not necessarily having only one winner. Winners should be defined as the ones who can achieve their defined objectives. In this case, teams will compete against their own standards instead of competing with other teams.

(2) Various roles should be introduced in a game to minimise the head-on competition. The scenario of having all teams starting at the same situation, selling the same product in the same market, in fact, encourages competition. Business games are educational and all teams starting at the same point is not necessary.

6.1.6 Lack of Reinforcement

Opponents of games believe real experiences in the field are much more effective than simulations, and they claim that whatever participants learn from games dissipates quickly after the game is over. This is especially true if the players do not receive reinforcement back in the work place.

Suggestion:

Except 'on the job' training, most classroom teaching methods suffer from this weakness. However, a post game assignment may partly cure the problem. Typical questions in the assignment can be:

(1) Identify and describe what you have learned from the game?

(2) Describe how you can apply what you have learned in your work?

These questions can reinforce the learning process and the assignments are particular helpful for company executives who work on a real business, however, for full time students, case studies may have to be provided to them in order to answer those questions.

6.1.7 Fabrication

Since games are fabricated by people, the assumptions of a game's designer often distort reality or project bias. That is, games have all the misconceptions and flaws of the designers' thinking.

Suggestion:

Design itself is a personal task and it is impossible to design a purely 'unbiased' game. However, two ground rules are suggested to avoid the extent of bias in design:

(1) An educational game should be built upon on the well-established text book theories.

(2) High degree of freedom should be given to the game administrators to set the scenario parameters.

Moreover, it is hard to measure bias which is quite subjective. A person's bias may be the other person's truth!

6.1.8 Unstated Objectives

Some game administrators treat business games as a sort of treasure hunt exercises that the participants have to dig out the messages and concepts by themselves. Although an after-game debrief session is definitely helpful to clear up the objectives, it may be too late for some students who may have overlooked the objectives when the game was played and there is no second chance for them to take a closer look.

Suggestion

Research has shown that the learning process can be enhanced if the students are told in advance the education objectives of the games. It does not mean that the administrator has to give a 'lecture' before a game is run, however, the participants at least should know what they are looking for. For example, in a production game, the administrator may want to remind the participants to pay attention to the relationships between 'lead-time', 'capacity' and 'inventory' level.

6.1.9 Thickness of a User Manual

It is uncommon to find a game user's manual of less than ten pages and in fact, most of the business game user's manuals are more than 50 pages. However, the players rarely be found to read through the whole manual before the game is actually played. The thickness of game manuals sometimes scares the participants. In fact, some game administrators suggest that the players manuals should not be more than 5 pages but the game may be so complex that the game designer does need fifty pages to describe the game to the players. There comes a dilemma.

Suggestion:

If a game manual is too thick (more than 10 pages), it should be divided into different parts. Typical sections are:

Users manual: It provides enough information for the participants to start the game. No details or advice will be given.

Reference manual: It provides detail and advice on each element and function of the game. So, the players can look up the required detail when necessary.

Quick Reference: It lists out the decisions and their meaning for quick reference purpose.

6.2 Common Weaknesses Found in Engineering Business Management Games

6.2.1 Too Simple or Too Complex

In order to cover the whole engineering business management which mainly composes of marketing, production and finance, most total management games simplify each functional area in order to make the game manageable and hence a lot of details are missing. On the other extreme, in order to give a more realistic picture on each functional area, some total management games have a total of more than 70 decisions per round which may de-motivate the students. (Kidd 1975)

Suggestion:

EBM is always complicated. One of the ways to solve this problem is to provide a game with various levels of abstractions so that the students can get familiar with the complexity gradually.

6.2.2 Kill to Survive

Most total management games have all their teams starting from the same settings, producing the same products and competing in the same markets. This set up places heavy emphasis on the competitiveness which no doubt exists in the actual business world. However, a successful company needs not only to beat your rivals but also to form your alliance. For example, one of the most important factors in implementing 'Just In Time' philosophy is to have good relationships with your suppliers. In fact, the primary objective of a business game is to 'learn' not to win, there is no point for all the teams to play the same role and try to tear each others apart. Moreover, 'killing' does not mean winning because the survival one may also be seriously hurt.

Suggestion:

Different roles should be introduced in the game so that the participants can learn to co-operate with others. For example, a 'supply chain' structure will enable the participants to play different roles in a game. Since teams will be divided into suppliers and manufacturers, they not only have to complete in their own circle but to form allied with the others.

6.2.3 Discourage Originality

Another criticism of business games is that they discourage originality among players. Decision outcomes mimic those which usually occur in business practice; but in reality, daringly original policies sometimes pay off. (Stewart 1961)

Suggestion:

Business game is a simulation model which is guided by rules and logic. All possible movements or decisions are restricted by the design. Case study in this case is more superior to game. This weakness comes partly from the nature of a business game and the problem is more obvious in a computer business model.

For example, in most (if not all) existing games, the number of products and the nature of the products are fixed. Though some games do allow the introduction of a new product during the running of the game, that 'new' product is pre-defined by the game authors. For example, in a game of producing tape recorders, companies may allow exploring the compact disk market but any other alternatives, like going into the digital tape recorder market, is prohibited.

This particular weakness can be partly rectified by incorporating some 'unknown' or 'empty' elements in a game. That is, some 'space' have been reserved in the design to incorporate 'new' ideas. For example, 'new products' and 'new features in an existing product can be added during the game. Firstly, the teams are free to develop any new products in the same market and the game administrators can decide how these new products will affect the market behaviour and the production capacity, etc. Then, the administrators can input these new products' parameters in the game during the run. Since these new products' ideas are not fixed or pre-programmed by the game authors, teams are encouraged to think with originality.

With the same token, new product features can be 'created' by the teams on the existing products during the game. Some 'empty' features have been reserved for each product and the teams can fill in the blank. How these new features affect the sales and production are once again decided by the administrators. In fact, this 'empty element' idea can be incorporated in different parts of a game. Certainly, more work are required by the game administrators but as long as the education objectives can be achieved more effectively, the work is definitely worthwhile.

6.2.4 Human Factors

Most total management games overlook human factors in modern manufacturing management. Although most educators agree that human factors are important in EBM, most business games emphasis heavily on analytical skill which most participants have already possessed. Moreover, most middle and senior management's work do not require too much analytical skill.

Suggestion:

Some game designers are aware of this and suggest to play games in teams so that team members can interact with each other. It does partly solve the problem, however, human factors and communication skill do not just apply to your colleagues but also to your competitors and suppliers. Negotiation with other companies and signing contracts are vital parts in a business world. According to Dr. Myron Uretsky of New York University, one of the game developers: '*Business is not really made up of decisions or mathematical models; it is people*' (Maital & Morgan 1988).

Naturally, the 'supply chain' element introduced in the previous discussion can be further elaborated to incorporate the role play element. Contract negotiations become face to face activities between companies (teams). So, if a team (manufacturer) wants to purchase some parts, the team members have to negotiate the deal with the other team which manufactures the parts. A lot of interesting issues should emerge from this set up like breaching of contracts which will add in more realism in the game.

6.2.5 Easy Money

In a finance function of a total management game, the students should learn how to balance their cash flow and utilise their cash resource in a most effective way. However, most of the management games fail to convey the message. Fundamentally, there are usually two ways for participants to raise money in a game:

Some game designers allow the participants to sell and buy shares of their own or even other companies in the stock market and this arrangement is perfectly acceptable if it is a financial game. However, in an EBM game, it may bring adverse effect to the teaching objectives. Participants may try to make profit from the stock market instead of managing a company. In an EBM game which tries to highlight the integrated idea, the existence of 'share' element may disturb the objective.

Another way of fund raising is to borrow money from a bank. Again, there are two common ways for a game to handle 'borrowing': limited borrowing and unlimited borrowing. In 'limited borrowing', companies are allowed to borrow an amount according to their past performance (usually profit) or existing assets. Hence, a winning team can always borrow more than a losing team. This may be true in a real business world but this is an educational game. This arrangement never allows a losing team to catch up even though they may come up some good strategies to reverse the situation. As a result, the losing teams lose interest on the games and made random decisions because they recognise that they do not stand a chance anyway. Some total management games even allow the happening of bankruptcy and take-over. Certainly, this is a tough life but the game designers have gone too far. They sacrifice the education objectives for realism. What are the bankrupted team members going to do for the rest of the game -- unemployed?

On the contrary, some game designers allow unlimited borrowing. Teams usually may have to pay different interest rates according to their assets and amount that they borrowed. This arrangement has a serious drawback. Cases have been seen that teams compete with each other on the advertising expenditures which in fact several times more than their sales volume. Since money comes in too easy, they do make ridiculous decisions.

Suggestion:

The above problems may be solved by having a 'bank'. This bank may be one of the teams or the game administrator himself. When a company (team) wants to borrow money, an investment plan or a proposal has to be drawn up and submitted to the bank for assessment. The bank may reject, accept or lend out less amount depending on the quality of the proposal. In here, some equations or formulae used by the credit centres may be helpful for the administrators in making credit judgement. Interest rates and repayment terms are negotiable between teams and the 'bank'. (another role play element). After all, a bank is also a profit making company. This arrangement force the participants to consider carefully before they spend. At the end, critical thinking is what the game wants to teach.

6.2.6 Too Quantitative

Manufacturing games are usually too quantitative and they usually ignore the qualitative factors such as the quality of decision. It is because a lot of production and marketing techniques are mathematical models which can be easily programmed by computer. For example, most of the games assumed higher expenditure on advertisement will increase the sales volume. In reality, this may not be the case because it also depends on the quality of the advertising campaign and numerous other factors.

Suggestion:

It is always a draw back of simulation when qualitative factors and human factors have to be incorporated. However, qualitative decisions may also emerge from a combination of quantitative factors if the environment is realistic enough. For example, sales is a function of 'quality', 'price', 'advertising', 'goodwill', 'features' and 'competitions'. Quality is then a function of 'R&D', 'maintenance of machine', 'Investment of technology' and so on. Some functions follow the law of diminishing return and some do not. Although all these relationships will be represented quantitatively, the complex relationships between them should be able to show the participants that an increase in advertising alone may not increase the sales directly. A set of well-balanced decisions is important to increase sales and this message is ***qualitative.***

Another important design consideration is how to avoid the participants play against the formulae in the game instead of making decisions qualitatively. In order to avoid that, a game has to be reasonable realistic and complex and most importantly, as long as the participants are making decisions on the right route, the difference between a dollar or two in a decision entry, such as the price or advertising should not make a big disturbance to the result. Thus, the participants will not waste their time to dig out the 'magic number' on a decision but to analyse the situation with a integrated approach.

6.2.7 Information Overload

Participants are often overloaded by reports after each round. It is not uncommon to find over ten reports (Production summary, production schedule, inventory report, market research, profit and loss account, balance sheet, customer buying habit, etc.) are provided to the participants after each period and each report may have three or four pages long. Yes, this is reality. Managers nowadays are bombarded with an ever-increasing volume of information, both formal and informal. But does this kind of realism contribute to the education purpose?

Suggestion:

Research has been carried out on the effect of amount of market information acquired to the decision making process. The results showed that in a relatively unstructured decision making process, excessive use of data may serve a useful purpose in the early stages of decision-maker learning. Clearly, though, the answer is not to begin by giving them everything. Self-selection of required data with a significant cost per item charge seems to stimulate an analytic process that promotes more rapid decision-maker learning while avoiding information overload. This analytic process was prescribed by Ackoff 1967 and conducted not by decision science experts, but by decision makers themselves, as Dearden 1972 would advocate (Chorba & New 1980). The research demonstrated that there is considerable potential value in structuring an information system in a manner that motivates analysis by the decision maker. Allowing the decision maker to select his own data under economic constraints is apparently one way to motivate such behaviour with positive effects on performance.

So, in order to attain the maximum effect on a game, only a few basic reports should be provided automatically. Other reports should be purchased at a high cost with the arrangement that some reports may be missing or biased. There are three reasons for this arrangement:

(1) In a real business world, information is not easy to obtain even though one is willing to pay.
(2) Data obtained may be biased and careful interpretation is required
(3) If the reports are highly expensive, the participants will consider carefully before they purchase the report and examine them in detail after receiving them.

However, it ought to be stressed that acquisition of market information is obviously not the only factor of explaining success. On the contrary, this variable should be an intervening variable expressing the behaviour of those teams which successfully manage to use market information. Along with the suggestion of Dearden, one could conclude that if a team has the ability to convert information into high quality decisions, then they credit more from acquiring information than those teams which does not have this ability and, hence they do acquire more information than their competitors. Accordingly, the success in business games seems to be a question of ability and rational behaviour (Lerviks & Paltschik 1982).

6.2.8 Changing Internal Environment

Most total management games are played in teams. A team may consist of any number of people. One of the advantages of playing a business game is to develop team work or team spirit. Although most game designers suggest to divide the job in a team to build up a small organisation, this does not always happen especially in a computer interface game. Anyway, natural leader always emerges in a team to guide the discussion but it is always the same leader and same team members through out the whole game. Unfortunately, in a real business world, people do switch from one job to another. One of the most important management skills in a business world is to get along with your new boss as well as your new colleagues. Most management games overlook this factor.

Suggestion:

Business decisions happen against a shifting background of economic, social and technological changes. However, a changing environment inside a company is equally important. In fact, one may face more pressure from the changes of the personnel inside a company than the changes of the outside market. This feature may by introduced by swapping team members and team leaders between groups in the middle of the running. Since this is a manual function, the game administrator has full control on the distribution of participants and the timing of the swapping. This arrangement will also raise a lot of interesting issues in human factors as well.

6.2.9 Time and Pressure

One of the distinct advantages of using business games is it can simulate the pressure from the environment. Time is limited in a classroom environment. In most management games, the participants always find that they have too little time and too much work. It is good because it does happen in the real world. Also, it can force the participants to select the important information and issues to address on. However, most game designers create scenarios which are so complicated that it is impossible for the participants to make reasonable judgement within the allocated time. Since the available time is too far from enough, the participants sometimes just give up the work and make random decisions because they do not think they can make any sensible analysis.

Suggestion:

Too little time to make too complicated decisions will only force the participants to guess and this definitely de-motivates some participants. If a game can firstly be introduced in a simple format and gradually increased the complexity, this may partially solve the problems.

On the other hand, in a business world, executives have a lot decision support aids to help them to make decisions. For example, they may use a forecasting model to forecast the demand. However, most of the time, what the students used are only calculators, pencils and paper. It is not fair. If they have to face the same complexity as in the real world, they deserve the same supports.

There are cases that some game authors use the game model itself as a 'decision support tool' which can provide a what-if analysis as in 'Stratplan'. In this case, participants can just randomly vary their decisions in the what-if model to maximise the output without knowing why. If the game model is used to simulate a real business environment, how can one use the 'real business environment' as a decision support tool. This is a fundamental error in design. Strictly speaking, the participants should only be provided with the commercially available decision support tools which are independent from the game itself. However, to make the life easier, most data and reports of the game should be presented in a spreadsheet format.

6.2.10 Number Crunching

Number crunching is common in business games. The participants are always looking for a magic combination of decisions that can make them win. They are not playing a business game. They are in fact competing on an mathematical IQ test - to see who can first guess the answers. Most game designers put too much emphasis on mathematical details and overlook conceptual issues. Even if the game is built on an operational level, not much can be learned by just varying the transfer batch size between machines in a 'Just in time' production game.

Suggestion:

Since the target players are master students, middle or senior company executives, number crunching does not teach them a lot and in fact, optimal solution hardly exists in a real world. For example, in teaching 'Just in time' philosophy, besides the culture factors, the main idea is to have the right amount of high quality material/parts at the right time at the right place. Unfortunately, in a 'Just in time' game played in Warwick Manufacturing

Group, the most important elements like 'supplier-manufacturer' relationships and 'quality' are ignored in the game. It is impossible for a student to understand the JIT philosophy by just varying the transfer batch sizes between machines. A good game should provide an environment for the participants to understand the philosophy and acquire the appropriate concept instead of doing endless manipulation on numbers.

6.2.11 Equal Pay Off

One of the reasons why some people do not like game is because they think game is unrealistic. In a business game, all customers are equally informed; the result of equal effort is constant over time; there are no missed deadlines, transmission error or unexpected deaths of key personnel; etc. And in a real business world, all these happen. (Stewart 1961)

Suggestion:

The comments made by the opponents of games are absolutely true. However, it should be borne in mind that alternative teaching methods are similarly afflicted. Some of the problems mentioned above may be solved by the appropriate actions of game administrators.

For example, the game administrator can transfer participants between teams during the running so that during some rounds, some teams may have less members than others. Also, the removing of the 'natural' leader in the team should bring the similar effect of 'the leave of the key personnel' in a company. Also, a game should be able to produce different versions of a marketing report to different teams as requested by the administrator. In this case, teams will not be equally informed. In addition, if random events may be triggered by the administrators, 'equal-pay-off' is not necessarily true.

In conclusion, common weaknesses discussed above are extremely valuable in design of EBM games. They can serve as a reference for game authors in this subject area in the future.

6.3 Conclusion

Most of the weaknesses can be rectified by the game authors. The findings is summarised below to give a guideline to the game authors in design of EBM games.

Table 6.1 Summary of weaknesses in EBM games

Weakness	**Recommended ways of rectification**
It is not real	Nature
Availability and suitability	Game library should be set up Game logic should be released
Time and Money	Standard interface should be set-up
Literal transfer	Used in conjunctions with lectures and discussions
To win not to learn	Multi-winners and various roles in a game
Lack of reinforcement	Post-game assignment
Fabrication	Based on well established textbook theories High degree of freedom in setting scenario parameters
Unstated Objectives	Pre-game session
Thickness of a user manual	Several game booklets
Too simple or too complex	Various levels of abstractions
Kill to survive	Various roles are introduced
Discourage originality	Introduction of empty 'features' and 'elements'
Human factors	Incorporate Supply chain structure
Easy Money	Application of a 'bank' loan
Too quantitative	Qualitative inter-relationships between element
Information overload	Reports purchased at high cost
Changing internal environment	Swapping members between team during the game
Time and pressure	Available of decision support tools
Number crunching	Avoid magic numbers
Equal pay off	Reports with acceptable error

CHAPTER 7

EVALUATION OF THE GAME DESIGN - USERS' AND ADMINISTRATORS' OPINION

7.0 Introduction

A 'good' business game is one which can achieve its objectives effectively. However, if the users or administrators do not 'like' it for whatever reasons, the game can never be good. Although opinions from users or administrators may be subjective due to their difference in backgrounds, their views and expectations are always valuable in the design and evaluation process. During the last three years, two questionnaire surveys and several interviews have been carried out to obtain the potential users' and administrators' views on integrated business games. Since Warwick University is one of the leading universities which emphasis on integrated approach, the students and staffs there especially in the two leading management departments (Warwick Manufacturing Group and Warwick Business School) should form good representative samples of the potential users' and administrators' populations. Specifically, sample users were randomly chosen from the Warwick Manufacturing Group master degree students. Sample administrators were randomly selected from staff in Warwick Manufacturing Group (WMG) and Warwick Business School (WBS). In analysing the survey results, their backgrounds and experiences in using business games have also been taken into account. Naturally, less weightings are given on people with limited game experience. Overall, there were two questionnaire surveys and one interview survey.

Survey 1: Views from Experience Users

Questionnaires were handed out to 100 randomly chosen master students in WMG to obtain their views and expectations on business games and a response rate of 71% was obtained. Since this survey was carried out at the end of the academic year (92/93), most students in WMG had already gained some game experience from their MSc. courses.

Survey 2: Before and After Game Survey

After the first survey, some areas were required further investigation. Hence, a second survey was performed. This time, 36 students were invited to participate in the experiment at the beginning of an academic year (93/94). The students were divided into two groups to play two games with different levels of complexities (medium complex and highly complex). Questionnaires were filled in before and after the games. Unfortunately, part of the results were found to be biased after the survey.

Survey 3 Interviews of Subject Tutors

More than 15 subject tutors from WMG and WBS were interviewed. Normally, there must be some reasons in their minds for using or not using games in their teachings. Since the sample size is too small to perform any significant tests and in order to avoid any restrictions on their answers which were assumed to be more valuable than the previous surveys because of their subject expertise and experience in education, open questions were asked instead of questionnaires being filled in .

7.1 Survey 1: Views From Experience Users

7.1.1 Objectives of Survey

The objectives of the survey were to obtain the views from users on two areas:

1) Their preference on formats and features of games

2) Their views on using business games as a teaching tool

7.1.2 Design Structure of Questionnaire

A sample questionnaire is attached in appendix I. The design format of the questionnaire was structured to obtain three key areas of information:

-Users Information

Question 1, 2, and 3 were designed to extract information on the levels of game experience from students. Most participants were expected to have reasonable experience on playing business games from their MSc. courses since the survey was held at the end of an academic year.

-Preference on Formats and Features

Question 4 to 7 were designed to extract the users' preference on formats and features of games to confirm the findings from previous chapters.

-Use of Business Games as a Teaching Tool

At last, students were asked to compare the effectiveness of different teaching media in different stages of teaching (Introduction, Revision, Integration). Although previous research findings had shown that business games are more appropriate to be used as an 'integrated' tool, users opinions are always welcome.

7.1.3 Sampling

A total of 184 full time students were available for the academic year 92/93. Of 184 full time students, 61 (33%) students were major in Engineering Business Management (EBM), 87 (47%) students were major in Information Technology for Manufacture (ITFM), and 36 (20%) were major in Manufacturing Systems Engineering (MSE).

100 students were randomly selected for the survey and responses were received from 71 students giving a response rate of 71%. Among the 71 responding, 31 (43.7%) was from EBM, 26 (36.6%) were from ITFM, and 14 (19.7%) were from MSE. Since the distribution of these three streams in the sample was reasonably closed with the original distribution, these 71 students should have formed a good unbiased representative sample.

7.1.4 Result Interpretation

7.1.4.1 Users Information

Of the 71 responding, 65 students (92%) indicated that they had some business game experience. (Table 7.1) 37 of them had experience on marketing games, 47 on production games, 60 of them on financial games, and 15 on project planning games. On the average, at least 2 games have been played per student.

Table 7.1 Types of games played

Type of Games	Game Experience
Marketing	37 (52.1%)
Production	47 (66.2%)
Financial	60 (84.5%)
Project Planning	15 (21.1%)

Since each student in WMG has to take twelve modules in a year, hence around one out of six modules (17%) used games as part of their teaching media during 92/93 academic year if all the students' game experience were obtained from the MSc courses. (Later on, survey 2 confirmed that students obtained almost all their game experience from their master courses.) However, most of the games played were 'functional' based. Furthermore, the only integrated game used in the group, 'Stratplan' in 'IGDS Business Policy module', has been taken out from the module lately. The module tutor explained that:(1) there was not enough time (2) the game was never fully understood (3) the game was not fool-proof. e.g. one typing error in the decision entry could mess up the whole game. In fact, these weaknesses have been discussed previously.

7.1.4.2 Preference on Features and Formats

7.1.4.2.1 Manual or Computerised

Table 7.2: Process format

Process Format	No. of Respondents
Manual	29 (43.3%)
Computerised	26 (38.8%)
Either	12 (17.9%)

29 (43.3%) students preferred manual games to computerised games. They believed manual games were more interactive (between team members), more realistic, more easy to participate and understand. On the other hand, 26 students (38.8%) preferred computerised games because they thought computer games were more user friendly, more interesting, provided more fun and faster feedback. The others (12 of those) did not have any preference. They believed manual game could provide more participation among members but on the other hand, computerised games could avoid tedious calculation involved. The results do agree with the previous findings that computers divide people and manual interface can increase participation and interaction between participants.

7.1.4.2.2 Important Features in Business Games

Top five features in business games were ranked and the results are summarised in percentages in the following table:

Table 7.3 Considerations in design of business games

Priorities	1*	2*	3*	4*	5*
Education	42.9%	18.0%	16.1%	16.4%	14.1%
Entertainment	17.5%	16.4%	6.5%	11.5%	23.4%
Interactive	4.80%	16.4%	29.0%	18.0%	15.6%
Non-interactive	0%	0%	0%	0%	0%
Realism	19.0%	16.4%	21.0%	9.8%	9.4%
Fictional	3.20%	9.8%	6.5%	24.6%	12.5%
Team-Working	12.7%	23.0%	21.0%	16.4%	16.5%
Singer player	0%	0%	0%	3.3%	9.4%

(* 1 = most important element, 5 = least important element)

The following list is obtained by putting the features in the order of priority one:

1) Education	(42.9%)	5) Interactive	(4.7%)
2) Realism	(19.0%)	6) Fictional	(3.5%)
3) Entertainment	(17.5%)	7) Single player	(0%)
4) Team-Working	(12.7%)	8) Non-interactive	(0%)

Undoubtedly, 'Education' is the main purpose of the development and 'Realism' has been well recognised as the most important feature in a business game. Most students do not notice that business games are by nature entertaining although the feature may be further enhanced by introducing non-educational elements which however may add in unnecessary complexity. Team working itself is always a good motivator. The findings are encouraging because the participants seem to know what are important in business games. However, the findings also re-confirm one of the common drawbacks of games -- they play to win instead of to learn. From the survey, no students prefer to play 'non-interactive' games which involve no competition between teams. However, non-interactive games do have their own advantages and they are particular suitable in some subject areas such as shop-floor scheduling.

7.1.4.2.3 Duration

Table 7.4 : Game duration

Hours	Expectation
0-2	15 (21.4%)
2-5	43 (60.0%)
> 5	13 (18.6%)

One of the major weaknesses of business games is time consuming in terms of both development and playing. Only 18.6% of participants prefer to play games more than 5 hours. This is quite contradictory with the literature findings that more than 50% of the existing games designed to run more than one day. However, the length of a game depends on too many factors and it is impossible to standardise them. Furthermore, since business games are different from the traditional teaching media which the students have experience for tens of years, participants may probably require more than two game experience to realise the full benefits of games. So, although students do enjoy business games, they may think they can learn more by attending lectures for the same amount of time.

7.1.4.3 Effectiveness of Business Games in Education

A 'five' points weighting system was used to analyse the effectiveness of different teaching tools in management training in different stages of teaching. 5 points for most effective, and 1 point for least effective. Since the game process format (manual or computerised) might affect their weightings, they were itemised separately. The overall system is summarised into average ratings as follow:

Media\ Average Rating	Introduction	Revision	Integration
Lectures	4.4	3.36	4.2
Seminars	3.06	3.57	2.94
Videos	3.06	3.36	3.15
Case Studies	2.70	3.99	3.99
Computerised Games	2.34	3.57	3.57
Manual Games	2.34	3.15	3.15

Table7.5: Effectiveness of the learning tools during a) Introduction b) Revision c) Integration

Most students still think that lectures are the best tool for initial teaching of business management knowledge and no other teaching method can even come close on the scale. May be that is the reason why lectures are still the most commonly used teaching method in universities. On the other hand, business games are thought to be ineffective in introduction of a subject which is quite true as discussed before.

During the revision stage, all teaching methods were ranked closely between 3.0 to 4.0. It is interesting to find that besides lectures, all teaching methods received higher rating than in 'introduction' stage. In particular, the ratings of case studies and business games had increased by more than one point scale from before. These findings once again confirm that both case studies and business games are good integrated tools. The superiority of case studies over business games can be explained by the longer history of existence in education. The other two teaching methods, videos and seminars seem to be quite consistence through out all three stages. However, it is surprised to find that most students still prefer lectures to case studies and business games in the integration stage of the learning process. From this survey, one can estimate that lectures will probably remain as the sole teaching method in the education system in the near future.

However, the students do recognize the importance of other teaching tools at the revision and the integration stages. In fact, lectures by themselves provides little help in transferring complex and dynamic knowledge to students. Lectures need the assistance from others tools such as videos, case studies, and business games to revise and integrate the knowledge learned from lectures. The survey also shows that the participants did not distinguish too much between computerised games and manual games in the ranking of effectiveness.

Overall, students still believe lectures are the best way of teaching. All other teaching methods in fact did not make too much difference in their minds. In general, the findings are quite similar to the usual teaching practices in universities. Lecturers usually start a subject with series of lectures with seminars and videos in between. At the end, case studies and business games are introduced to integrate the knowledge and then lectures may be given again to summarise the teaching. Nevertheless, 92% of students believed that business games are an effective tool and 95% of students would like to have business games as one of the ongoing training programs in their courses. In short, the findings do support that business games are as effective as other teaching media in education. Other comments on business games include:

- Enable high participants involvement.

- Realism is crucial.
- Enable to practice planning, communication, and team-working.
- Enable to integrate the knowledge learned from lectures and use them practically.
- Should not be too complicated.
- Interesting and enjoyable.
- User friendly, easy to operate, graphical interface, and clear instructions are important

In inclusion, the results imply that students are in fact expecting a variety of ways of learning although they still believe that lectures are most important in teaching. Besides education objectives, four most important and welcome features recognised by the students in business games are 'realism', 'team working', 'entertaining' and 'interaction' which mostly agree with the previous findings.

7.2 Survey 2: Before and After Game Survey

The objectives of the survey were to study the:

1) The appropriate level of complexity in a game

2) The subject areas that business games can teach more effectively

The survey was broken down into two parts: Before game and after game.

Before game: The participants were briefed on the nature of business games before questionnaires were filled in. (Appendix II). The students were then divided into two groups to play two different integrated games. Group 1 (19 people) played Game 1 (highly complex game - 35 decisions) and Group 2 (17 people) played Game 2 (medium complex game - 20 decisions). Each group was composed of four teams with four to five members each. Both games were played for four rounds which lasted for 5 hours with the present of two administrators. In order to avoid any unknown logic in the games, both games were products of previous master projects in WMG.

After game: Questionnaires (Appendix III) were filled in again by the participants. The purpose of this arrangement was to study their views on different complexities of games.

However, due to the timing of the survey, all students who have gone through the first survey had graduated. Fortunately, 36 students from current academic year were willing to participate in the survey. When this survey was planned, the students were assumed to have some game experience in their undergraduate studies which however, turned out to be completely untrue. In addition, several drawbacks were discovered during the survey which might also lead to biased results.

7.2.1 Biased Survey

There are several reasons to lead to believe that survey 2 was biased. The main problem was the incorrect assumption that the students would have some game experience from their undergraduate studies. However, after the questionnaires were collected, almost all students were found to have no game experience and some of them have not even heard about education games before. So, their views on games are untrustworthy. In this case, the after game survey results would be predominately affected by the quality and the structure of the games which they have just played. If the games were bad, they all got bad experience on games and hence the results are biased.

Also, during the survey, when the participants discovered that the games were in fact the work of previous students, some of them began to show negative attitudes and they criticised on every minor detail of the games. Furthermore, since the games were not taken place inside a module, the students knew that they were playing the games because of the survey. They did not behave seriously. To a large extent, this might affect their response because they should have played the games for some education objectives.

Besides, the students were found to be frustrated when the games were played. It seemed that they had difficulties of understanding the engineering and business jargons and theories although most of them were graduated with engineering or business degrees. One possible explanation is that they were trained by traditional teaching methods and they were not used to 'apply' what they have learned in real practice. Running an integrated games may be too heavy for newcomers.

However, some of the findings which do not directly relate to game experience are quite interesting. Although these findings do not contribute too much on the design process, they do give some hints on the use of games as a teaching tool.

7.2.2 Interpretation of the Results

Since almost all students were found to be ignorant on business games, most of their views on games are ignored. As a result, the interpretation approach of the results is modified and also, the data in survey 1 is used for comparison.

Assuming the intakes of WMG of 92/93 and 93/94 were compatible and they belonged to the same population, there were three different sets of data representing three stages of the students:

1) Sample with two or more game experience (Survey 1)

2) Sample with no game experience (Survey 2 -- Before Game)

3) Sample with one game experience (Survey 2 -- After Game)

Comparing the survey results in sample 2 and 3 is probably misleading. However, some comparisons made between sample 1 and 2 is acceptable so that the changes of their attitudes can be studied. In the following, only selected data which are game experience independent are studied. The rest of the data are attached in Appendix IV for reference.

7.2.3 Users Information

There were 189 master degree students studying in the WMG during 93/94. A sample of thirty six students should have formed a good representative sample (Table 7.6). The distribution of the three different streams was similar to Survey 1. Since students participated the lengthy experiment voluntarily, they would express their views to the best of their knowledge. Unfortunately, around 90% of students were found not playing any types of education games before (Table 7.7). Since most participants freshly graduated from engineering or business degrees, one can conclude with 95% confidence that less than 21% of undergraduate students in engineering or business disciplines have played business games in their courses.

Major	EBM	MSE	ITFM
No. of students	19 53%)	10 (28%)	7 (19%)

Table 7.6: Backgrounds of the Students

With Game Experience	4 (11%)
With No Game Experience	32 (89%)

Table 7.7 Business Game Experience

Some Interesting Findings on Sample 2

Type of Game	Computerised game	Manual game
No. of Students prefer	33 (94.3%)	2 (5.7%)

Table 7.8 Game Process Format

Before any game is run, 94.3% of students preferred computerised games. This can be explained by a common perception that students like computers. However, comparing with survey 1, students shifted their preference significantly from computerised games to manual games. Statistically, there is a 95% confidence that number of students favoured manual games has been increased by between 42% and 69%. So, participants are sort of disappointed by computer games.

Another interesting finding is most of the students (73%) prefer to have games less than 2 hours. It is quite understandable that if one has not played any game before, he/she would probably like to make it short for the first time. On the other hand, this finding can also reflect that their ignorance on games.

In conclusion, the experiment was a failure. However, two important findings have been learned from the failure:

1) The present usage of business games in undergraduate degree courses is minimum.

2) The set up of a unbiased game experiment is difficult. Especially, when games are needed to play in the survey, the experiments are then potentially biased. That may be one of the reasons why different researchers always obtain different findings in their experiments on the same subject area such as

complexity. This in fact agrees with the previous discussion that 'compatible' environments are difficult to obtain for game experiments. So, though much effort have been used to study the methods of evaluation of games, at the end of the day, the effectiveness of a game is still predominately subjective.

7.3 Survey 3: Interview of Subject Tutors

7.3.1 Introduction

Module tutors or lecturers are the ones who decide whether or not games should be used in their teaching. It is because abuse or misuse of business games in classes may bring disasters to the education objectives. So, although their views on games may be subjective, they can never be ignored. During the last two years, 15 lecturers from WMG and WBS have been interviewed. The length of the interview and the depth of detail discussed vary according to the tutor's game experience. Since the sample size is relatively small, significant tests may not be performed at a high confidence level. Needless to say, the tutors were expected to provide more constructive opinions than the students. Hence the survey was designed in an open-ended format so that:

(a) No restriction would be put on their answers

(b) Additional questions might be asked according to their response

Although a set of questions was prepared (Appendix V), the order and the approach was flexible. Generally, the questions were constructed around four different areas:

- What type of knowledge a manufacturing manager should possess?
- What are the best educational methods of teaching management knowledge?
- Why games are/are not used in their teaching?
- What are the main features of an effective business game.

There are four objectives in this survey:

1) To support the previous findings

2) To understand the needs of the lecturers

3) To discover any tutors' misunderstanding on games

4) To stimulate any new design idea from the interview

7.3.2 Questions and Answers

Q1. Which is the best educational method or combination of methods for management students to understand and practice ' real ' management of a company?

A1: Most tutors believe that practice makes perfect. So, on-job training is the best way to develop one's management skills. In terms of classroom teaching, a combination of lectures, case studies, and seminars given by industrial speakers seem to be most effective. Other teaching methods such as videos and business games may also be used occasionally.

Comment:

There is no such thing called the best teach teaching method or the best teaching combination in management education. Whether a teaching method is appropriate or not depends on numerous factors such as the nature of the subject, the background of the students, the depth of detail covered, time allowed, the education objectives as well as the class size. Combination of lectures, case studies and seminars have been adapted in teaching management subjects in universities for years. Although the results are reasonably effective, it does not mean that other teaching media are less effective and cannot be better in some occasions. A fixed combination of teaching methods to all management subjects may over-simplify the problems in management education. Since business games and case studies are proved to be equally effective, there are reasons to believe that a different combination, says lectures, business games and seminars may bring in better performance in some subject areas.

Q2.(a) In the institution where you teach are there any business games available for teaching purposes ?

(b) If yes, have you ever used them as part of your teaching programme?

(c) Why do / don't you use them ? What are the results obtained ?

(d) What are the advantages and disadvantages of games?

A2: Around 1/3 of the module tutors have not used any games in their teaching programs because they believe that business games are not applicable to their fields (e.g. manufacturing strategy) or they do not have relevant experience to use them. However, 2/3 of the tutors have used games in their teaching and most students seem to like it. A list of advantages and disadvantages of games are compiled from their views (Appendix VI). Certainly, tutors who like games emphasis more on advantages and tutors who dislike games stress on the disadvantages.

Comment:

Most issues mentioned have been discussed previously. Although a number of disadvantages has been raised, no suggestion on improvement or rectification has been mentioned. Most tutors just take the weaknesses as they are the nature of games. It is also important to stress that although most students like games, it can not be used as a measure of effectiveness because games by nature are entertaining. For example, students may enjoy a lecture because the lecturer has raised a lot of jokes, however, they may not have learned anything.

Q3.(a) What functional expertise should a general management manager have?

(b) What a management student should learn before they go out to work?

(c) Which of these functional areas are appropriate to be incorporated in a business game?

A3: In order to manage a manufacturing company, a general manager needs to possess a wide range of knowledge and skills. The list, which is attached in Appendix VII, is endless For management students, they are expected to gain adequate knowledge in six different areas before they graduate: General management, Economics and finance, Strategic and operational management, Business environment, Production planning and control, and Business Strategy. Each area is further broken down in detail which is attached in Appendix VIII. Furthermore, a general manager need not to know the full detail of each department but to understand the inter-relationships between them. Besides 'human factor' and 'technology', most of the issues mentioned can be incorporated in business games.

Comment:

Interview results are always subjective. Since the tutors possess various backgrounds and expertise, they always lean on their own subject areas. In fact, all subject areas are equally important. Trying to weight the relative importance between 'production planning' and 'marketing' is just like choosing between an apple and an orange. Technology is definitely not something that business game can handle. However, most tutors have a misconception that business games cannot handle human factors effectively as they have never imagine to incorporate role play element in a business game.

Q4. What do you think is more important for a business game ? Simplicity or credibility ? Why ?

A4: Arguments which support simplicity in a business game are:

- to introduce new ideas
- to shorten the warn up period
- to avoid de-motivating the students

On the other hand, arguments which support credibility are:

- to reflect a real company
- to train senior managers to face complexity

- to integrate all functional areas

Most tutors believe that although a well balance is important, realism is always the more important one.

Comment

Most module tutors share a perception that realism means complexity and simplicity means fictional. However, these are not necessarily true and sometimes, realism may bring adverse effect to the education objectives. In fact, it is quite difficult to define complexity because different people may find the same game easy or difficult according to their knowledge and game experience. That is why a 'good' game should have various levels of complexity to suit different needs.

Q5. What is the role of the game administrator ?

A5: Only a small number of tutors believe that game administrators are not essential in business games because in real business world, there are no game administrators. Also, games which are specifically designed to be played by one user, do not require game administrators. Fortunately, most tutors believe game administrators are crucial because they can:

- facilitate the learning process
- specify the winning criteria, assesses students, gives feedback (e.g. umpire of the game)
- set and change the constraints of the games
- formulate the groups (choose the right members)
- explain the results to students and make sure they get the right ideas.
- put in the computer the decisions made by the students
- show students how to tackle problems in a logical and systematic way.
- provide confidence to students
- remind students of their roles
- motivate the participants
- help the losing groups to remain competitive

In addition, game administrators are particularly important in manual games.

Comment:

It is happy to see that most tutors recognise the importance of game administrators although some still have the tendency of sacrificing the education objectives for realism. In fact, less than 8% of currently used games are designed for less than 3 people. Furthermore, another major duty of game administrators is to stop the games at various points for small debrief sections so that teams can identify their positions and learn from others (success or failure) before it is too late.

Q6. What features do you think would make a business game more friendly to the user ? How should the interface look like ?

A6: Most tutors consider that the reports of a game does not need to be too transparent since in real life, the reports that managers have to work at are not always very clear However, the operation interface should be kept user friendly especially in computer games which should be structured that computer literacy is not a prerequisite for the users. Elements like windows, databases, spreadsheets, graphics, pictures, tables, help screens and use of colours, etc. may increase the user friendliness of the environment.

Comment:

Most tutors seem to have some bad experience on computer interface and various ways of improvement are suggested. However, in real life, managers do not make decisions in front of computers and for educational purpose, manual interface is far more superior. Furthermore, although the reports need not to be 100% correct, their formats should be compatible to most decision support tools.

Q7. Should a business game have multi-winning criteria or a specific goal to be achieved ?

A7: Multi-winning policy are believed to be more appropriate in integrated games but participants can have the same specific goal in a functional game. Also, there is not necessarily to have a winner in a game. The strategies and operations that lead to the result are far more important than the result itself. Thus, a team should decide its own objectives before the game and the winner will be the one who has achieved its objectives most effectively. Under this arrangement, a game can be used repeatedly with different objectives. A list of winning criteria is suggested below:

General criteria	Marketing criteria	Financial criteria
- quality	- profitability	- profit
- delivery	- market share	- share price
- innovation	- cash flow	- stock market
- service		- financial growth
- good price for customer		- investment
- good conditions for employees		

Comment:

Most tutors seem to agree on of multi-winning policy. Furthermore, multi-winning policy can also reduce the degree of competition between teams. However, some minimum requirements should be enforced on the teams' chosen objectives so that the participants cannot abuse the system.

Q8. What are the benefits of

(a) syndicate group participation

(b) individual user participation in a business game ?

A8: All tutors believe that group playing is more beneficial than individual user participation and an ideal team size should be around 3 to 4 people. The only advantages of one-player group are 'full participation' and 'less people are required to play the game'. However, the advantages of group playing are lengthy:

- members can discuss and share ideas
- role playing to form a mini organisational structure of a company
- decision process can be shortened
- development of team working skill
- development of the negotiation and communication skills

- amplification of the learning experience
- realism (people work in group in real life)
- brainstorming can be taken place

Comment:

The tutors' views match greatly with the previous findings on group playing with two exceptions:

(1) The tutors always mix up 'non-interactive' and 'singer player' in business games.

(2) They overlook the importance of forming mixed abilities groups

Q9. How much would you pay for a business game ?

A9: Most of the tutors develop their own business games because commercial games do not fit their needs. For a good business game which can fulfil their needs, they are prepared to pay hundreds of pounds while for an excellent one they might pay up to thousands of pounds. Some tutors comment that business games are not given enough priority in their departments and more resource should be put in developing effective business games.

Comment:

Obviously, cost is not a concern in here. The problem is where one can purchase a 'good' game. Without an integrative effort (e.g. help from the library or the department), finding a suitable game for a course is just like finding a pin in a haystack.

Q10. How long should a business game last approximately ?

A10: A business game should last around 15 hours during a five day module, or more generally 1/3 of the length of the module.

Comment:

The length of a business game depends on a lot of factors and standardisation is impossible. However, a game is better to run intermittently instead of continuously during the module. It is because:

(1) It takes time for the administrators to process the game

(2) If anything goes wrong, the administrators can have time to rectify it.

(3) The participants can consider their decisions more thoroughly.

(4) Running the game for 8 hours a day may be quite tiring for some participants.

Q11. What do you think are the advantages and disadvantages of manual and computerised business games ?

A11: Lists of advantages and disadvantages of manual games and computerised games are complied and they are attached in Appendix IX and X respectively. In general, the majority of tutors seem not to trust computerised business games. They claim that sometimes computer games produce wrong results which put the administrators in a awkward position. Overall, the advantages of manual games out number the disadvantages and on the contrary, the advantages of computerised games are far less than the disadvantages.

Comment:

For educational purpose, it is obvious that manual games are better. However, when the level of complexity increases, the use of computer is unavoidable. Fortunately, centralised control computer games seem to provide a compromise. In fact, most computer games do not produce 'wrong' results but the internal logic is fully protected and covered so that there is no way to trace back the decisions from the results.

7.4 Conclusion

Overall, business games are generally admitted as an effective tool in management education if they can reach their full potential. Also, most module tutors seem to be quite knowledgeable about games with only minor confusion. However, they do not know where can they get hold of some 'good' games. On one hand, students enjoy playing gaming, but on the other hand, they are afraid of wasting their time. Their views on games need to be straighten up. In conclusion, these surveys support most previous findings on game design although no stimulating ideas can be obtained.

CHAPTER 8

INTEGRATION OF EDUCATION SIMULATION GAMES

8.0 Introduction

Developing a simulation game is always a time consuming and a complex task especially when the complexity increases with the number of elements incorporated in a game. Naturally, the game authors tend to incorporate more elements in a total management game internal integration than in a functional game. Although hundreds of functional games have been developed covering almost every aspect of engineering business management, they can not be used in the development of a total management game because they are not compatible in either scenario or formats. A lot of developing effort has been duplicated as the same functional elements can be found in different games. Hence, it is important to establish a standard platform so that good functional games can be integrated together to form a good total management game.

8.1 Internal integration

An engineering business company can be broadly divided into three main functional areas: production, marketing and finance. However, finance should not be treated as a separate function in engineering business management because the major function of a finance department is to balance the cash flow. Speculation in the stock or money market to make profit should not be treated as the main business of an engineering business firm which should manufacture products and sell them in the markets to generate revenue. Hence, most of the EBM functional games are addressed on two areas, production and marketing. A typical production game:

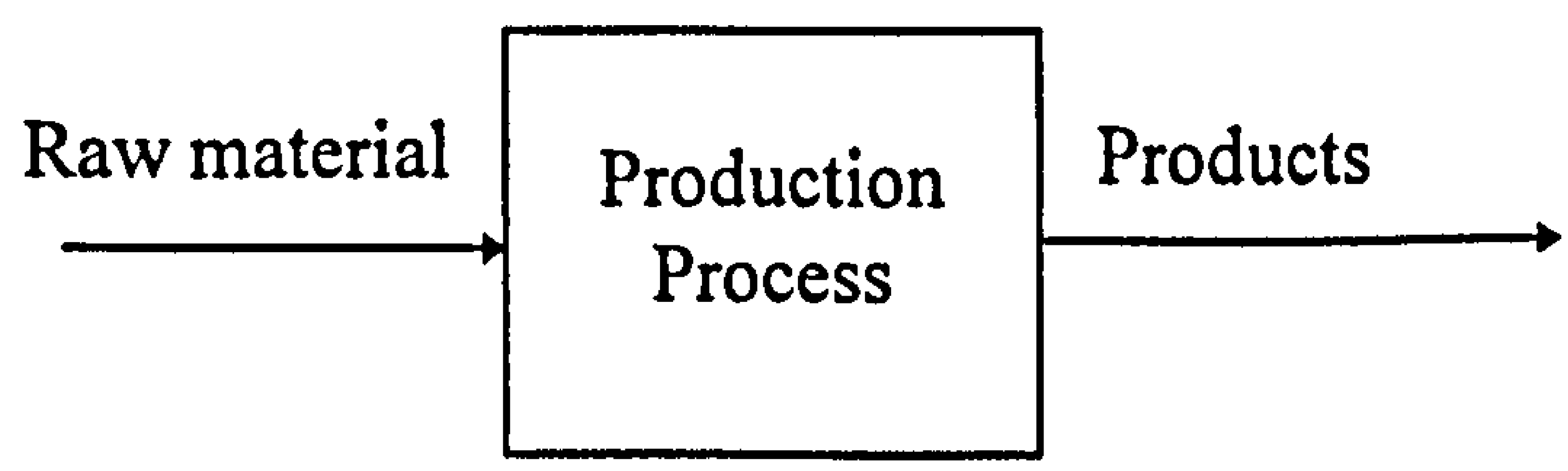


Fig. 8.1 A Production Game

Production games are usually non-interactive and the goal is to deliver the products on time at low cost at the right quality level. Each game period, the participants make decisions on various production parameters in order to meet the demand. The demand of each product is usually pre-set by the game administrator and the arrangement of the production process depends on the objectives of the game. For example, several production lines with machines and labors may be found in a 'production scheduling game'. However, in a 'JIT' game, the machines may be arranged in cells. Reports like machine summary, work schedule and inventory report can be usually found.

On the other hand, a marketing game is usually interactive. Different firms sell the same product at the same market which is usually controlled by a computer program. The number of firms in a game may vary. Depending on the nature of the product, the market will react differently to the market parameters. For example, advertising may increase the sales significantly of a consumer product, however, quality is a more important factor for the selling of an industrial product. Reports like sales report and market research can be found. A typical marketing game is shown below:

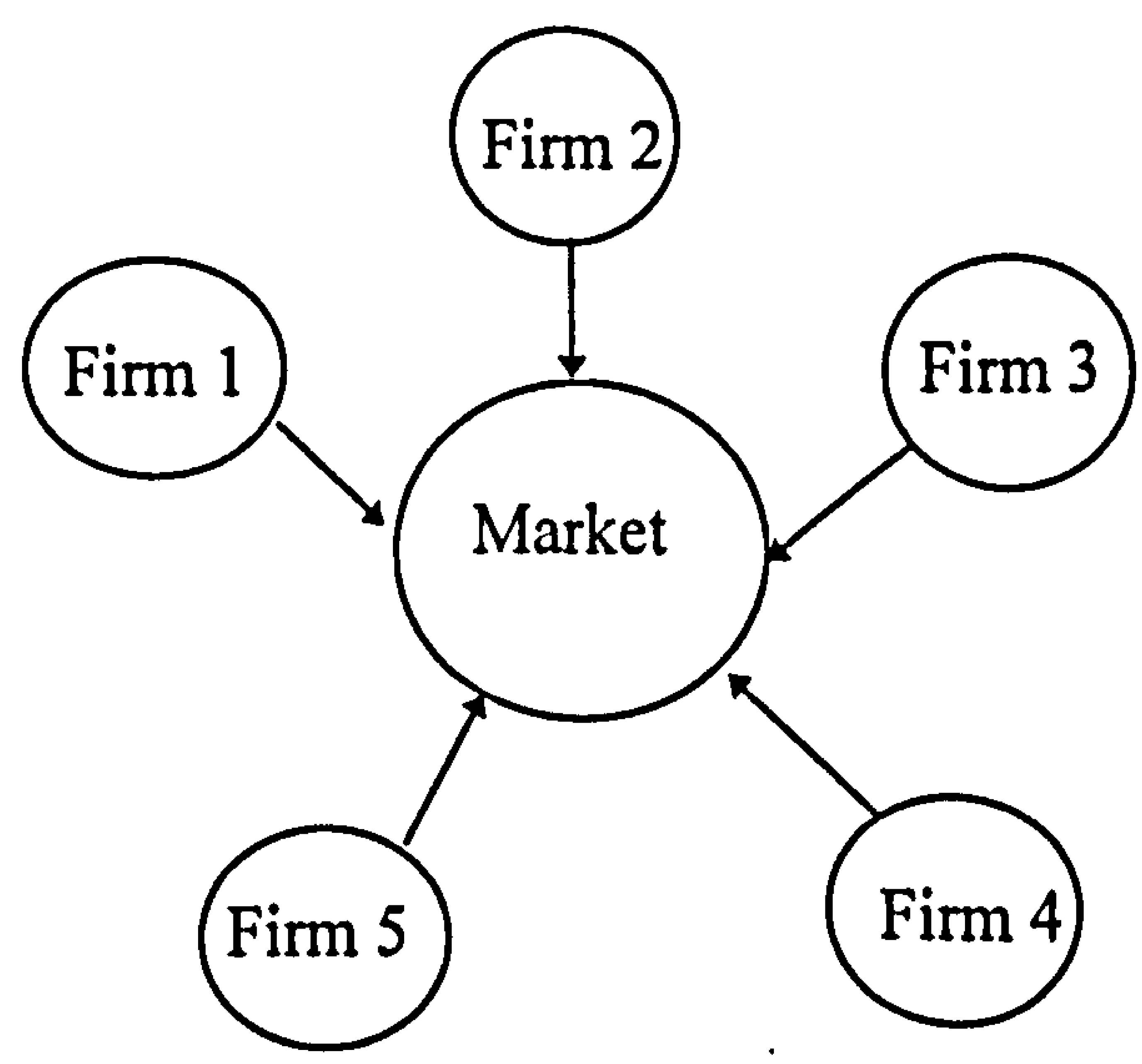


Fig. 8.2. A Marketing Game

The major drawback of a functional game is the lack of integration. In a production game, participants do not have to worry about whether the products can be sold or not. On the other hand, in a marketing game, the participants do not have to consider whether or not the products can be delivered from the production line to the market on time at the right quantity and quality level. Integrating these two functions together allow the participants to view the company as a whole. A typical total management game (internal integration):

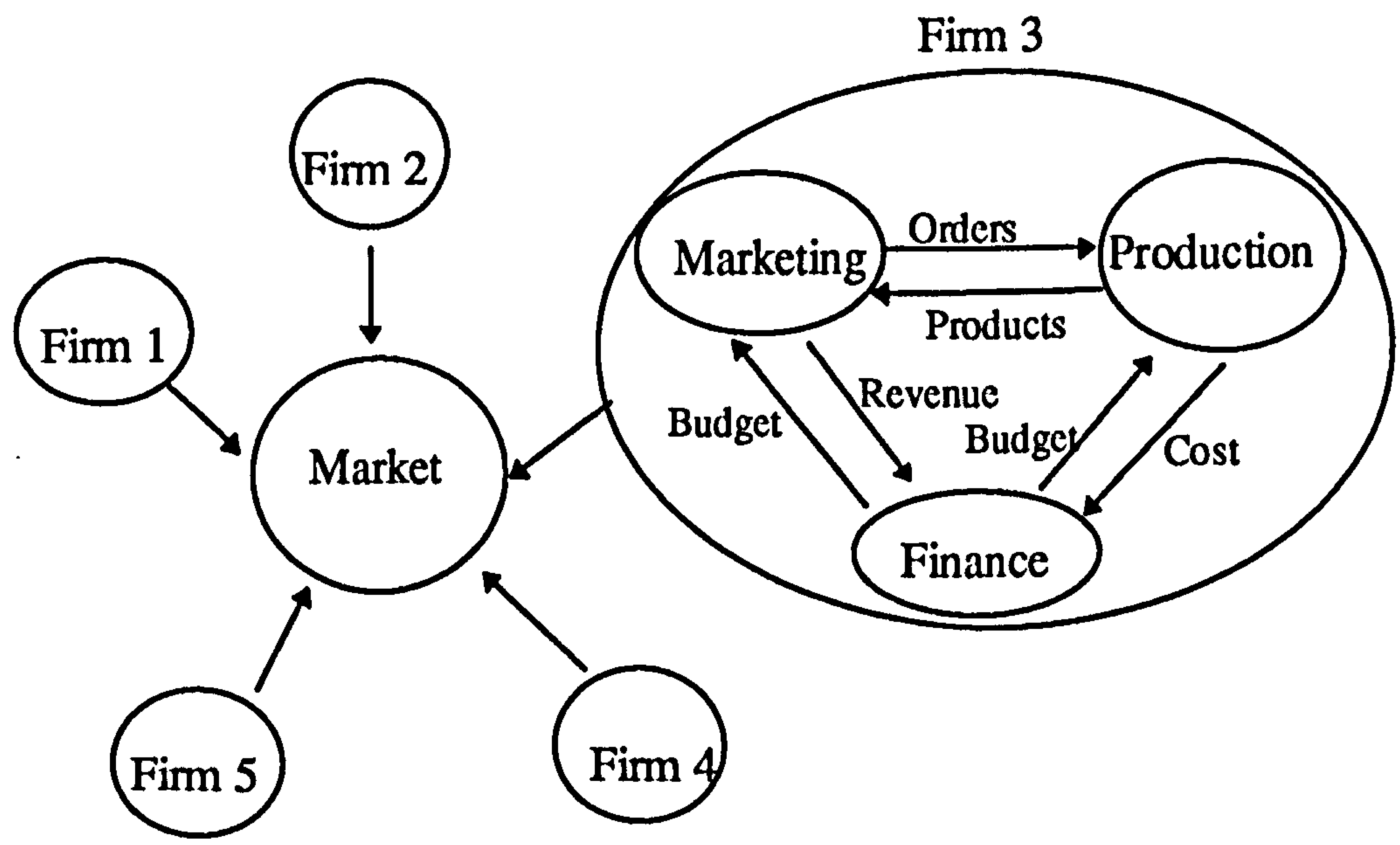


Fig. 8.3 A Total Management Game

Each firm is composed of three areas: marketing, production and finance. Although design and innovation are also important areas in a manufacturing company, they will not be discussed in this paper and they may be integrated into the above model as functional elements in the future. In a typical total management game, all firms will have the same production process, use the same type of machines and hire the same number of labors. However, this is not true in the real world. Although they produce the same products and compete in the same market, firms may possess different production technology and produce different quality level products at different cost. So, it is important to set up a standard interface in which different production games can communicate with the same marketing game so that firms in the marketing game can have different production processes.

8.2 Standard Interface

The standard interface for production and marketing is described in the following diagram:

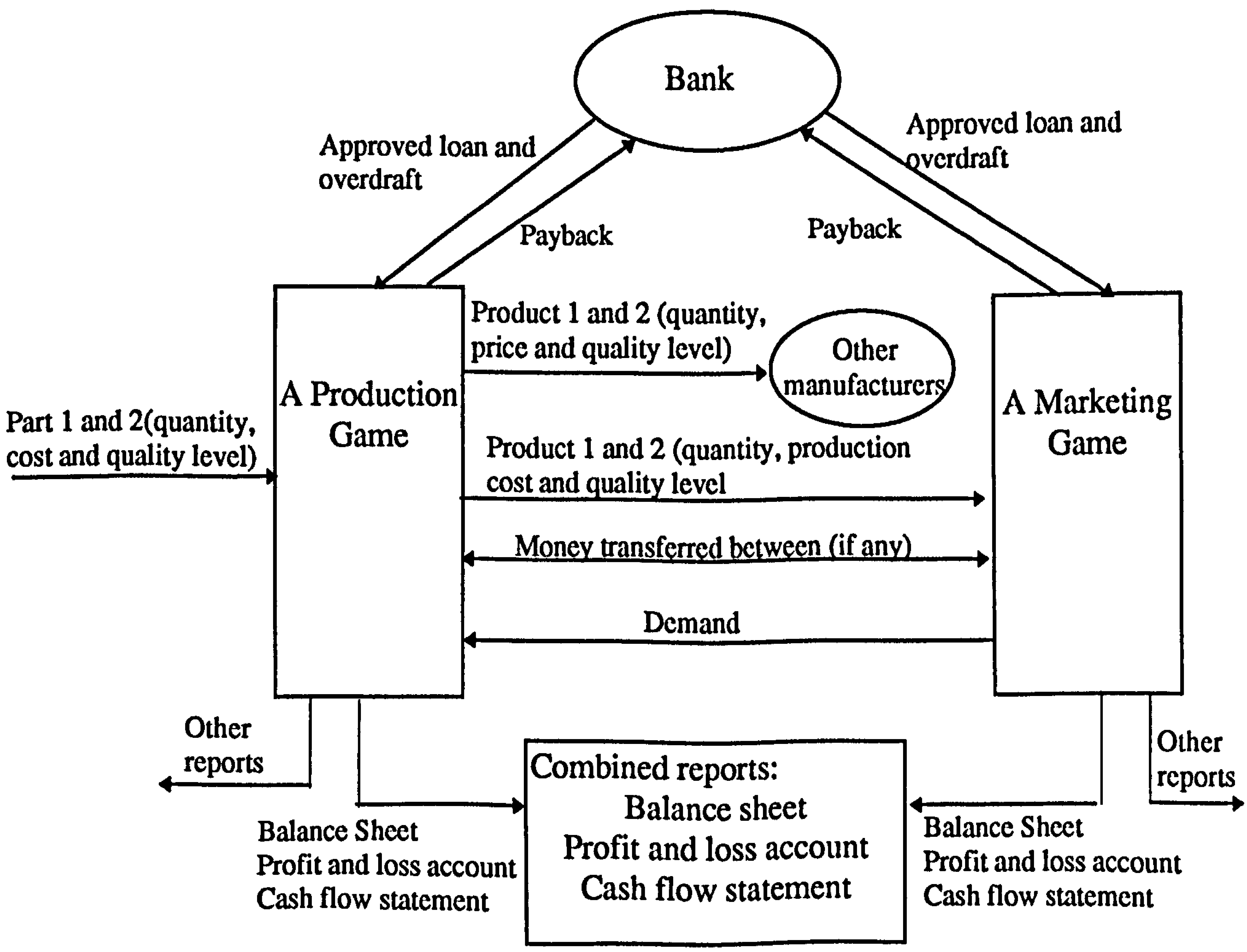


Fig. 8.1 Standard Interface

Three types of information flow between a production and a marketing game: product information (quantity, cost and quality), money and reports. The parameters of the interface are list below (Table 8.1):

Table 8.1 Standard Interface Parameters for Production and Marketing

Production	**Marketing**
Part 1: Quantity / Quality level / Cost/item — recursive	Product 1: Quantity / Quality level / Cost/item — recursive
Part 2: Quantity / Quality level / Cost/item — recursive	Product 2: Quantity / Quality level / Cost/item — recursive
Product 1 Demand Quantity transferred (to marketing) Production cost/unit Product 2 Demand Quantity transferred (to marketing) Production cost/unit	Money transferred in (from production) Money transferred out (to production) Approved loan Payback loan
Direct Selling to manufacturers Product 1 Quantity / Quality level / Price — recursive Product 2 Quantity / Quality level / Price — recursive	
Money transferred in (from marketing) Money transferred out (to marketing) Approved loan Payback loan	

The number of parameters in 'production' are more than in 'marketing' because the interface not only allows a 'production' to communicate with a 'marketing' but also with another 'production'. Also, production involves both raw material and final product in its process while marketing does not change any product information in the marketing process.

A standard interface must be simple so that it can be adopted by different game authors and the existing functional games can be modified to this standard with minimum effort. However, the interface must also be generic enough so that it can accommodate different types of information that flow between. In order to design a simple and generic interface, the communication between 'production' and 'production'; and 'production' and 'marketing' are examined.

8.2.1 Interface Between Production and Marketing

A manufacturing company does not produce something just for the sake of production unless 'marketing' gives a signal that this product can be sold in the market. Production in fact is a market-driven unit and the marketing department is the frontier of a company. The marketing department examines the customers requirement and forecasts the demand of a product and inform the 'production' to produce according to a particular specification which includes quantity, quality level, time to deliver, cost and product features.

Base on the demand forecast and the product specifications, 'production' will plan its resources (raw material, labors and machines) accordingly.

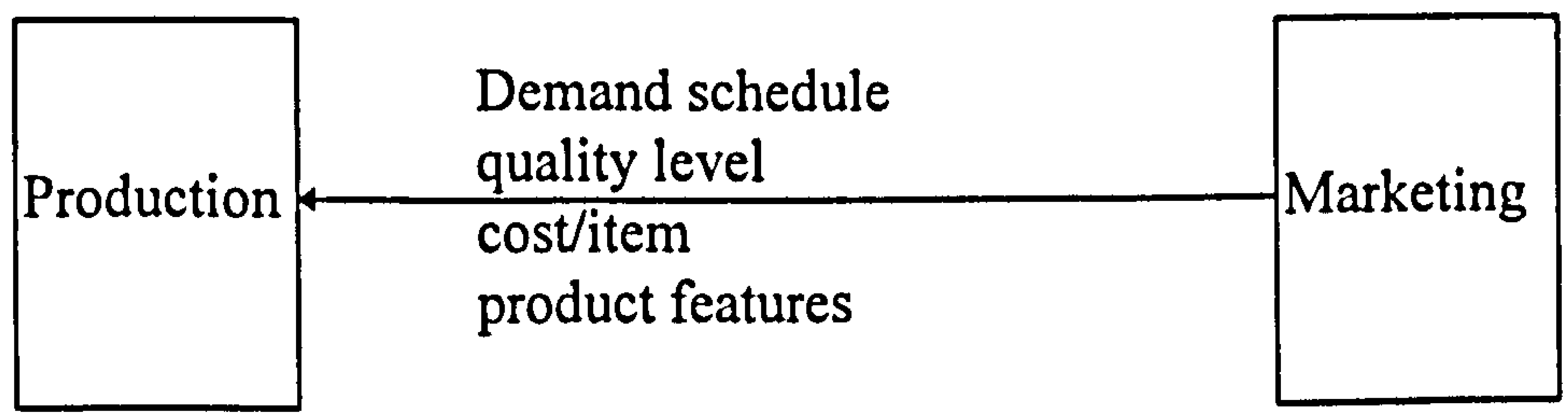


Fig. 8.5 Information flow between production and marketing

These information can be classified into two categories: Direct and Indirect. Direct information refers to the data used directly as an input to the 'production' such as the demand schedule. Indirect information (cost/item, product features and quality level) refers to the data used only as decision support variables. For example, a production manager cannot just inform the shopfloor to produce a product below x dollars/unit because production cost of an item depends on several factors adding up together such as 'cost of raw material', 'production efficiency', 'reject rate' and 'labor cost', etc. Thus, these indirect information will be used for 'production' to make decisions on its production variables but the information itself cannot be used directly as input.

After the 'production' completes the orders, the products will be shipped to 'marketing' for sales. Since the interface allows up to two products selling at a market, two groups of product information (product 1 and product 2) are transferred from production to marketing which includes quantity, quality level and cost/unit. Based on these information, the 'marketing' can determine a appropriate marketing strategy. The input process of these information to 'marketing' is recursive as the products may come from different sources. That is, 'marketing' can be just a stand alone unit which purchases the products from the wholesalers instead of manufacturing the products. In this case, the purchase price becomes the cost/unit.

Certainly, money can be transferred between the two departments if they belong to the same company. Both 'production' and 'marketing' can borrow money from the bank because these two units may be run as stand alone functional game. However, when they are integrated together, only one channel of borrowing should be used and the borrowed funding can be transferred across to each other.

Although both 'production' and 'marketing' will produce their own production and marketing reports, three reports are in common: Profit and Loss Account, Balance Sheet and Cash Flow Statement. A program is written to merge these two sets of reports together so that the participants can analyze the financial reports of a whole company.

8.2.2 Interface Between Production - Production

The interface allows a 'production' unit to supply products to another 'production' unit as raw material say from production A to production B. Production A then becomes a wholesaler or a supplier of raw material. For each transaction, production A transfers the product information (product 1 and product 2) which include quantity, quality level and price to production B which will input the information as raw material (part 1 and part 2). The interface allows recursive inputs and outputs as a supplier may supply its products to several manufacturers and vise versa.

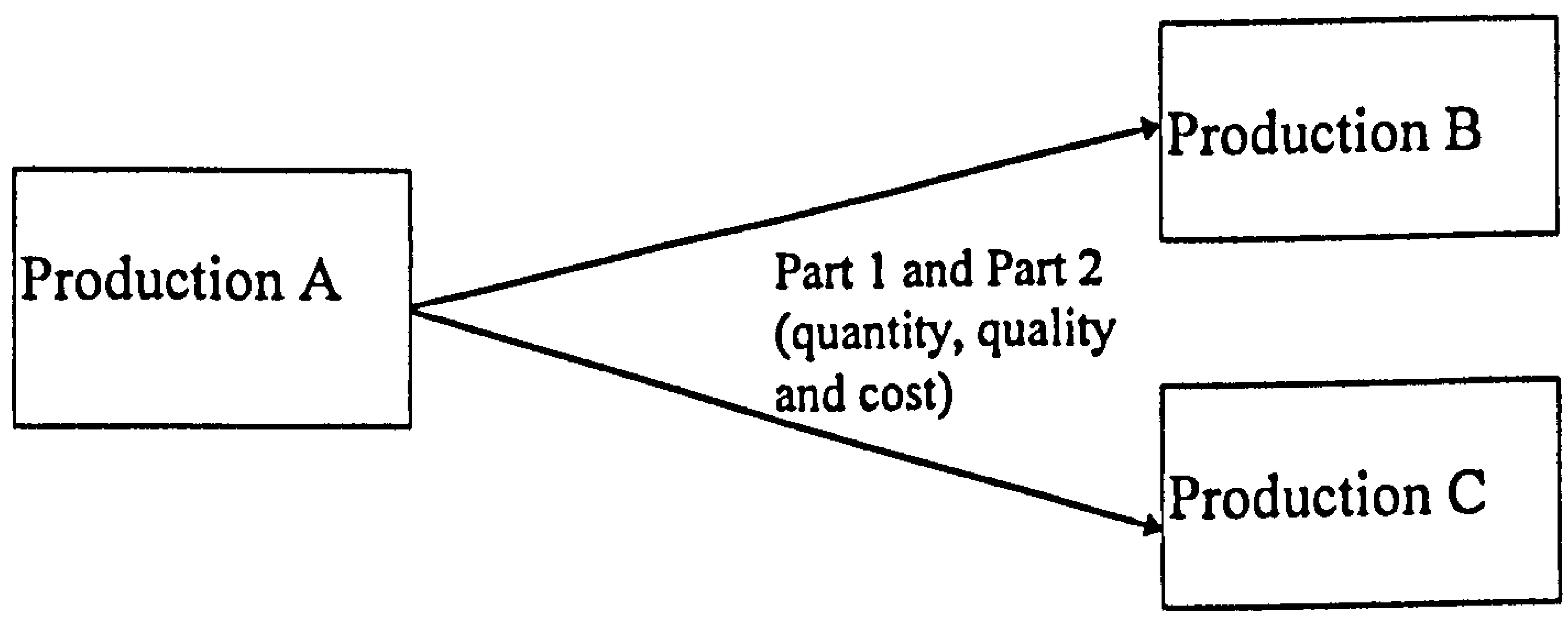


Fig. 8.6 Supply of Raw Material

In short, this standard interface allows the 'production' to produce up to two products from two different parts which can be purchased from different sources. The products can be transferred to the 'marketing' department for selling as end products or they can be sold directly to manufacturers as raw material. Money are freely transferred between these two functions.

In conclusion, the setup of the standard platform allows the game administrator to choose any 'good' production and marketing games to be integrated together. The participants not only have to tackle the elements incorporated in both functional games but also have to manipulate the cash flow between these two functions. This financial element is incorporated in the game automatically when the internal integration takes place. In the game, A bank (the game administrator) will examine all the loan applications and co-ordinate the money transferred between. Combine financial reports including balance sheets, profit and loss account and cash flow statement are generated for the whole company.

With the presence of the standard interface, a marketing game can now be integrated with several production games at the same time so that different firms may produce the same product with different production processes. However, the game administrators have to be particularly careful in setting up the game parameters when more than one production game is used in a total management game. Otherwise, a firm up may end up with production facilities far behind its competitors.

The formation of a total management game from functional games has several major advantages over traditional way of game development:

1) Save time and money

The development time and cost will be significantly reduced as the functional games have already existed. The development time will be largely spent on the selection of appropriate functional games for integration.

2) Correct by construction

Since each functional game has been well developed and debugged, the integrated game will be 'correct by construction'.

3) Various levels of abstractions

The game administrator can introduce the total management with various levels of abstractions and complexities which have been proved to be a important factor in game design.

Level 1: Production game

Level 2: Marketing game

Level 3: Total management game

4) Flexibility

Since more than one production game can be attached to a marketing game, the game administrator has the total freedom to design the structure of the industry. Also, when the game administrator would like to convey different production philosophy to the participants, instead of introducing a new game, the marketing part of the total management game can be remained unchanged.

8.3 External Integration

Although the above internal integration allows the students to view the company as a whole, a company does not exist by itself. There are suppliers as well as competitors. In order to manage an engineering business company, the manger has to know the industry and keep close relationships with the suppliers. The design of the standard interface not only allows the 'production' to talk with the 'marketing' but also with their manufacturers and suppliers. With the existence of the standard interface, structures of external integration can now be implemented. Although there are several key areas in supply chain management, relationship between suppliers and vendors; and the information flow between them probably are the two most factors in supply chain management. With the advance in technology, information flow between warehouses, distributors and manufacturers becomes more efficient and accurate. However, the simulation of an efficient information system may be covered by an individual simulation game but not within the external integration structure itself. The communication between suppliers and manufacturers are done manually in the external integration structure in which face to face negotiation are emphasized. The external integration structure emphasizes on the supplier-manufacturer relationship which is hardly covered by any other game.

Although hundreds of external integration can be formed, four basic structures are presented below:

-Material Supply Chain

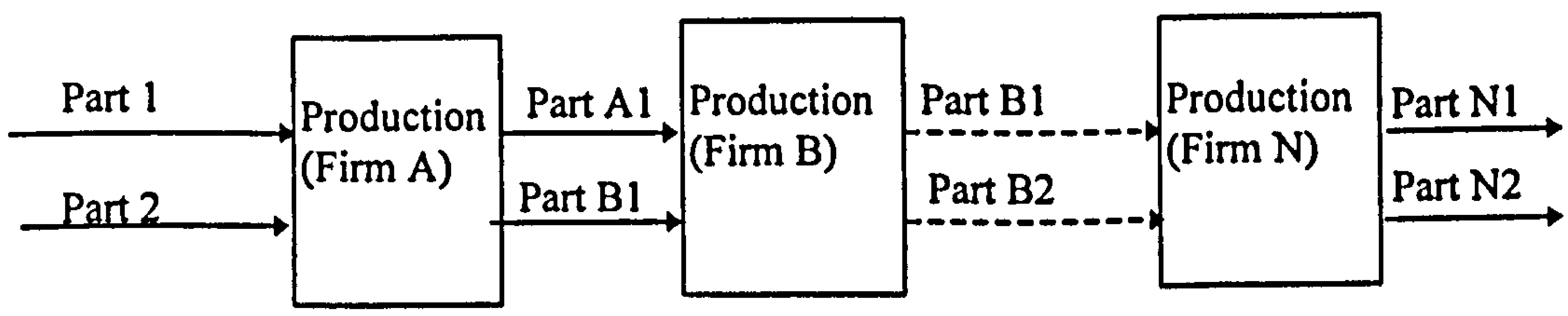


Fig. 8.7 Material Supply Chain

Each production firm is a production game. The chain can compose of any number of firms. They can be the production game under the same or different parameter settings running recursively. Or they can be different production games with different production process involved. The demand of firm N (the last one) will be set by the game administrator. Although participants may work in different firms, they in fact work together for the same goal that firm N can meet the demand on time while individual firm tries to lower their production cost and improve their product quality. Any mistake made by any firm in the chain will affect the production process. Instead of competing with all other teams as in most simulation games, participants can understand that production is a chain process and it is important to work together with your suppliers and manufacturers.

- Material Supply Network

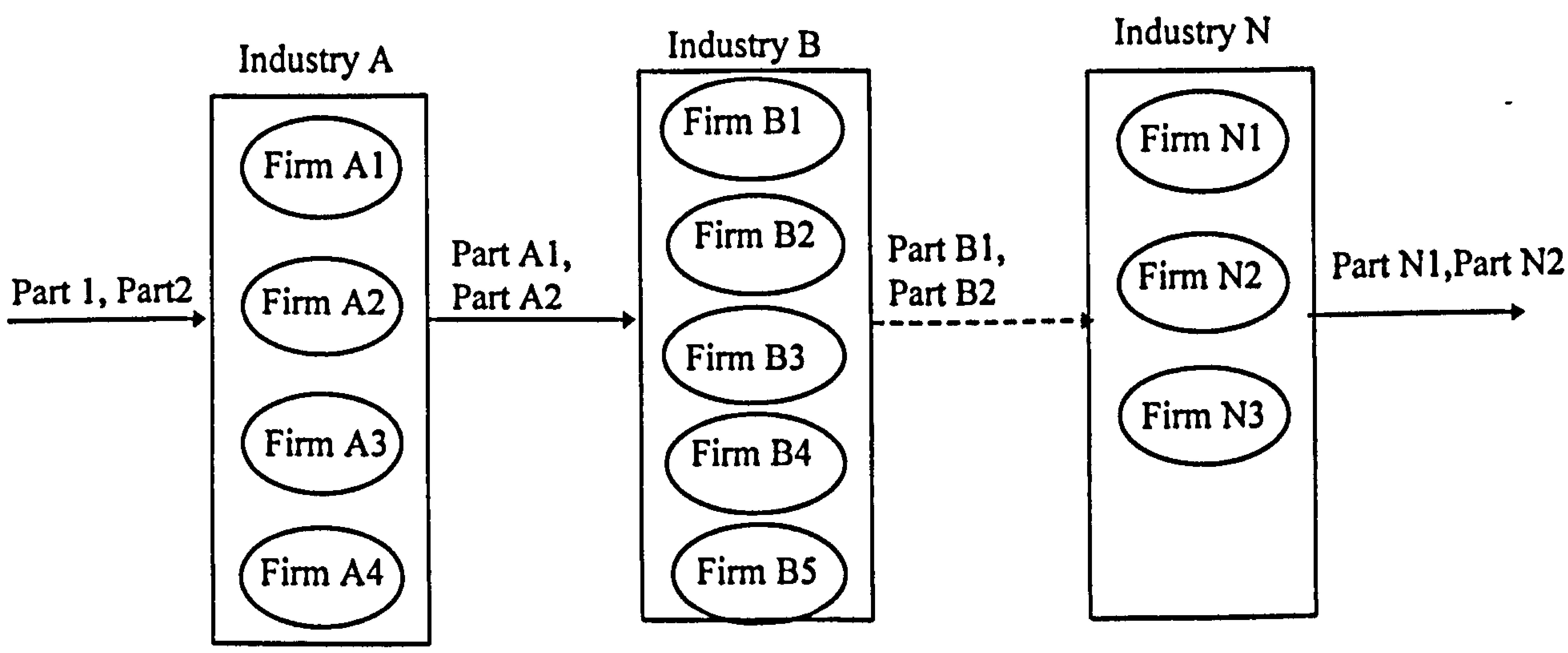


Fig. 8.8 Material Supply Network

The material supply network is similar to the material supply chain with the exception that the manufacturers can now have the freedom to choose their part suppliers. So, industry B is in fact an industrial market of part A1 and part

A2 for industry A. In order to deliver the products on time, a firm has to ensure the continue supply of the raw material. Hence, good relationships have to be maintained with the suppliers and 'human' factors have been emphasized in this structure. Role play element is incorporated in this structure that the manufacturers have to negotiate (face to face) with their suppliers on the terms and conditions of the transaction which includes quantity, quality and price level. Business contracts may be signed up between firms. On the other hand, a firm may have multiple sources on the same part to minimize the risk of shortage of raw material. In short, firms not only have to compete with their competitors but form allied with their suppliers and customers. Also, different marketing strategies can be tested under this structure. Several interesting scenarios may arise under this structure:

1) Firms from the same industry, say Firm B1 and B2 may team up together to purchase all the parts from industry A in a particular period to push Firm B3, B4 and B5 out of the industry.

2) When certain parts are in demand, although contracts may have been signed between firms, some firms may be willing to break the contracts and pay the penalty so that they can sell their parts to other firms at higher price.

3) Since the price are negotiable, bulk purchase discount is expected. A firm may have to balance the 'inventory level' and the bulk purchase discount to get the best economical effect. JIT may not always be the best production philosophy at all time.

4) In an industrial market, customers are more knowledgeable on the products. Firms have to compete on price, quality as well as prompt delivery. If a firm cannot meet the demand on time, the customer can go somewhere else.

5) Supply management always suggests to minimize the number of suppliers on raw material, however, this is just like putting all the eggs in one basket. If anything goes wrong with the suppliers, the manufactures may have to face the risk of stopping production. The participants can test out the ideas in the structure and balance the pros and cons.

-Product Supply Chain with Marketing

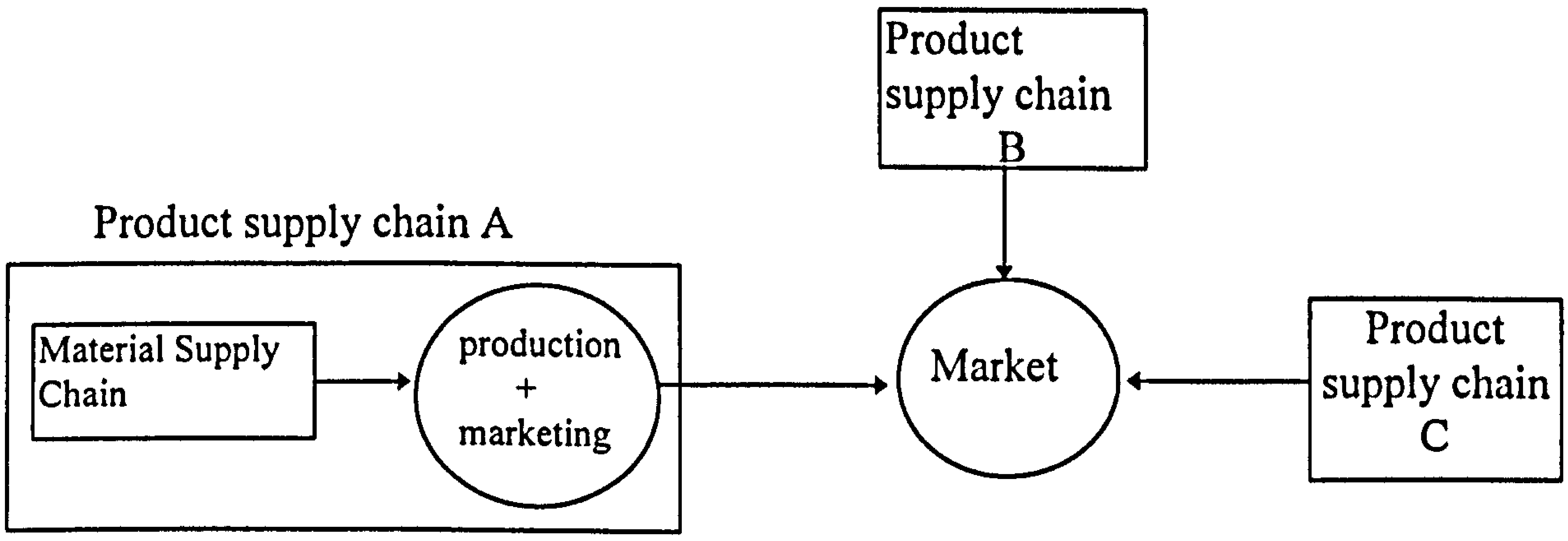


Fig. 8.9 Product supply chain with marketing

This structure integrates 'material supply chain' and the consumer market together. The structure describes the complete cycle of a product - from raw material to the finished product; from manufacturers to customers. The number of 'product supply chains' competing in the market as well as the number of firms in the material supply chain are totally flexible. Similarly, a structure of 'material supply network' + marketing can also be formed. This structure allows the participants to prepare the products from raw material to the end products selling in the market. Firms belonged to the same material supply chain have to work together to deliver the products to the market.

- Material Supply Network with Part Market

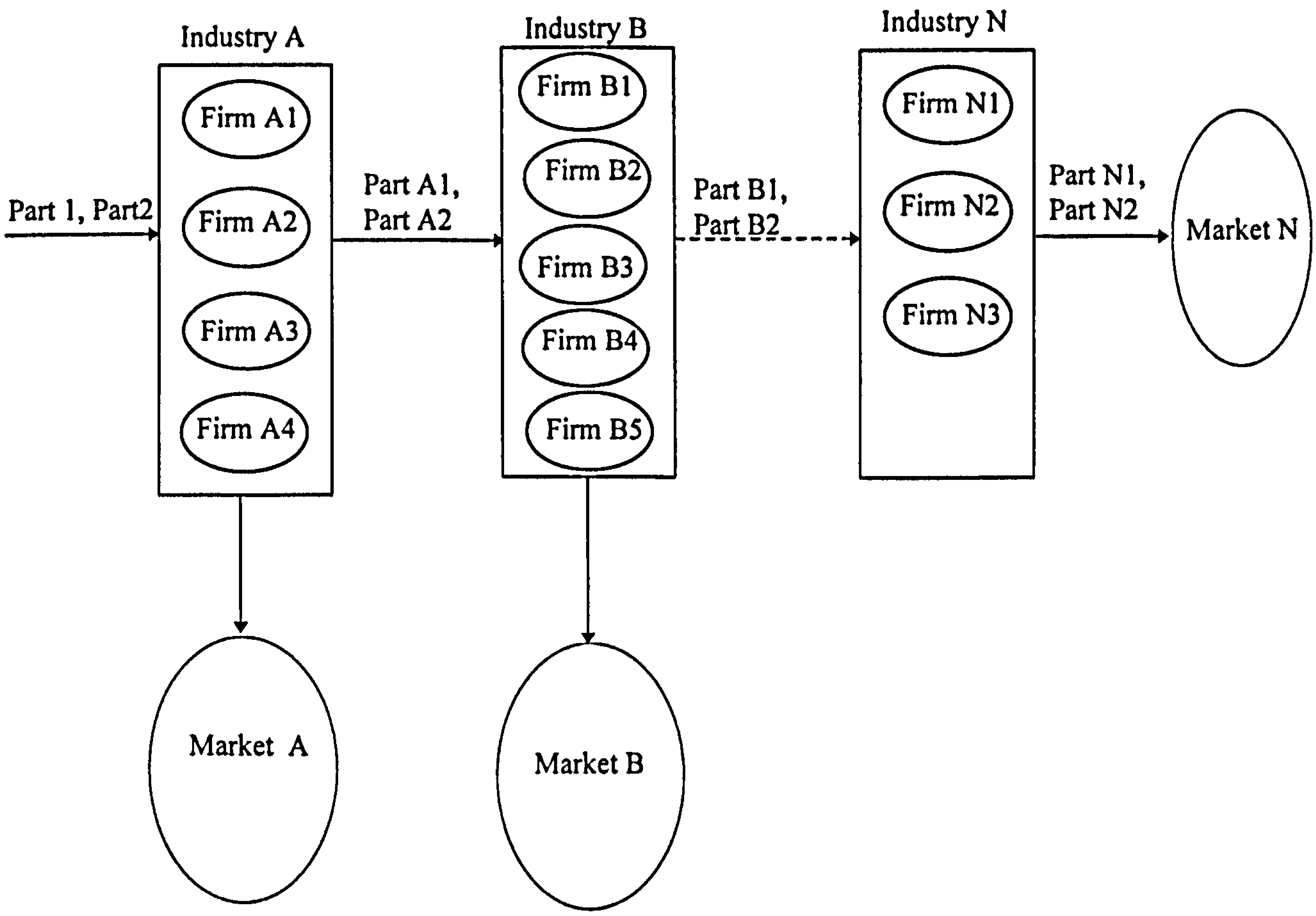


Fig. 8.10 Material supply network with part market

This structure not only allows the firms to sell their parts to the manufacturers but also to the open industrial markets. For example, car tires can be sold to car companies or to small garages or even to the consumers directly.

In formation of the above structures, it is assumed that a firm will input two parts and produce two products. In fact, a firm can produce only one product from two parts or produce two products from the same part. The external structure that can be formed will become even more flexible and the combination is infinite.

There are several advantages of using external integration game for education purpose:

1. The importance of supplier-manufacturer relationship has been highlighted.
2. The participants are able to view the overall structure of an industry.
3. The same game can be played again and again with the participants rotating to play different roles.
4. The design and development time of an external integration management game can be highly reduced.
5. It saves time for both students and game administrators in getting familiar with the game scenarios and formats.
6. The role rotation allows the participants to understand that they will speak different languages when they are in different roles. If business agreements have to be made, both sides have to be co-operative.
7. Firms are not necessary to compete with each others as they may work together to form allied.

8.4 Hierarchical, Modular Structure

The approach described above to building simulation games make use of the concept of modularity endorsed by Zeigler (Zeigler 1987, 1990). As the design of simulation systems becomes more complex and expensive, researchers have to pay more attention to the ability to model as well as to simulate these systems. There are two main research themes in discrete event simulation modeling - How to speed up the execution and how to simplifying the modeling? On speeding up the execution, the introduction of the distributed simulation (or parallel simulation) and Time Warp operating system provide a forum for discussion (Abrams, Lomow 1990; Zeigler, Christensen 1990). Special languages, like Maisis, Sim++ and CPS have been developed for parallel simulation (Bagrodia, Liao 1990; Baezner, Lomo, Unger 1990). Also a number of distributed simulation algorithms broadly classified as 'conservative', 'optimistic' and 'conditional-event' have been constructed.

On simplifying the modeling process, most research work is built on the foundation of 'The Discrete Event System Specification (DEVS) Formalism' which provides a formal means of specifying systems in a modular and hierarchical manner (Zeigler, 1990; Meter, Deisenorth 1991; Zeigler, Lee, Lim 1990). To facilitate manufacturing simulation modeling, object-oriented simulation program generators, like Modsim, Robots and Robot-Sim have developed. These, and the development of GIBSS (Generalised Interaction Based Simulation Specification) and Heterogeneous simulation models enable engineers to design manufacturing subsystems into various levels of abstractions (Meter, Deisenorth 1991). In the following, terminology used in DEVS will be bracketed with < >.

A functional game < atomic model> is a basic unit of a simulation game. A functional game can either be a production game or a marketing game. Since each functional game is a separate program which can be run and debugged individually, one only has to consider the interfaces between functional games during the process of the integration. Analogy is like playing with Lego. These bug-free functional games constitute a library for future usage. This 'correctness by construction' has been applied in the development of systems for the design of Integrated Electronic circuits by many vendors.

Theoretically, a functional game and a marketing game can make up any engineering business system but the time involved in linking up each game together increase greatly with the complexity of the system. Standard units of enhanced functionality are built to facilitate the process of modeling. An analogy is like construction of an electronic circuit. Basically, only three types of logic gates (AND, OR, NOT) are required for any circuit but one can imagine the size of a personal computer if only three types of gates are used. The industrial structures constructed above, 'total management game', 'material supply chain', 'material supply network', 'Product supply chain with marketing' and 'Product supply network with part market' are standard units <coupled models> which can be available form the library <model base> to build a more complex system. This hierarchical approach of building models enriched the library endlessly so that even more complex models may be built quickly and easily.

Zeigler, in his book, 'Object-Oriented Simulation with Hierarchical, Modular Models' defines System Entity Structure as:

"The System Entity Structure (SES) directs the synthesis of models from components in the model base. The SES is a knowledge representation scheme that combines the decomposition taxonomic, and coupling relationships. Associated with an SES is a model base which contain models which may be expressed in any of dynamic formalisms mentioned earlier. the entities of the SES refer to conceptual components of reality for which models may reside in the model base. Also associated with entities are slots for attribute knowledge representation" [Zeigler 1990]

The developed approach is object-oriented. Each type of a functional game (production and marketing) is a 'object'. Copies of the same game are 'instances'. The operations performed in the game are 'methods'. The formation of a external structure is referred as 'The Composition Tree' under

DEVS formalism. The functional games, the total management games and the supply chain networks form a model base. Different possible ways of linking games together describe the system entity structure. The entities of the SES refer to any simulation game which may be reused, that is, they can reside in the model base. Since each functional game is generic as the game scenario can be largely set by the instructor and with the application of the object-oriented approach of constructing hierarchical, modular models, thousands of total management games and external integration structures can be formed for different educational purposes.

8.5 Conclusion

The integration of functional games to form total management game or external integration structure is making use of Zeigler's Concept of Modularity. This basic idea is not something new. However, the idea was never applied in the formation of education simulation games in engineering business management before. The development of the standard interface between production and marketing enables the concept to be applied in a new area. Also, this methodology allows the formation of external integration structure which no other game has ever done it before.

CHAPTER 9

IMPLEMENTATION AND EVALUATION

9.0 Introduction

The research project, 'On the design and integration of education simulation game in engineering business management' covers two main areas: Design and Integration. In the design part, a list of design criteria has been drawn up on the development of a 'good' education simulation game in engineering business management. In the integration part, functional games are integrated together to form total management games and external integration models. Hence, the implementation and the evaluation of this project are also divided into two parts:

(1) Design :Implementation and evaluation

(2) Integration: Implementation and evaluation

9.1 Implementation and Evaluation of the Game Design

A list of criteria for designing a 'good' engineering business games has been drawn up in Chapter 5 and 6. A marketing game and a production game have been developed according to these guidelines to illustrate the suggested approach. However, whether these criteria can lead to the development of a 'good' game is largely subjective.

Game authors have long been divided on the best way to evaluate games and simulations. Some argue that they are so different from traditional means of teaching that they require novel methods of evaluation, while others argue that it is important to apply the traditional methods such as drawing up the of a list of criteria for success before a game is played and then checking off its performance against each item on the list. Attempts to draw up general criteria for evaluation are described in Armstrong and Taylor (1971), Eraut (1972), Elgood (1975), Shirts (1975) and Megarry (1976), while useful illustrations of the various claims and counter claims made for and against games are contained in Boocock and Schild (1968) and in Avedon and Sutton-Smith (1971). More specification evaluations of particular games are contained in Percival (1978), Percival and Ellington (1979), Hornby (1980), MacCullum Stewart, Brand and Walker (1982) and Evans and Sculli (1984).

Even though much effort has been used, there is no conclusion on how a game should be evaluated. There are still a long way form being able to explain exactly how and why a particular game 'works' and perhaps still further from being able to explain why one game is better than another, but this not necessarily a matter for much concern. A person can continue to find it difficult to define an elephant but he can still recognize one easily when it comes along and know when it is useful and when it is not! Perhaps the main cause of our difficulties is our penchant for the application of traditional, quantitative techniques to new areas in which such techniques are inappropriate. A more imaginative use of attitude surveys or qualitative rather than quantitative techniques of evaluation could be much more productive.

Hence, three surveys have been carried out between 92 and 94 and the results have been discussed and examined in Chapter 7 to confirm the findings. The surveys have dual purpose. On one hand, they gather the users' and administrators' opinion to confirm the previous findings on game design. On the other hand, they also contribute in the design process. The design and the evaluation are in fact concurrent procedures. Thus instead of presenting the survey results in this Chapter, they are presented immediately after the findings on the design philosophy to minimize the confusion.

It also has to be emphasized that although a production game and a marketing game are developed according to the design guidelines, the games should be not used for evaluation of the design guidelines otherwise biased results may be obtained.

Firstly, although both games are developed under the developed guidelines, it does not guarantee that both games must be good. Since the guidelines are only used to minimize the risk of designing a bad game and the success of a game depends on numerous factors which are not fully controlled by the game designer.

Secondly, even both games are played, there is no conclusive way to evaluate whether the education objectives of the games have been met. Although the participants may like the game, it only implies that the games are entertaining.

Thirdly, although both games are developed under the guidelines, they are only particular examples in game design. They do not represent the design philosophy.

That is why only general questions are asked in the surveys which are independent with the games played.

9.2 Implementation and Evaluation of Integration

In the second part of the research, a standard platform has been formed so that functional games can be integrated together to form total management games as well as external integration structures. This application of hierarchical, modular modeling into the formation of engineering business management games has never been mentioned by any game author before. The establishment of a standard interface opens a new approach for game development. Several game elements, such as human factors and supply chain logistic, which are hardly covered by other EBM games can now be easily incorporated in the structure. The integrated games are now automatically having various levels of abstractions and complexities. Since the integration of functional games covers an untouched area in engineering business management that no other game has ever covered before, the evaluation is self-explanatory.

The implementation is in fact the evaluation. If the integration of function games on the standard platform can be successfully implemented, the concept is then confirmed. Two functional games, marketing and production have been developed for this purpose. In a micro design level, both games are designed according to the set of design guidelines developed in Chapter 4, 5 and 6. In a macro level, these two games can be integrated to form a total management game and the recursive use of the games are able to form tens of external integration structures.

9.3 The Production Game

Objective: To deliver the product at the right quality level at the right time

Formats:

Location:	Classroom environment
No. of teams:	1 - infinite
Team size:	3
Number of rounds:	4 - 6
Time allowed:	4 - 6 hours
Administrator:	1 - 2 people
Game control:	Computer centralized control with manual interface

Features:

Realism:	Acceptable
Interaction:	Non-interactive
Focus:	Subject area: Product quality
	Use: Educational
Complexity:	10 to 19 decisions
Entertainment:	By nature and no random factor

The developed production game is a 'quality' game which emphasises the idea that product quality is affected by the input (raw material, machines and labors), production process (learning curve, staff turnover, machine breakdown and maintenance, etc.) and output (inspection and average outgoing quality level). The game administrator will be the bank to approve the loan. The objective of the game is well defined and the message is clear. The game can be run with 2 level of complexities and besides the three standard financial reports, only one production summary is printed. There is no magic combination of decision variables and high quality product can be produced if and only if the participants balance the quality control on all three phases (input, process and output). The program details, flowcharts, program listings, users' manuals and a sample output can be found in Appendix XI - XII.

9.4 Marketing Game

The marketing game allows multi-winners. The participants are allowed to define their own objectives which can either be 'profit making' or 'market share maximization' or a combination of both.

Formats:

Location:	Classroom environment
No. of teams:	4 - 6
Team size:	3
Number of rounds:	6 - 8
Time allowed:	6 - hours
Administrator:	2 people
Game control:	computer centralized control with manual interface

Features:

Realism:	acceptable
Interaction:	Interactive
Focus:	Subject area: Product and service quality
	Use: Educational
Complexity:	11 - 14 decisions
Entertainment:	By nature and no random factor

The marketing game emphasizes on the idea that although price and advertising are important factors for selling of a product, quality of service and the product quality are two everlasting factors which the customers are looking for. Since firms may not have the same objective in the game, this can reduce the degree of competition between firms. Again, there is no magic formula and most of the elements follow the law of diminishing return. It has to be emphasized that both marketing and production games should be used in conjunctions with lectures and discussions. The objectives of the games have to be clearly defined and post game assignments are necessary. The program details, flowcharts, program listings, users' manuals and a sample output can be found in Appendix XVIII - XXIV.

9.5 Standard Interface

Zeigler defines modularity as "*the description of a model in such a way that it has recognized input and output ports through which all interaction with the external world is mediated.*" So, in developing the production and simulation games, ports are reserved in both games so that they can communicate with the each other. There are three main ports.

1) Money transferred in and out

This port allows money transferred in and out of a functional game. That is, the production game can transfer the goods to the marketing game at the production cost or the goods can be sold to manufacturers to obtain revenue.

2) Transfer of raw material and end product

The production game inputs raw material and produce end product which may be the raw material for other manufacturers. For marketing, it inputs products and convert them into money. Fortunately, these inputs/ouputs (I/O) are already standard elements in functional games and there is no need to create these particular ports in both games. However, the I/O formats have to be compatible so that they can communicate with each other. There are two standard rules in the I/O formats on the standard platform:

- There are always three attributes in a product: quantity, price, quality level. If the transferred product do not have all the required attribute like the quality attribute, it can be set to the default value (0% defective)

- The input and output are recursive unless a 0 is entered to the quantity attribute. This allows a functional game to communicate with several functional games at the same time.

3) Financial Reports

Each functional game produces its own kind of reports. However, under the standard platform, all functional games have to produce three common reports: Balance Sheet, Profit and Loss Account and Cash Flow Statement. When marketing is integrated with production, their financial reports will be merged together.

9.6 Internal and External Integration

When internal integration takes place, the marketing department can only sell the products which produced by the production department. For accounting purpose, the marketing department will pay the cost of production to production department. Combined financial reports, Balance Sheet, Cash Flow Statement and Profit and Loss Account will be produced. The combined reports and the manuals are shown in Appendix XXV - XXIX.

For external integration, the marketing firm can purchase the products from any production firm and sell them in the market. On the other hand, a production firm can purchase their raw material from the industrial market. Since products are freely transferred between firms, infinite industrial structures can be formed. Sample for material supply chain can be found in Appendix XXX.

It has been demonstrated that a production game can now talk with other production games and marketing games under the standard interface. Internal and external integration structures can be formed with these channels of communication.

9.7 Selection of Software for Implementation

Both games and the interface are developed in C under Disk Operation System (DOS). Disk Operating System is one of the most popular operating systems in the world. Data can be either inputted by files or typed in by users. C, a procedural language is chosen to implement the object-oriented paradigm because C is more popular and portable than any object-oriented language. Hence, many traditional game authors can develop their new functional games on the standard platform without changing their programming styles. Object-oriented concept are utilized in the integration structure level, but within the functional game itself, top-down design is still applicable. Thus , the formation of the internal integration models and external structures follows the object-oriented concept but programming of the functional games can be done by using traditional computer languages like Pascal or Basic.

9.8 DEVS With Distributed Simulation

The developed approach of building hierarchical modular games is built on the foundation of Zeigler's DEVS. The developed approach is further confirmed and elaborated by incorporating distributed simulation in building complex simulation models. Instead of integrating functional games together, machines, resources and even the control algorithm are represented in cells which are integrated to form complex simulation models. Each cell itself is a program and information are transferred between cells through Dynamic Data Exchange under Windows environment which allows multi-tasking. Same cells can be used recursively. A prototype consists of 9 machine cells and a simulation clock has been developed using QuickC for Windows and the result has been published in 1993. The details can be found in Appendix XXXII - XXXIII.

9.9 Conclusion

All the program details, flowcharts and sample outputs can be found in the appendixes of this thesis. The writing up of the programs and testing take more than a year to complete. Although only two functional programs have been built on the standard platform, tens of hierarchical modular models can be formed from these two games. The game authors can now build their functional games on the standard platform and these games will be added in the library from which game authors can choose the 'good' ones to be integrated together for different education purposes.

CHAPTER 10

CONCLUSION

10.0 Introduction

There are two objectives in this project, 'On The Design And The Integration Of Education Simulation Games In Engineering Business Management'. The first objective is to develop a game design philosophy for engineering business management games. The second objective is to form a standard platform so that functional games can be integrated together to form more complex simulation games. Both objectives have been fully met especially the formation of standard platform opens a new area for game development.

10.1 First objective - Game Design Philosophy for Engineering Business Management Games

Game design is always a 'black box' and game design in engineering business management is even a 'black box' within a 'black box'. Although there are numerous factors affect the effectiveness of a simulation game, the design of the game itself must be one of the most important factors. The existing systematic approach of game design does not provide any concrete help to the game authors especially the approach is mainly designed for social science games. For game design in engineering business management, game authors have to learn from their own mistakes or experience. However, education do not welcome mistake as a 'bad' business game may bring adverse effect to the participants. Although game design is always a personal task, it is necessary to have a framework or a set of guideline for game design in engineering business management to minimize the risk of developing a 'bad' game. In Chapter 4,5 and 6, a list of design criteria has been drawn up and has been evaluated in Chapter 7 in which the survey results have been examined.

While Chapter 5 forms a framework (formats and features) for engineering business management games, A list of common weaknesses is discussed in Chapter 6. Although the list may not be exhaustive, it in fact has already covered most of the drawbacks commonly found in engineering business management games. Together with the systematic approach discussed in Chapter 4, the game design approach of engineering business management games has been established and thus the first objective is met.

10.2 Second Objective - Development of a Standard Platform

While the developed game design approach can help the game authors to develop good games, the development of a standard platform allows these good games to be integrated together to form more complex simulation models. As the manufacturing industry changes from production-oriented to customer oriented, the management approach evolves from functional optimization to total integration which includes ‘functional optimization’, ‘internal integration’ and ‘external integration’. Functional games are always good at covering optimization of functional areas. However, there are several problems with the existing internal integration management games as mentioned in chapter 5. Also, the development time and money increase exponentially with the complexity of a game. Naturally, more elements will be incorporated in an internal integration management game than in a functional game. Hence, it becomes even more difficult to develop a game to cover external integration as several firms may be involved in the game. In fact, there is hardly any existed game which can simulate external integration of the industry. A new way of constructing complex management games is required. By modifying Zeigler’s Discrete Event System Specification Formalism, a complex simulation game can now be formed by integrating functional games together. The standard interface between functional games has been designed and developed using C language under DOS environment. Two functional games, a production game and a marketing game have also been developed to illustrate the formation of hierarchical modular simulation games. The concept and the methodology are further confirmed and elaborated by incorporating distributed simulation by using Dynamic Data Exchange to transfer data between modules under

Windows environment which allows multitasking to be taken place. A prototype consists of several machine cells has been successfully developed using QuickC for Windows and the results has been published in 1993. The second objective is obviously met.

10.3 Summary of Findings and Contribution to Knowledge

- The weaknesses of using simulation games in education have been identified and rectified.
- The proposed game design philosophy is the first game design philosophy ever suggested in Engineering Business Management area.
- The establishment of a standard platform allows different functional games to be integrated together and this is the first standard platform that has ever been built for simulation games. The existence of the standard platform opens a new era in game design.
- Various levels of abstractions and complexities are automatically incorporated into the simulation games formed on this standard platform.
- The internal integration structure allows the integration of existing functional games into a total management game. Functional games become re-usable in different total management games. This 'correct by construction' concept has never been mentioned in any game literature before.

- The external integration structure simulates the manufacturing industry and their interaction which can hardly be found in any other existing game. The game authors now have the total freedom to formulate the structure of the industry according to their own education objectives.
- Human factors and supply chain management are incorporated in the external structure and these elements are hardly covered by any other game.
- DEVS has never been used in game design and this modular approach of designing simulation games opens a new research area for game designers.

Chapter 6 and Chapter 8 are original work and they are the core of this paper. In short, the 'black box' in game design has been opened and the application of hierarchical modeling in game design has covered areas that no other game has ever covered before.

10.4 Limitations

- There is no graphical support in the user-interface in the standard platform which are programmed in C under DOS environment. Also, data are transferred between functional elements by files or user-input.
- On the standard platform, only the generic protocols among 'production and production' and 'production and marketing' have been considered, protocols

between other important functions like design and innovation have not been implemented.

- In order to make a simulation game realistic, real-life data may need to be linked up to the game. At the moment, the standard platform is not able to communicate with a database or a spreadsheet.

10.5 Difficulties Encountered

The research was started in 1989 and a total of seven yeas have been spent on the project. Since business games are not commonly used in teaching engineering business subjects, there are difficulties to locate appropriate reference materials. Also, as game design is always a personal task, it is important to keep the view be objective in developing the design criteria. Language problem is probably one of the major obstacles when the thesis is written. Since the project is more on philosophy than technology, some findings and conclusions are argumentative that good presentation technique is required.

10.6 Experience and Knowledge Acquired

Besides the knowledge directly gained from this research (gain design, hierarchical modeling, DEVS, use of business games in education, etc.), several things have been learnt from this research project. Since the project involves several areas integrating together, logical and systematic thinking is learnt from organizing and integrating various information together. In addition, as some project areas are controversial, analytical and intellectual arguments have to be presented. Also, time management and presentation skill are equally important in completing this project. After all, a research project is a process for training of mind.

10.7 Further Development and Recommendation

There are three areas that can be further investigated and developed based on the findings of this project:

- As different games have different education objectives, they have different requirement on formats, features, elements, ways of running and roles of the administrators. An expert system can be developed to assist the game authors to built a complete framework of a game based on the education objectives. The designed philosophy proposed in this research can be further developed as expert system's rules so that decision tree can be formed. This project should form a good proposal for a Master degree research.

- Research can also be performed on further breaking down a functional game into elements which can be integrated together by hierarchical modeling. Instead of viewing a functional game as an entity, machines, tools, raw materials and labours become the atomic models. Each of this entity is a program by itself. A production operation will be composed of tens of these entities interacting with each other under Windows environment which allows multi-tasking to be taken place. Data can also be transferred through Dynamic Data Exchange (DDE). This concept integrates DEVS with distributed simulation in design of simulation models. A primary study has already been carried out and shown in Appendix XXXII and XXXIII. This should form a good base of a Ph.D. project which may last up to 3 years.

- Work can also be done to incorporate other functional elements such as innovation and design in the standard interface. Also, another set of protocol should be designed to communicate with a database system which provides the real-life data for a simulation game. The introduction of graphical interface will make the standard platform more user-friendly and all future development work should be carried out under Windows environment. The work will last for one to two years depending on the depth of detail and this should form a good MPhil research project.

REFERENCES

Abrams, M., Lomow, G. , 1990, 'Issues in Languages for Parallel Simulation', Distributed Simulation, Proceedings of the SCS Multiconference on Distributed Simulation, San Diego, edited by David Nicol, Society for Computer Simulation.

Ackoff, R.L. 1967, 'Management misinformation systems', Management Science, 14, B147-B156

Allen, T. J., Marquis, D. G., 1964, 'Positive and negative biasing sets: The effects of prior experience on research performance', IEEE Transactions on Engineering Management, EM-11, 158-161.

Andlinger, G.R. 1958, 'Business Game-Play one', Harvard Business Review 36, March -April, pp. 115 - 125

Aplin, J.C. and Cosier, R.A.1979, 'The use of intensive simulation in executive development and academic settings', In S.C. Certo and Brenenstuhl (Eds.), Insights into experiential pedagogy, Tempe : Arizona State University Press

Ausubel, D. P. 1968, 'Educational psychology: A cognitive view', New York:, Rinehart, and Winston.

Baezner, D., Lomow, G., Unger, B.W., 1990, 'Sim++: the Transition to Distributed Simulation', Distributed Simulation, Proceedings of the SCS Multiconference on Distributed Simulation, San Diego, edited by David Nicol, Society for Computer Simulation.

Bagrodia, R.L., Liao, W.T., 1990, 'Maisie: A Language and Optimization Environment for Distributed Simulation', Distributed Simulation, Proceedings of the SCS Multiconference on Distributed Simulation, San Diego, edited by David Nicol, Society for Computer Simulation.

Berelson, B., Steiner, G. A., 1964, 'Human behaviour: An inventory of scientific findings', New York: Harcourt, Brace and World.

Biel, W.C. 1965, 'Training programs and Devices', in R.M. Gagne Ed, Psychological Principles in System development, New York, Holt Rhinehart and Winston

Birmingham Post ,May 1, 1986, 'University takes a byte at success'

Bobillier, P.A., Kahan, B.C., Probst, A.R. 1976, 'Simulation with GPSS and GPSS V', Prentice-Hall, P. 6

Boud, D., Pascoe, J. 1978, 'Experiential Learning: Developments in Australian Post-Secondary Education. Sydney: Australian Consortium on Experiential Education.

Boydell, T. 1976, 'Experiential Learning', Manchester: Department of Adult and Higher Education, University of Manchester

Bruner, J.S. 1966, 'Toward a theory of instruction', New York: Norton

Butler, P., Strang, D. 1979, 'An Extension of Wolfe's study of simulation game complexity', Decision Sciences, 10, 480-486

Butler, Pray, T. and Strang, D.,1979, 'An Extension of Wolfe's study of simulation game complexity', Decision Sciences, 10, 480 - 486

Carson, J.R. 1969, 'Teaching Decision Making', Business Games Handbook, American Management Association, pp. 35-45

Butterworth, J. 1989, 'The Right Stuff', London, Purnell.

Carson, J.R. 1969, 'Teaching Decision-Making', Business Games Handbook, American Management Association.

Cheung S.W. 1991, 'Development of a Computer Interactive Graphical Simulation in Manufacturing Education', Proceedings of The Fifth International Manufacturing Conference in China, Guangdong University Press.

Cheung S.W., Roy R., Bhattacharyya S.K. 1992, ' Development of a Hierarchical, Modular Manufacturing Simulation Model', Proceedings of The Second International Conference on Automation Technology, Edited by Kuo-Shong Wang, Center for Automation Technology Yuan-Ze Institute of Technology Press.

Cheung S.W., Wong H.M., Chan C.H. 1994, 'Object-oriented, Hierarchical, Modular Simulation Models with DEVS and Distributed Simulation.', Proceedings of the 9th International Conference on Computer-Aided Production Engineering, Nanjing, China.

Chorba, R.W., New, J.L. 1980, 'Information support for decision maker learning in a competitive environment: an experimental study', Decision Sciences, Oct., Vol 11:4, pp 603-615

Cohen, K.J. and Rhenman, E., 1961, 'The Role of Management Games in Education and Research', Management Science, 7,131-166

Cohen, K.J., 1960, 'The Carneigie Tech Management Game', The Journal of Business, Oct, No. 4, Vol. 33.

Cohen, K.J., Dill, W.R., Kuehn, A.A. & inters, P.R. 1964, 'The Carnegie Tech Managment Game', Richard D. Irwin, Homewood, Ill.

Cohen, K.J., Rhenman, E. 1961,'The Role of Management Games in Education and Research', Management Science 7, July, pp. 131-166

Coleman, J.S. 1967, 'Academic games and learning: Design and Implementation', Columbus, Ohio: Merrill

Couger, J. Daniel, 1977,' 20 Most Popular Games', Computing Newsletter, Nov. p.1

Craig, Robert L. (Ed.) 1976, 'Training and Development Handbook, 2nd Ed.;, (Sponsored by the American Society for Training and Development), McGraw Hill

Dale, E. 1952, 'Planning and developing the company organisation structure' (Research Report No.20), New York: American Management Association.

Davis, L. 1989, Datamation, Mar. 15, Vol. 35, pp. 65-72

Dearden, J. 1972, 'MIS is a mirage', Harvard Business Review, 50(Jan/Feb), pp.90-99

Dill, W.R., Dopplet, D.N. 1963, 'The acquisition of Experience in a Complex Management Game', Management Science 10, No. 1

Distribution - August 1988, pp. 63 -68

Distribution - September 1990, pp. 33 - 40

Dobles, W.R., Zimmerman, F.R. 1966, 'Management Training Using Business Games', Training and Development Journal 20, No. 6.

Duke, R.D. 1979,'Nine steps to game design', article presented in 1979 conference of the International Simulation and Gaming Association (ISAGA), on the theme, 'How to build a simulation game'

Easton, G. (1992), 'Learning from Case Studies', Prentice Hall

Elgood, C. 1993, 'Handbook of Management Games', 5th Ed. , Gower

Elgood, Chris 1988, 'Handbook of Management Games', Gower, 4th Ed.

Elgood, Chris 1993,'Handbook of Management Games', 5th Ed., Gower

Ellington, H., Addinal, E., Percival, F. 1984, 'Case studies in Game Design', Kogan Page

Ellington, Henry, Addinall, Eric and Percival, Fred 1982' A Handbook of Game Design', Kogan Page Ltd.

Ely, D.P., Gerlch, V.S. 1980, 'Teaching & media, a systematic approach', Prentice- Hall

Evans, J.R. 1993, 'Applied production and operations management', 4th Ed., West Publishing Company

Faria, A.J. 1989, 'Business Gaming: currage Usage Levels', Journal of Management Development, Vol. 8, Pt. 2, pp.58-65

Faria, A.J. 1987, 'A Survey of the Use of Business Games in Academic and Business', Simulation and games, Vol 18, No. 2 , June , p. 207 - 224

Fenton, E. 1967, 'The New Social Studies', New York: Holt, Rinehart and Winston, Inc.

Filley, A. C. 1970, 'Committee management: Guidelines from social science research', California Management Review, 13, 13-21.

Fripp, J. 1984, 'Games can be Educational too', Journal of European Industrial Training, vol. 15, pp,27

Fripp, John 1984, 'Games for Research', Journal of European Industrial Training, Vol 8:3, P. 17 - 22

Gibbs, G.1987, 'Learning by Doing', London: Further Education Unit.

Ginter, P.M. and Rucks, A.C. 1983, 'War Games and Business Strategy formulation', Managerial Planning, Sep-Oct, Vol 32:2, p. 15 - 19, 34

Gooding, C., Zimmer, T.W. 1980, 'The use of specific industry gaming in the selection, orientation and training of managers', Human Resource Management, Spring

Graham, R.G., Gary, C.F. 1969, 'Business Games Handbook', American Management Association, New York.

Greenblat, C.S. 1988, 'Designing Games and Simulations: An Illustrated Handbook', SAGE

Greenblat, C.S., Duke, D. 1981, 'Principles and practices of Gaming/Simulation', Sage Publications, Beverly Hills, California.

Greenblat, C.S. 1980, 'Group Dynamics and Game Design : Some Reflections', Simulation and Games, 11 (March) : 35 - 58

Greenlaw, Paul S., Herron, Lowell W. and Rawdon, Richard H. 1962, 'Business Simulation in Industrial and University Education', Prentice Hall

Guetzkow, H. and Jensen, L.1966, 'Research Activities in Simulated international Processes', International Studies Quarterly, Vol 9, No. 4,p. 261 - 274

Hamachek, D. 1990, 'Psychology in Teaching, Learning, and Growth', 4th Ed., Allyn and Bacon

Hartley, D.A., Johnson, P.V., Fitzsimons, A., Lovell, J., Chippendale, B. and Clayton, J.K.S. 1979, 'A Case Study on the Development of the Home Defence Training Game, HOT SEAT', Operational Research society, Oct, Vol 30:10, P. 861 - 871

Henry, J., 'Meaning and Practice in Experiential Learning', In. Weil, S.W. & McGill, I. (Ed.) 1989, Making Sense of Experiential Learning, The Society for Research into Higher Education & Open University Press.

Hinton, Roy W. and Smith, Daniel C.1985, 'STRATPLAN', Prentice-Hall

Hogarth, R. M. and Makridakis, S. 1981, 'The Value of Decision Making in a Complex Environment : An Experimental Approach', Management Science, Vol 27, No. 1, p. 93 - 107

Holloman, C. R., Hendrick, H. W. 1971, 'Problem solving in different sized groups', Personnel Psychology, 24, 489-500.

Horn, R.E., Cleaves, A. 1980, 'The Guide to Simulation/Games for education and training', SAGE

Horn, Robert E. and Cleaves, Anne 1980, 'The Guide to Simulation/Games for education and training', SAGE, 4th Ed.

Ip W.H., Cheung S.W 1993., 'Intelligent State Space Modelling with Hierarchical modular simulation', Proceedings of The Sixth International Manufacturing Conference with China', Hong Kong Polytechnic University.

Jackson, J.R. 1959, 'Learning Experience in Business Decision Games', California Management Review 1, No. 2

Jewell, B.R. 1993, 'An integrated approach to Business Studies', 2nd Ed., Pitman Publishing.

Jones, Ken 1985 'Designing Your Own Simulations', Methuen

Joyce, B., Weil, M., Showers, B. 1992, 'Models of Teaching', 4th Edition, Allyn and Bacon

Kelley, H. H., Thibaut, J.W. 1969, 'Group problem solving'. In G. Lindsey & E. Aronson(Eds.), The handbook of social psychology. Reading, Mass.: Addison-Wesley.

Kibbee, J.M. 1961, 'Model Building For Management Games', Simulation and Gaming: A Symposium, Management Report, No. 55, New York: American Management Association, pp. 8-15

Kidd, J.B. 1975, 'Task Complexity and the Performance of Decision Makers', White, D.J. and Bowen, K.C. (Eds.), The role and Effectiveness of Decision Theories in Practice, Hodder and Stoughton.

Kilty, J. 1982, Experiential Learning', Guildford: Human Potential Research Unit, University of Surrey.

Kitchen, D. 1970, 'Games in Industrial and Management Training', in R. Armstrong & J. Taylor Ed., Instructional Simulation Systems in Higher Education, Cambridge Institue of Education.

Kolb, D., Fry, R. 1975, 'Towards an applied theorey of experiential learning', In Cooper, c.L. (Ed.) Theories of Group Press, pp. 33 - 56, Chichester: John Wiley

Kossack, C.F. 1961, Simulation and Gaming: A Symposium, Management Report, No. 55, New York: American Management Association, pp 49

Krajewski, Lee J. 1990, 'Operations Management', Addision-Wesley Publishing Company.

Lancaster, G.A. 1988, 'Essentials of marketing', McGraw-Hill Int'l Ltd.

Larreche, J.C. 1987, 'On simulation in Business Education and Research', Journal of Business Research, Dec. vol. 15, Pt. 6, pp.559-571

Lau, W.S., Frankland,S. 1993, 'Cooperative Education: The Hong Kong Experience', ASEE Annual Conference Proceedings.

Lerviks, A-e, Paltschik, M. 1982, 'On factors associated with success in a complicated business game -- some emperical findings', Meddelanden fran Svenska handelshogskolan, 83, pp. 1-15

Logistics Today, Jan/Feb 1993, 'Playing the Supply Chain Game'

Maital, S., Morgan, K. 1988, 'Playing at Management', Across the Board, Across The Board, April

Mayer, R.E. 1985, Mathematical ability. In R.J. Sternberg (ed.), 'Human abilities: An information-processing approach', New York:Freeman.

McKenney, J.L. 1967, 'Simulation Gaming for Management Development', Boson: Graduate School of Business Administration, Harvard University, 1967.

Mckenney, J.L. and Dill, W.R. 1966, 'Influences on learning in simulation games', American Behaviour Scientist, 10, 28 - 32

McKenney, J.L.1963, 'An evaluation of a decision simulation as a learning environment', Management Technology, 3, No. 3, 56 - 67

McRaith, J.f. & Goeldner, C.R. 1962, 'A Survey of Marketing Games', Journal of Marketing 26, July, pp. 69-72

Megarry, J. (Ed.) 1979, 'Perspectives on Academic Gaming & Simulation 4, Human factors in games and Simulations', The proceeding of the 1978 Conference of SAGSET, Kogan Page

Meier, Robert C. , Newell, William T. and Pazer, Harold L. 1969, 'Simulation in Business and Economics', Prentice Hall

Meter, E.C.D. Deisenorth, M.P., June 1991, 'GIBSS: A model specification framework for multi-stage, manufacturing system design', Simulation, pp 413-421

Miller, R.B. 1954, 'Psychological Considerations in the Design of Training Equipment', WADC-TR-54-563 Ad-71 202 Wright air Development Centre, Wright Patterson AFB.OH

Mintzberg, H. 1973, 'The nature of Managerial Work', Harper and Row

Mintzberg, H., Raisinghani, D. and Theoret A. 1976, 'The Structure of Unstructured Decision Processes', Administrative Science Quarterly, Vol 21, No. 2, p. 246 - 275

Moore, F.L. 1967, 'Business Games versus Cases as Tools of Learning', Training and Development Jouranl 21, No. 10

New, C.C. 1991, 'UK manufacturing: the challenge of transformation', in 'Integration and Management of Technology for manufacturing' edited by E.H. Robinson, H M Ryan and D.Wilcock, Peter Pereginus Ltd.

Partridge, S.E., Sculli, D. 1979, 'Cases versus Gaming', Management Education and Development 10.

Peter, P. 1984, 'Games that Teach you to Manage', Fortune, Vol 110:9, pp 51-54.

Pray, Thomas F., 1987, 'Management Training: It's all in the game'. Personnel Administrator, Vol. 32, Iss. 10.

Pritsker, A.A.B., Pegden, C.D. 1979, 'Intoduction to Simulation and SLAM', Systems Publishing Corporation.

Pritsker, A.A.B.1979, 'Compilation of definitions of simulation', Simulation, Vol. 33, No. 21, August, pp. 61 -63

Raia, A.P. 1966, 'A Study of the Educational Value of Management Games', Journal of Business, Vol. 39, No. 3, July, pp.339-352

Rediffusion Simulation Ltd. 1986, 'Simulation Training, A Guide for Trainers and Managers', A report by Rediffusion Simulation Ltd. for the training technology section of MSC, published by Manpower Services Commission, Sheffield.

Ricciardi, F.M., et al.1957, 'Top Management simulation: The A.M.A. Approach', Elizaberth Marting, Ed., American Management Association.

Richards, T. 1985, 'Stimulating Innovation: A system approach', Frances Pinter, London.

Richardson, H.L., Trunick, P., Dec 1991, 'Position for growth in the 1990s', T&D, pp. 20 -25.

Robinson, J.N.1985, 'Using Games to Present Economics to Managers: A Survey of the Literature', Management Education and Development, vol 16:1, pp17 - 30

Rowland, K.M. and Gardner, D.M. 1973,'The Uses of Business Gaming in Education and Laboratory Research', decision Sciences, Vol 4, p 268 - 283

SAGSET (Society for the Advancement of Gaming and Simulation in Education and Training), Resource lists on the following topics :

(1) Introductory Reading
(2) Game and Simulation Design
(3) Game and Simulation Evaluation
(4) Chemistry
(5) Economics
(6) International Relations
(7) Business and Management Relations
(8) Mathematics
(9) Teaching English as a Foreign Language
(10) Health Education
(11) Education Management
(12) Human Relations
(13) Geography
(14) Music

Sampson, N.W. and Sotiriou, C.E. 1980, 'Business Policy simulation and the intense course structure', In S.C. Certo and D.C. Brenenstuhl (Eds.), Insights into experiential pedagogy, Tempe : Arizona State University Press,

Sayre, Farrand, Map Manoeuvres and Tactical Rides,1961, Springfield Printing and Binding Company, Springfield, Massachusetts, 1908, as quoted by Kalman J. Cohen and Eric Rhenman, 'The Role of Management Games in Education and Research', Management Science, 7,131-166

Schruben, L.W., Margolin, B.M., 1978, 'Random Number Assignment in Statistically Designed Simulation and Distributing Sampling Experiments', Journal of the American Statistical Association, Vol. 73, No. 363, September, p.504

Shannon, R.E. 1975, 'Systems Simulation: The art and the Science', Prentice Hall, p.2

Shaw, M. E., 1932, 'A comparison of individuals and small groups in the rational solutions of complex problems', American Journal of Psychology, 44, 491-504.

Shim, J.K. 1978, 'Management Game Simulation: Surveu and New Direction', University of Michigan Business Review 30, May, pp. 26-29

Sherwood, R. & Verest, 1991, 'Technology Options for Multimedia in Distance Learning', A Report for the Commission of European Communities: Task Force Human Resources, Education Training & Youth, Commission of the The European Communities.

Shirts, R. Garry, 1985, 'Ten Mistakes commonly made by persons designing education simulation and games' SAGSET journal 5,4, pp.147-150

Shirts, R.G.1975, 'Ten mistakes commonly made by persons designing educational Simulations & Games', Sagset Journal, Oct, Vol 5, No. 4.

Shlomo, M. Kim, M. 1988, 'Playing at Management', Across the Road, April

Shubik, M. 1975, 'The Uses and Methods of Gaming', Elsevier

Shubik, M. 1975, 'Game for society, Business and War: Towad a theory of gaming', Elsevier.

Slavin, R.E. 1991, 'Educational Psychology, Theory Into Practice', 3rd. Ed., Prentice-Hall

Sloman, M. 1977, 'How operational Gaming can Help with the Introduction of a New technology', Operational Research Quarterly, Vol 28, No. 4, , p 781 - 793

Somogyi, Sandor and Kisimre, Istvan, 1983, 'Operational Gaming of Cattle Breeding' in 'Operational Gaming, An International Approach' edited by Ingolf Stahl, Pergamon Press

Stewart, R. 1961, 'A Survey of Business Games', Simulation and Gaming: A Symposium, Management Report No. 55, New York: American Management Association, pp. 16 -26

Stewart, R. 1976, 'Contrasts in Management', Mcgraw-Hill, New York, 1976

Tait, R. 1970, 'Business games are more than fun', International Management', Vol. 25, No. 10, pp 35 - 38.

'The Scotsman Management Game', The Training Officer, Dec. 1980, p. 333 - 337

Thomas, Clayton J. and Deemer, Walter L., Jr. 1957, 'The Role of Operational Gaming in Operations Research', Operations Research, Vol 5, No. 1 (February), p.6

Thomas, E. J., Fink, C. F. 1963, 'Effects of group size', Psychological Bulletin, 60, 371-384.

Thorelli, H.B., Graves, R.L. & Howells, L.T. 1962, 'The International Operations Simulation at the University of Chicago', Journal of Business 35, July.

Truemper, K., Dean, B.V. 1974, 'The OPRAD Research and Development Game', Management Science, Vol. 20, No. 6, pp.999-1009

Truemper, K., Dean, B.V. 1974, 'The OPRAD Research and Development Game', Management Science, Vol. 20, No. 6, pp.999-1009

Unwin, D. (Ed.) 1969, 'Media and Methods, Instructional technology in higher education', McGraw Hill

Vance, S.C., Gray, C.F. 1967, 'Use of Performance evaluation model for research in business gaming', Academy of Management Journal, 10, 26-37

Walker, C.H. 1978, 'The effective of progressively complex accounting simulations in increasing decision-making performance', In D.C. Brenenstuhl and S.C. Certo (Eds.), Exploring experiential learning : Simulations and experiential exercises, Tempe : Arizona State University Press

Watson, Hugh J. and Blackstone, John H. 1989, 'Computer Simulation', John Wiley & Sons, 2nd Edition

Weil, S.W. & McGill, I. (Ed.) 1989, Making Sense of Experiential Learning, The Society for Research into Higher Education & Open University Press.

Wolfe, J. 1975, 'A Comparative Evaluation of the Experiential Approach as a Business Policy Learning Environment', Academy of Management Journal, Vol 18 No. 3, pp. 442 - 452

Wolfe, J. and Chacko, T.I. 1983, 'Education Team-size effects on business games performance and decision-making behaviours', Decision Sciences , Jan, Vol 14: 1, P. 121 - 133

Wolfe, J. and Chacko, T.I. 1983, 'Education Team-size effects on business games performance and decision-making behaviours', Decision Sciences , Jan, Vol 14: 1, P. 121 - 133

Wolfe, J. Guth, G. R., 'A Case Approach versus Gaming in the Teaching of Business Policy: An Experimental Evaluation', The Journal of Business.

Wolfe, J.A. 1975, 'A comparative evaluation of the experiential approach as a business learning environment', Academy of Management Journal, 18, 442 - 452

Wolfe, J.A. 1978, 'Correlations between academic achievement, aptitude and business game performance', In D.C. Brenenstuhl and S.C. Certo (Eds.), Exploring experiential learning : Simulations and experiential exercises, Tempe : Arizona State University Press

Wolfe, J.A. 1978, 'The effects of game complexity on the acquisition of business policy knowledge', Decision Sciences, 9, pp.143 - 155

Wolfe, J.A. and Chacko, T.I. 1980, 'Cognitive structure of business game players : Relationship between an individual's cognitive processing equipment and business game performance and play', Simulation and games, 11, 461 - 476

Wolfe, T. 1984,'The Right Stuff', Bantam books.

Zeigler, B. P. Nov. 1987, 'Hierarchical modular discrete modeling in an object-oriented environment', Simulation, pp. 219 -230

Zeigler, B. P. 1990, 'Object-Oriented simulation with Hierarchical, Modular Models, Academic Press Inc.

Zeigler, B.P. 1976, 'Theory of Modelling and simulation', John Wiley & Sons Limited, P.3

Zeigler, B.P., Christensen, E.R., 'Distributed Discrete Event Simulation: Combining DEVS and Time Warp', Distributed Simulation, Proceedings of the SCS Multiconference on Distributed Simulation, San Diego, edited by David Nicol, Society for Computer Simulation.

Zeigler, B.P., Lee, C., Kim, T.G., 'A Hierarchical Methodology for Multilevel AI Computer Architectures', Distributed Simulation, Proceedings of the SCS Multiconference on Distributed Simulation, San Diego, edited by David Nicol, Society for Computer Simulation.

PAGE NUMBERING AS ORIGINAL

Expectations on Business Games

The objectives of the survey is to find out:

1) The effectiveness of business games in education
2) Your preference in formats and features of business games

Your co-operation in completing this questionnaire are greatly appreciated.

---

1) What course are you in?

☐ EBM ☐ ITFM ☐ MSE

2) Have you ever played a business game?

☐ Yes ☐No (Please go to question 4)

3) Please indicate the category of business games which you have played?

Marketing Strategy	☐ Yes	☐ No
Production Planning	☐ Yes	☐ No
Financial Control	☐ Yes	☐ No
Project Planning	☐ Yes	☐ No

Others (Please specify) ____________________

4) Which type of business games do you prefer?

☐ Manual ☐ Computer ☐ Others (please specify) ________

Please state why? ____________________

5) Please rank the top 5 elements in business games in order of preference (1 to 5).

______ Educational	______ Entertaining
______ Interactive	______ Non-interactive
______ Realistic	______ Fictional
______ Team Working	______ Independent (Singer player)

6) How long do you expect to play a business game?

☐ 0-2 hours ☐ 2-5 hours ☐ >5 hours

7) Do you prefer to play business games alone or in a team?

☐ Alone ☐ Team ☐ Either

8) Please rank the effectiveness of following tools in different stages of teaching:
(1 for least effective and 5 for most effective)

	Introduction	Revision	Integration
Lectures	______	______	______
Seminars	______	______	______
Videos	______	______	______
Case Studies	______	______	______
Computerised Games	______	______	______
Manual Games	______	______	______

9) Do you think business games are as effective as other teaching tools in education?

☐ Yes ☐ No

10) Would you like to have business games in your course?

☐ Yes ☐ No

11) Additional Comments

____

____

____

Thank You !!!

QUESTIONNAIRE ON INTEGRATED BUSINESS GAME (Before Game)

The objectives of this survey is to find out:

1) The effectiveness of integrated business games in education
2) Your preference in formats and features in integrated business games

Please fill in this questionnaire **before** the game is played. Your op-operation in completing this questionnaire are greatly appreciated.

---

1. Which course are you taking ?

☐ EBM ☐ MSE ☐ ITFM

2. Have you ever played a business game before?

☐ Yes ☐ No (go to question 4)

3. Please indicate the types of business games you have played.

Marketing strategy	☐ Yes	☐ No
Financial control	☐ Yes	☐ No
Project planning	☐ Yes	☐ No
Human resource	☐ Yes	☐ No
Quality management	☐ Yes	☐ No
R & D	☐ Yes	☐ No

Others (please specify)______________________________

4. What kinds of business games do you prefer ?

☐ Manual ☐ Computerized ☐ Others (please specify)______________

Please state why:______________________________________

5. Please indicate the level of usefulness of business games used as a teaching tool:

	Useless				Very useful
Marketing strategy					
Financial control					
Production & material planning					
Project planning					
Human resource					
R & D					
Quality management					

Other areas:____

6. Please indicate the level of effectiveness of the following tools in different stages of teaching management knowledge.(1 for least effective and 5 for most effective)

	Introduction	Revision	Integration
Lectures			
Seminars			
Videos			
Case Studies			
Tutorial			
On-job training			
Manual Games			
Computer Games			

7. What are the advantages of using business games as a teaching tool?

	Yes	No	Not Sure
Realistic			
Interactive			
Entertaining			
Team Working			
Convenient			
Competitive			

Other advantages______________________________

8. How long would you expect to play a business game ?

☐ < two hours ☐ 2-4 hours ☐ 4-6 hours ☐ > 6 hours

9. How many players should be in a team in playing business games?

☐Alone ☐ 2 players ☐ 3 players ☐ 4 or more players

10. How many functional areas should be integrated in a game?

☐ Single management function
☐ 2 management functions
☐ 3 or more management functions

THANK YOU VERY MUCH!!

QUESTIONNAIRE ON INTEGRATED BUSINESS GAME (After Game)

Please fill in this questionnaire after the game is played. Your op-operation in completing this questionnaire are greatly appreciated.

---

1. Please indicate the level of usefulness of business games used as a teaching tool:

	Useless				Very useful
Marketing strategy					
Financial control					
Production & material planning					
Project planning					
Human resource					
R & D					
Quality management					

Other areas:______________________________________

2. What are the advantages of using business games as a teaching tool?

	Yes	No	Not Sure
Realistic			
Interactive			
Entertaining			
Team Working			
Convenient			
Competitive			

Other advantages______________________________

3. How long would you expect to play a business game ?

☐ < two hours ☐ 2-4 hours ☐ 4-6 hours ☐ > 6 hours

4. How many players should be in a team in playing business games?

☐Alone ☐ 2 players ☐ 3 players ☐ 4 or more players

5. How many functional areas you think should be integrated in a game?

- ☐ Single management function
- ☐ 2 management functions
- ☐ 3 or more management functions

6. Have you enjoyed the game that you have just played?

☐ Yes ☐ No

Why? ____
____

7. Do you think the game is suitable for you?

☐ Yes ☐ No

Why? ____
____

8. Did you learn any management knowledge or skill from playing the game?

☐ Yes ☐ No

What you have learned? ______________________________
____
____

9. Would you like to have integrated business games in your course?

☐ Yes ☐ No

Why? ____
____
____

10. Additional Comments

____
____
____

THANK YOU VERY MUCH!!!

Results of Survey 2:

Duration	< 2 hrs	2-4 hrs	4-6 hrs	> 6 hrs
Before Game	26 (72.2%)	9 (25%)	0 (0%)	1 (2.8%)
After Game	28 (77.8%)	5 (13.9%)	1 (2.8%)	2 (5.5%)

Length of a business game

No. of Players/team	1 player	2 players	3 players	4 or more
Before Game	8 (22%)	11 (31%)	7 (19%)	10 (28%)
After Game	5 (14%)	15 (42%)	9 (25%)	7 (19%)

Team size

	Introduction					Revision					Integration				
Medium / Rating	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Lectures	1	3	5	11	1 6	1	2	11	13	9	1	2	12	15	7
Videos	3	6	8	16	3	0	8	11	11	6	3	7	10	12	4
Tutorials	7	5	9	9	6	0	4	13	15	4	0	1	12	17	7
Seminars	4	8	10	11	3	0	3	9	18	6	2	2	12	12	8
Case Studies	8	6	6	10	6	0	0	8	17	11	0	0	1	20	15
On-job training	7	2	13	7	7	0	0	8	14	14	0	0	5	14	17
Manual Game	6	6	13	7	4	4	6	12	12	2	2	4	10	18	2
Computerised Game	6	4	10	12	4	2	5	11	13	5	0	2	10	18	6

Effectiveness of different teaching media. (5 points for most effective and 1 point for least effective)

Advantages of using business games as a teaching tool:

Realism

	Yes	No	Not sure
Before Game	15 (42%)	7 (19%)	14 (39%)
After Game	16 (44%)	13 (36%)	7 (20%)

Entertaining

	Yes	No	Not Sure
Before Game	31 (86%)	1 (3%)	4 (11%)
After Game	32 (89%)	3 (8%)	1 (3%)

Convenient

	Yes	No	Not sure
Before Game	29 (81%)	4 (11%)	3 (8%)
After Game	28 (78%)	4 (11%)	4 (11%)

Interaction

	Yes	No	Not Sure
Before Game	30 (83%)	4 (11%)	2 (6%)
After Game	30 (83%)	5 (14%)	1 (3%)

Team working

	Yes	No	Not sure
Before Game	21 (59%)	4 (11%)	11 (30%)
After Game	25 (69%)	9 (25%)	2 (6%)

Competition

	Yes	No	Not Sure
Before Game	17 (47%)	6 (17%)	13 (36%)
After Game	20 (56%)	10 (28%)	6 (16%)

The effectiveness of business games on different subject areas (1 point for least effective and 5 points for most effective).

Marketing Strategy

Rating	1	2	3	4	5
Before Game	0	1	11	13	11
After Game	1	6	9	11	9

Financial Control

Rating	1	2	3	4	5
Before Game	1	2	8	16	10
After Game	1	7	7	14	7

Production and Material Planning

Rating	1	2	3	4	5
Before Game	0	2	5	11	18
After Game	1	5	10	11	9

Project Planning

Rating	1	2	3	4	5
Before Game	0	4	7	15	10
After Game	1	10	6	13	6

Human Resource

Rating	1	2	3	4	5
Before Game	2	8	10	13	3
After Game	4	17	7	5	3

Research and Development

Rating	1	2	3	4	5
Before Game	3	9	13	7	4
After Game	7	14	8	4	3

Quality Management

Rating	1	2	3	4	5
Before Game	1	3	14	13	5
After Game	5	11	12	6	2

Average ratings of the effectiveness of games in different subject areas:

Subject Areas	Average Rating		
	Before Game	After Game	Difference
Marketing Strategy	3.94	3.58	0.36
Financial Control	3.97	3.53	0.44
Production and Material Planning	4.25	3.61	0.64
Project Planning	3.86	3.36	0.50
Human Resource	3.19	2.61	0.58
Research and Development	2.81	2.5	0.31
Quality Management	3.8	2.69	1.11

It is interesting to find that all seven subject areas received lower ratings after games. Statistically speaking, assuming that the paired difference constitute a random sample from a normal distribution, a 90% confidence interval for the mean difference is given by (0.04, 1.08). This means that the mean reduction on ratings is between 0.04 and 1.08 with significant level of 0.1.

	Functional game	Integrated game with 2 mgt fns.	Integrated game with 3 or more fns.
Before Game	3	6	27
After Game	0	9	27

Preference on levels of integration

	Yes	No	Not Sure
No. of Students	26 (72%)	9 (25%)	1 (3%)

Introduction of Business games in the MSc courses.

Title: Development of the specification for an Integrated Business Game

Q1. Which is the best educational method or combination of methods for management students to understand and practice ' real ' management of a company ?

Q2.(a) In the institution where you teach are there any business games available for teaching purposes ?

(b) If yes, have you ever used them as part of your teaching programme?

(c) Why do / don't you use them ? What are the results obtained ?

(d) What are the advantages and disadvantages of games?

Q3.(a) What functional expertise should a general management manager have ?

(b) What a management student should learn before they go out to work?

(c) Which of these functional areas are appropriate to be incorporated in a

Q4. What do you think is more important for a business game ? Simplicity or credibility ? Why ?

Q5. What is the role of the game administrator ?

Q6. What features do you think would make a business game more friendly to the user ? How should the interface look like ?

Q7. Should a business game have multi-winning criteria or a specific goal to be achieved ?

Q8. What are the benefits of

(a) syndicate group participation

(b) individual user participation in a business game ?

Q9. How much would you pay for a business game ?

Q10. How long should a business game last approximately ?

Q11. What do you think are the advantages and disadvantages of manual and computerised business games ?

Survey 3: Advantages and Disadvantages of Business games in education

Advantages	Disadvantages
able to see how a company works as a whole	over simplified compared to the real world
risk-free experiment	time consuming to develop a game
team working / role playing	players have different reactions towards games and real business world.
simulation of real life situations	it is difficult to develop a good game
dynamic way of learning	takes time for students to understand the game
different options to try	when the game is too complicated, students get disillusioned.
much interaction	if the business game is not right, students learn the wrong thing.
give people experience of making presentations	business games may become stereotyped.
	commercial business games do not fit the need of lecturers
	not a good way to introducing new ideas.

Survey 3: The skills and knowledge that a general manager should know.

Abilities / skills	**knowledge**
- ability to motivate people	- finance
- ability to organize teams	- production planning & control
- ability to lead people	- marketing
- management of change	- operation management
- common sense	- personnel management
- ability to support your arguments	- technology
- time management	- market research
- ask the right questions	- research & development
- continuous improvement	- resource management
- environmental issues	- logistics

General management

- need for improving business process
- good team working principles (collaboration among departments)
- integrated supplier chain

Economics and finance

- economic environment
- financial implications of decision making
- insight into the effects of decision upon performance
- identify the difference between profit and cash
- understand and interpret financial reports
- role of planning
- survival of a business
- raising funds
- investment decisions

Strategic and operational management

- business environment
- human resources
- time management
- interpersonal relationships

Business environment

- international business environment
- business policy
- strategic management
- management of change
- technology
- changing relationships between government and business

- defining different environments in which a business should operate
- green issues
- ethics

Production planning and control

-scheduling
- inventory control

Business strategy

- deal with uncertainty
- strategic management
- competitiveness

The above mentioned are the most important issues that a manager should know for every specific area of management. Every expert gave his opinion for his/her own area of interests.

Survey 3: The advantages and disadvantages of manual business games:

Advantages	Disadvantages
- portable	- limited potential
- cheap	- do not handle numerical analysis
- easy to play	
- very effective	
- much interactive	
- very immediate	
- students gain more from the learning process	
- good for people without computer literacy	
- can easily be fitted to the teaching objectives	

Survey 3: The advantages and disadvantages of computerized games:

Advantages	Disadvantages
- user have the potential of trying different scenarios	-not portable
- fast in doing calculations	- availability of hardware is necessary
- favor individual use	- difficult to develop
- make students familiar with computer tools	- they follow a very simplistic approach
- more powerful model building	- it is difficult for a group of people to use the same terminal
- more attractive	- they are quite rigid in the reaction they give to a particular decision
	- restricted human interaction
	- programming languages put constraints to the game
	- difficult to modify
	- expensive
	- students miss much of the learning process
	- computer illiteracy is a barrier for players.

The production game - program overview

Introduction

In the production game program, it simulates a factory which contains two production lines (A and B). At the end of each period, an output report will be generated. The player can continue the game with the remained resource and start another period of production process. If the production is well organized then net profit would be gained.

Factors concerned in the production game :

1. Factory
- initial capital
- rent and rate
- factory overhead
- approved loan
- working hour

2. Labour
- assembly time
- hire worker
- fire worker
- new worker efficiency
- salary
- average salary in the market

3. Machine
- machining time
- machine age
- machine maintenance
- machine breakdown
- machine repairing
- machine price

4. Material
- raw material stock
- raw material cost
- raw material quality level

5. Product
- finished product stock
- finished product unit cost
- finished product quality level
- finished product inspection
- finished product selling price

6. Inspection

- no. of defects per sample size
- sample size per lot
- cost per inspection sample

Input variables :

1. Pay loan
   - return for the last period loan
   (> 0 and < loan value)

2. Approved loan
   - loan approved by the instructor
   (>= 0)

3. Production target (product A & B)
   - target quantity to be produced
   (> 0 and 0 <= stock)

4. Quantity of direct sell to marketing
   - quantity of product direct sell to the marketing
   (> 0 and 0 < production target)

5. Raw material quantity purchased (material A & B)
   - total raw material quantity purchased (0 to exit)
   (>= 0)

6. Raw material unit cost (material A & B)
   - average unit cost of the purchased raw material
   (> 0)

7. Raw material quality level (material A & B)
   - average quality level of the purchased raw material
   (>= 0)

8. No. of machine to be sold out (machine A & B)
   (0 <= sold out no. < exist machine)

9. No. of maintenance per period (machine A & B)
   (>= 0)

10. No. of machine to be bought (machine A & B)
   (>= 0)

11. No. of worker to be fired (worker A & B)
   (0 <= fired worker < exist worker)

12. No. of worker to hired (worker A & B)
   (>= 0)

13. Worker salary (worker A & B)
    (> 0)

14. No. of worker exchange (worker A & B)
    - no. of worker transfer from production line A to B or vice versa
    (0 <= worker exchange < exist worker)

15. No. of defects allowed (product A & B)
    - no. of defects allowed per sample size
    (0 < defects <= sample size)

16. No. of sample for inspection (product A & B)
    - no. of sample to be inspected per lot
    (0 < sample <= lot size)

17. Selling price (product A & B)
    (> 0)

18. No. of product can be sold out (product A & B)
    - product can be sold to the market
    (0 <= sold out <= product exist)

Source file : prod6.c and prodfun.c
Execution file : prod.exe
Parameter file : prodpara.txt
History file : prodhis1.txt
Output file : report file 'XXXX' (where XXXX is user defined)
: data file 'XXXX.d' (where XXXX is same as the report file)

Report type :
1. Production balance sheet.
2. Production's trading & profit & loss A/C for the end of the period

The production game - User's manual

User manual of prod.exe

Input the starting period : 'XX' - period number of the production

Input the history file name :'XXXX' - name of the history file (e.g. prodhis1)

Input the value for paid for loan : (0 ≤ paid for loan ≤ approved loan)

Approved loan : (from the instructor approved)

Production target for product A : (0 ≤ production target ≤ existing raw material)
Production target for product B : (0 ≤ production target ≤ existing raw material)
Quantity of direct selling to marketing (product A): (0 ≤ direct selling quantity ≤ stock on hand)
Quantity of direct selling to marketing (product B): (0 ≤ direct selling quantity ≤ stock on hand)

Order for raw material A quantity: (0 ≤) (Remark : It will loop back until input = 0)
Raw material A unit cost : (0 ≤)
Raw material A quality level : (0 ≤ quality level ≤ 100)

Order for raw material B quantity: (0 ≤) (Remark : It will loop back until input = 0)
Raw material B unit cost : (0 ≤)
Raw material B quality level : (0 ≤ quality level ≤ 100)

Input the number of machine A to be sold out: (0 ≤ number of machine sold out ≤ existing number of machine)
Input the number of machine B to be sold out: (0 ≤ number of machine sold out ≤ existing number of machine)
No. of maintenance for Machine A : (0 ≤)
No. of maintenance for Machine B : (0 ≤)
Input the number of machine A to be bought: (0 ≤)
Input the number of machine B to be bought: (0 ≤)

Production line A
Number of worker to be fired : (0 ≤ number of worker fired ≤ existing number of worker)
Number of worker to be hired : (0 ≤)
Salary for each worker : (0 ≤)
Number of worker interchange : (0 ≤ number of worker interchange ≤ existing number of worker - number of worker fired) (Remark : number of worker transfer from production line A to B)

Production line B
Number of worker to be fired : (0 ≤ number of worker fired ≤ existing number of worker)
Number of worker to be hired : (0 ≤)
Salary for each worker : (0 ≤)
Number of worker interchange : (0 ≤ number of worker interchange ≤ existing number of worker - number of worker fired) (Remark : number of worker transfer from production line B to A)

Product A - no. of defectives allowed : (0 ≤ defectives allowed ≤ sample size)
- no. of samples selected for testing : (0 ≤ sample size ≤ lot size)
Product B - no. of defectives allowed : (0 ≤ defectives allowed ≤ sample size)
- no. of samples selected for testing : (0 ≤ sample size ≤ lot size)

Selling price of product A : (0 ≤)
Selling price of product B : (0 ≤)
Number of product A direct sold out : (0 ≤ direct sold out ≤ stock on hand - direct selling to marketing)
Number of product B direct sold out : (0 ≤ direct sold out ≤ stock on hand - direct selling to marketing)

After completed all firms data entry:-

1. Input a file name for the output file : - files contain the production output reports ('XXXX' without extension)

2. Input the new history file name: - files contain the new production history file ('XXXX' without extension)

Report type :

1. Production balance sheet.

2. Production's trading & profit & loss A/C for the end of the period

The production game - Programmer's manual

Production

No work in progress between machining and assembly is allowed.
Target demand should not exist either 1.5 times of the machining capacity or 1.5 times of assemblying capacity.
Total number of defectives = number of defectives in raw material + number of defectives caused by machining + number of defectives caused by assemblying.
The equation of the product carrying cost is:

$$\text{Product carrying cost} = \left(\frac{pp - pt}{2} + pis\right) \times pcc$$

Where pp is the product produced,
pt is the product transferred,
pis is the product initial stock = leftover from the last period - direct selling, and
pcc is the product carrying cost per unit.

Material

The program will loop back until the material order quantity is zero.
The equation of the material carrying cost is:

$$\text{Raw material carrying cost} = \left(\frac{moh + mp - mu}{2} + ml\right) \times mcc$$

Where moh is the material on hand,
mp is the material purchased,
mu is the material used,
ml is the material leftover and mcc is the material carrying cost per unit.

Machining

The maximum age of the machines is equal to 10 periods. The machine will automatically sell out if the age of the machines is greater than 10 periods.
The equation of the existing condition age is:

$$\text{Existing conditional age} = \text{previous conditional age} + \sum_{i=1}^{n} \frac{(1-p)^i}{i}$$

Where p is the maintenance effect and n is the number of maintenance per period.
The number of machine breakdown is equal to the integer value of (breakdown factor) ^{n} -1.
Where n is the age number of the machine.
The breakdown repairing time is equal to the integer value of random number between the minimum machine repairing time and maximum machine repairing time.
The equation of number of defective produced is:
Number of defective produced = number of product produced × (quality factor)^n
Where n is the age number of the machine.

Assemblying

The equation of the efficiency of the new employed worker is:
Efficiency of the new employed worker = efficiency of the trained worker × learning factor.

Hiring cost = hiring overhead + number of new hiring workers × hiring cost per head
Firing cost = number of workers firing × firing cost per head
The number of worker leaving = existing workers in production line × (1- leaving factor)

$$\text{Leaving factor} = \frac{\text{production line wages}}{\text{average industrial wages}}$$

If the leaving factor is greater than 1, the leaving factor is set to a preset value (e.g. 0.005).
The equation of the turnover index is:

$$\text{Turnover index} = \frac{\text{new employees} + \text{employees leaving}}{\text{total number of employees on each production line}}$$

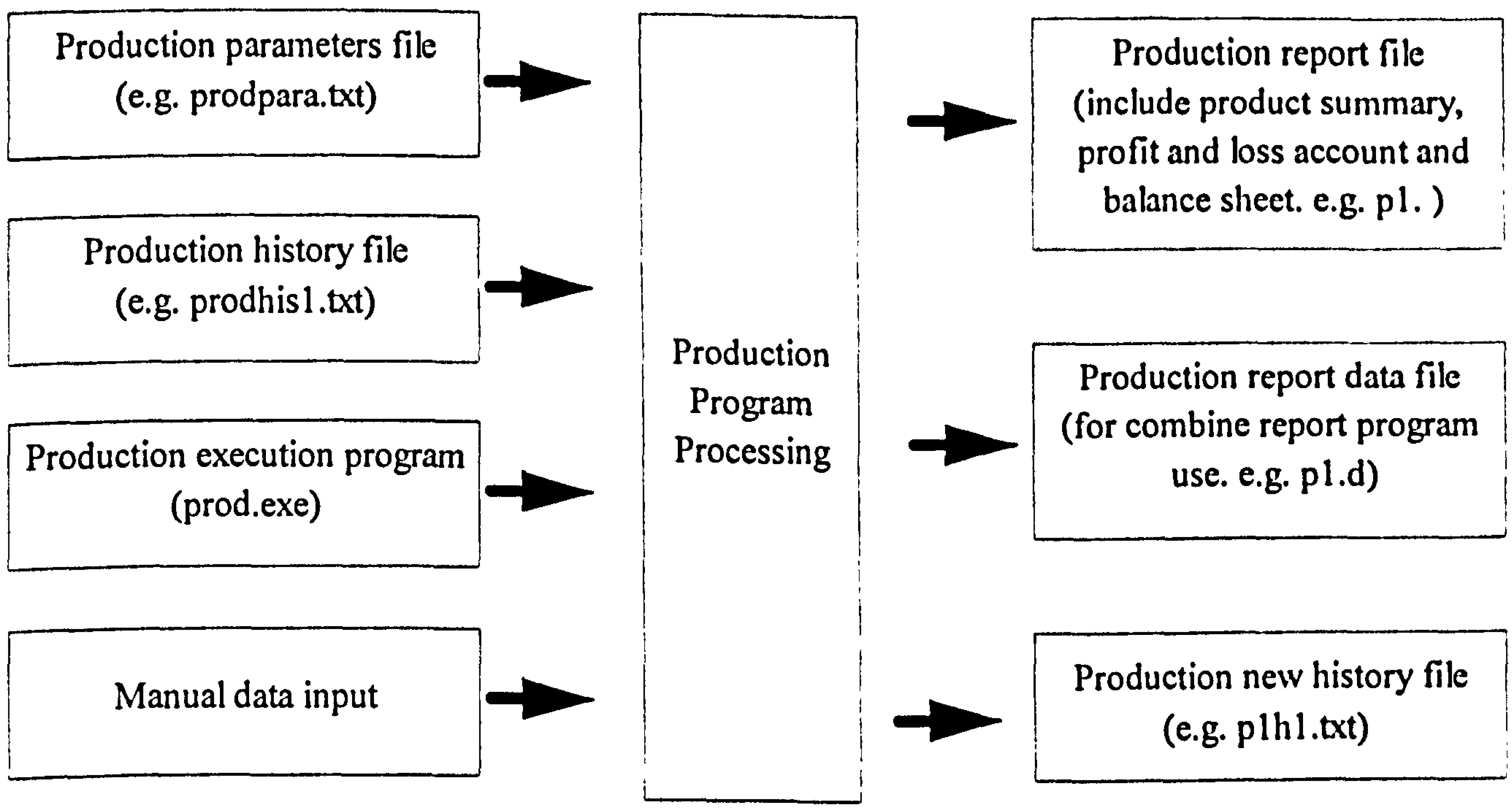
Defectives in machining = product produced × turnover index × turnover quality factor

Inspection
Assume the inspection time is negligible.
The equation of the average inspection number (AIN) per lot is n + [1-P(a)](N-n)

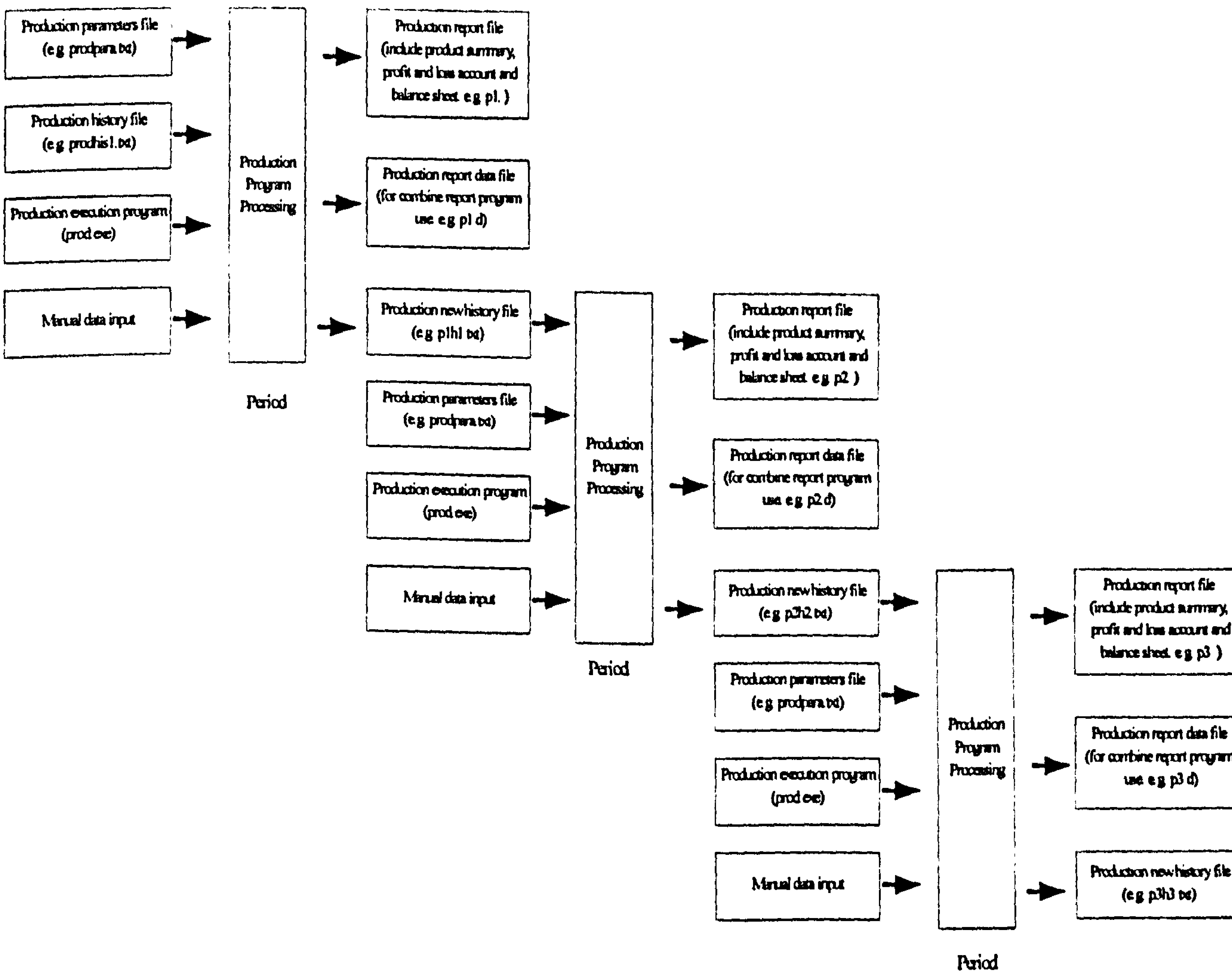
$$\text{and } P(a) = \frac{e^{-PD} PD^{a}}{a!}$$

Where a is number of defective allowed per lot,
PD is the actual defective rate,
N is the lot size and n is the sample size.
The equation of the average outgoing quality is P(a)[(N-n)PD]

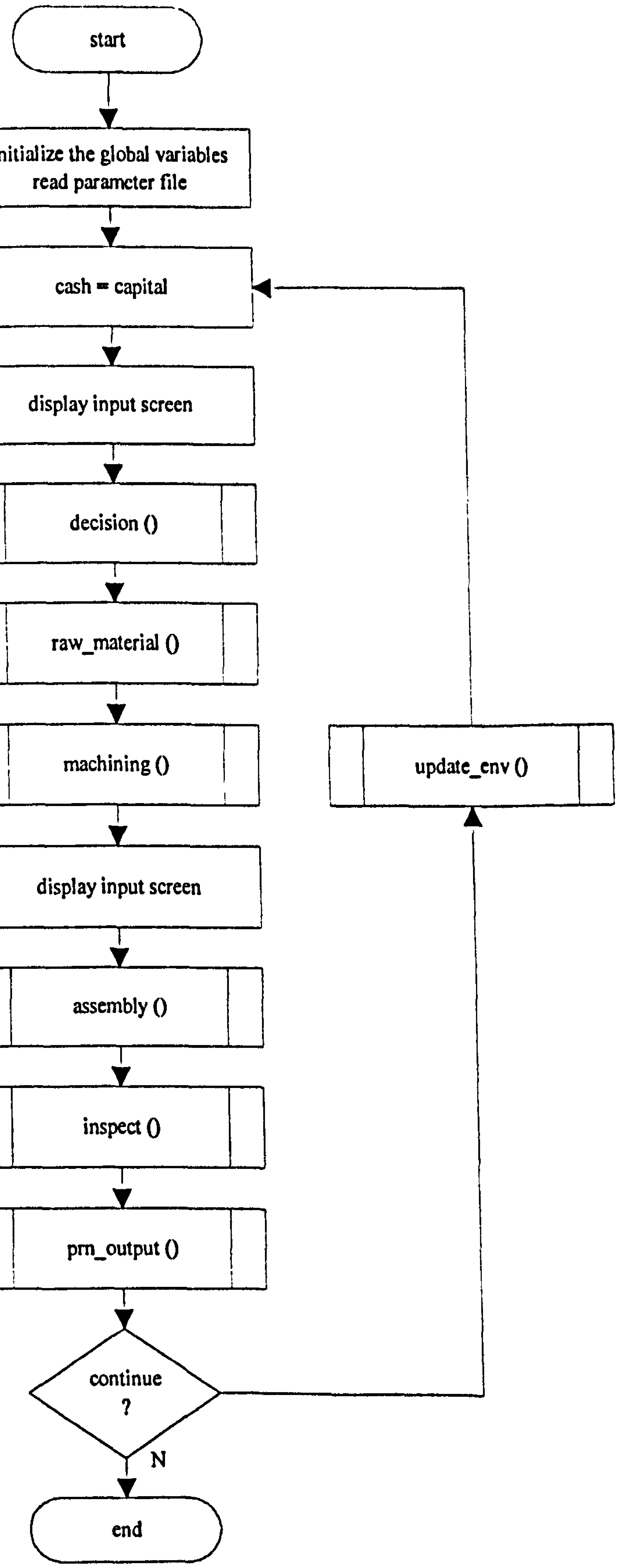
Production program data flow chart

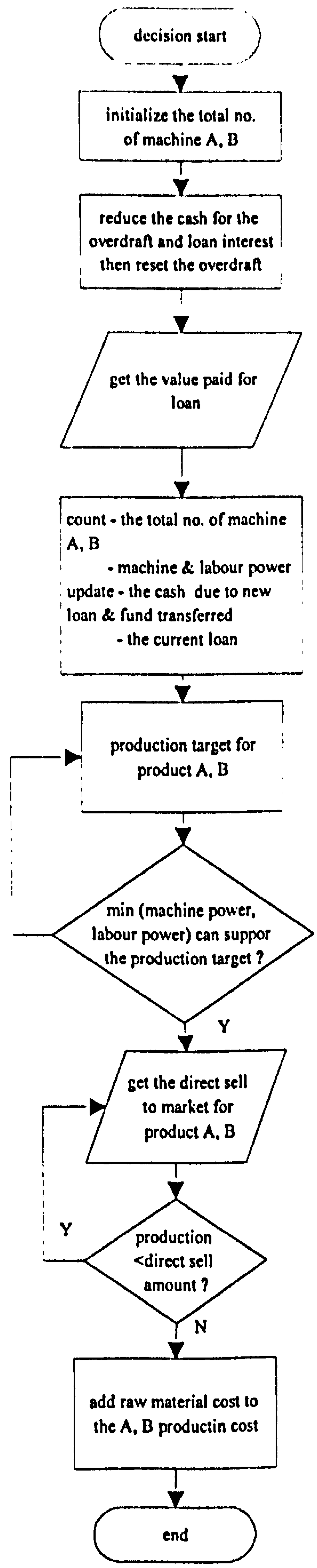


Production program in chain data flowchart

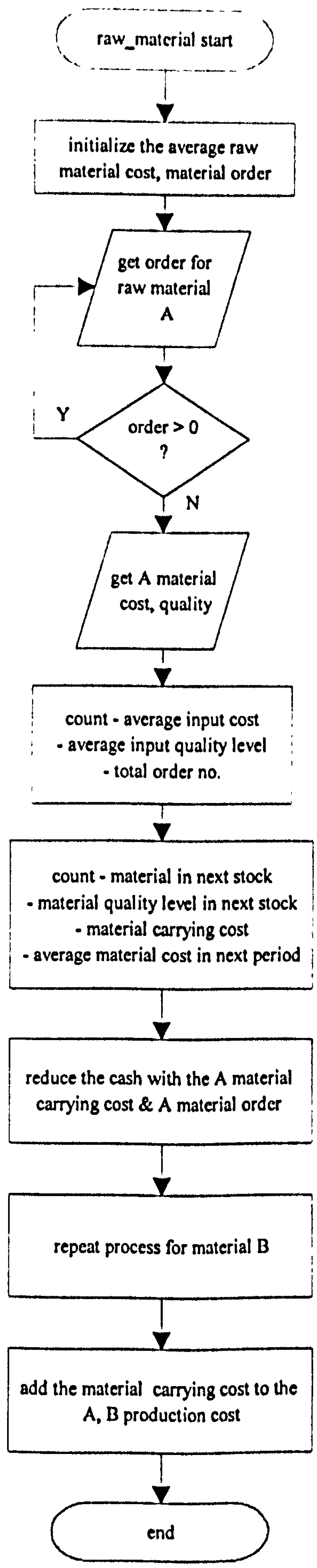


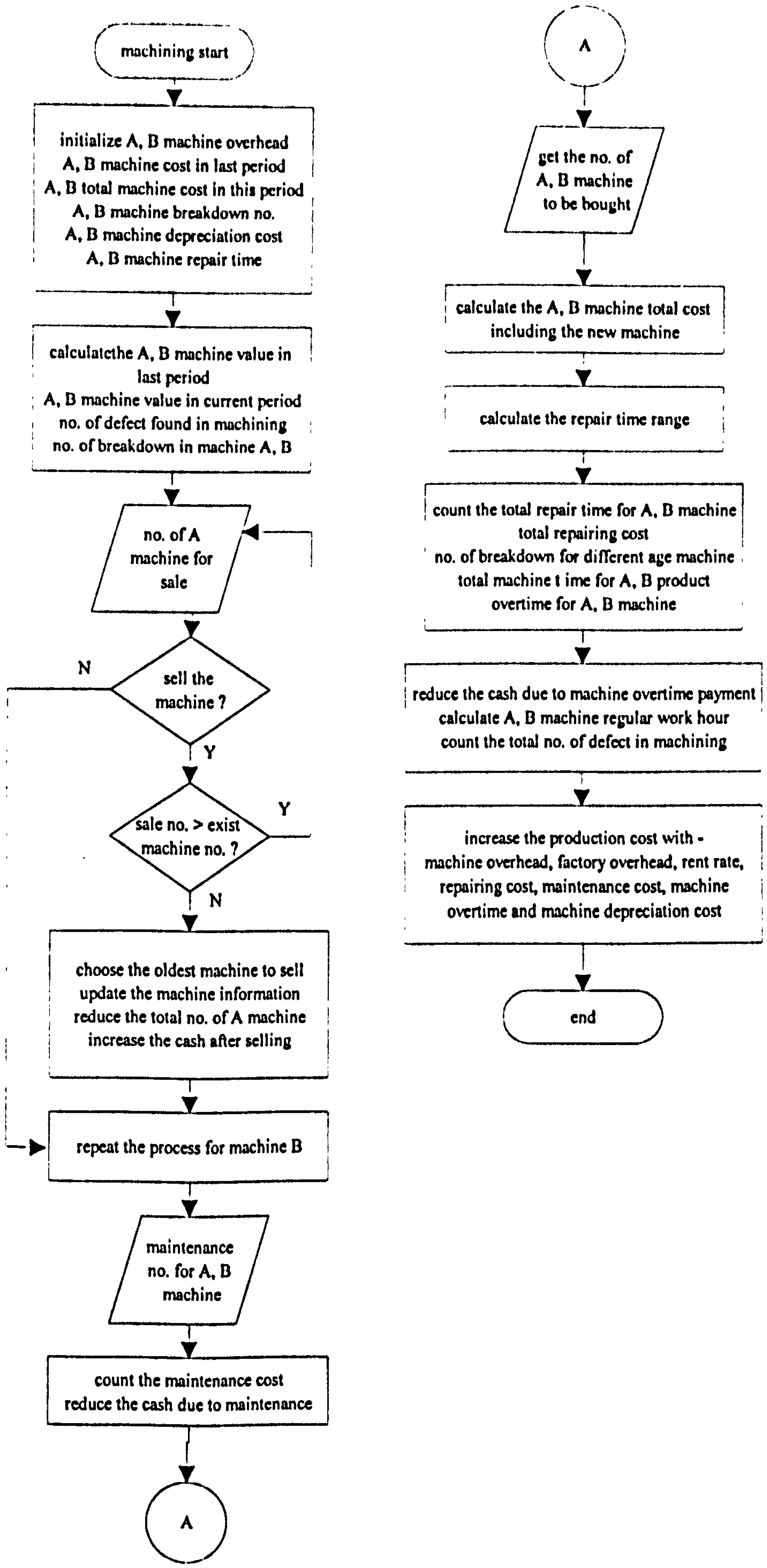
The production game - Flow Chart

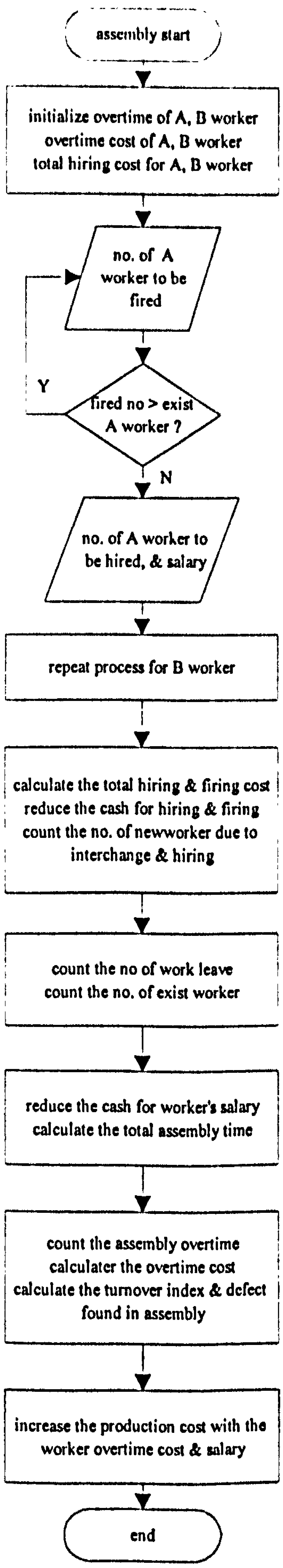


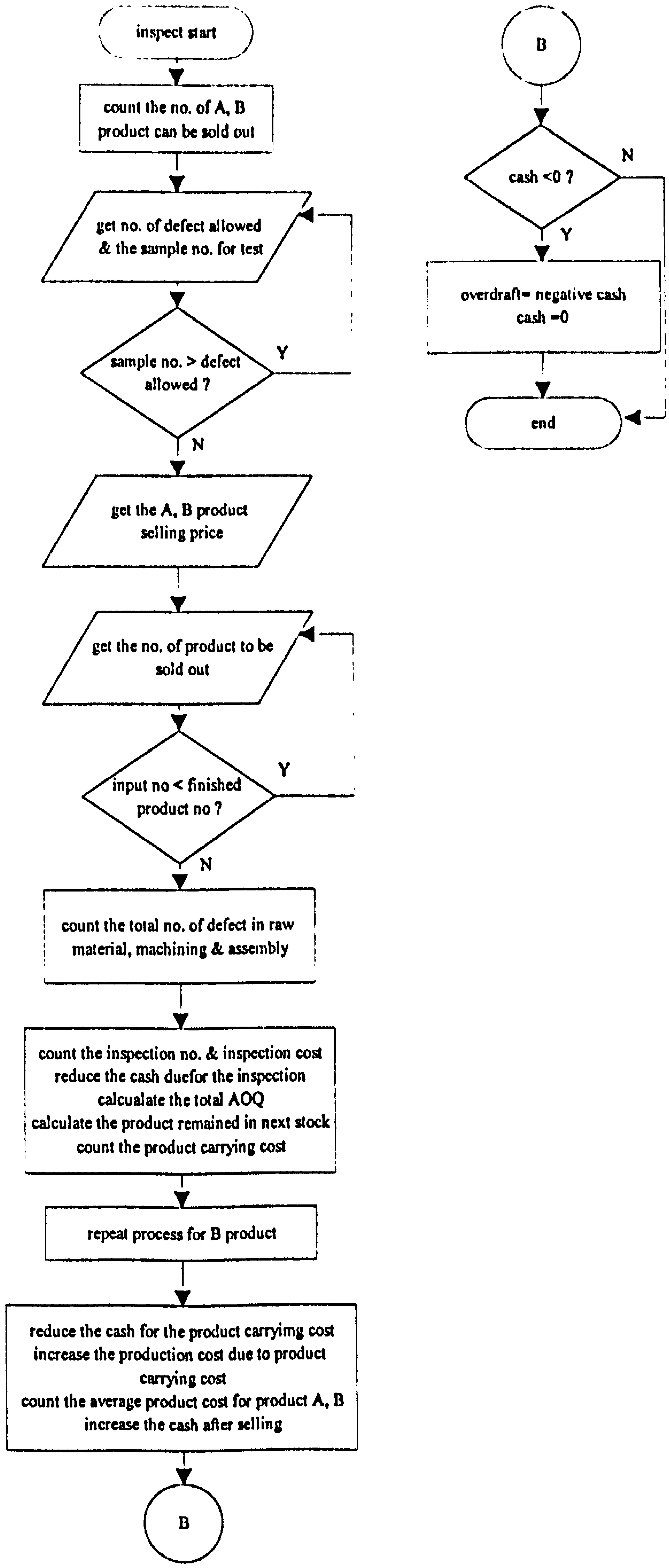


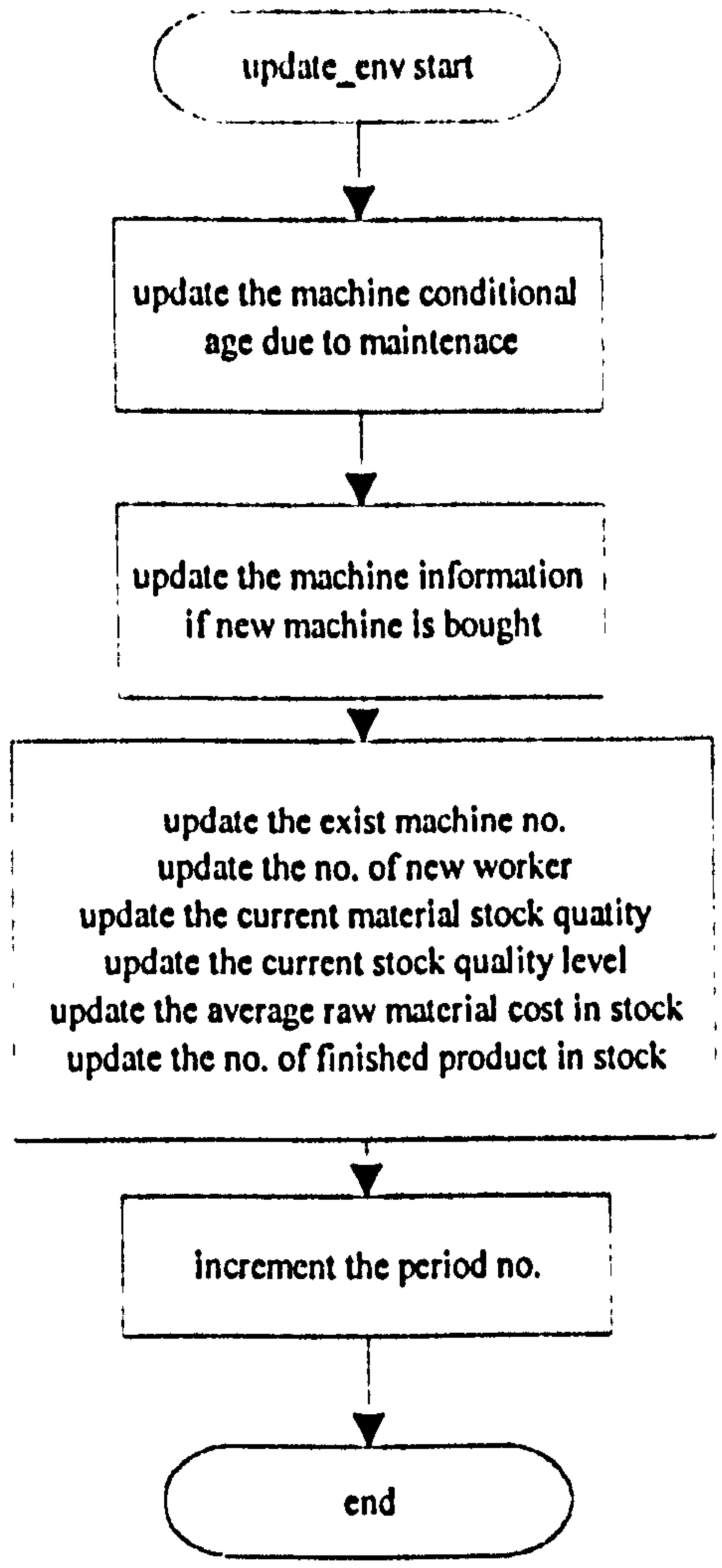
decision start
initialize the total no. of machine A, B
reduce the cash for the overdraft and loan interest then reset the overdraft
get the value paid for loan
count - the total no. of machine A, B
- machine & labour power
update - the cash due to new loan & fund transferred
- the current loan
production target for product A, B
min (machine power, labour power) can suppor the production target ?
Y
get the direct sell to market for product A, B
Y
production <direct sell amount ?
N
add raw material cost to the A, B productin cost
end


raw_material start
initialize the average raw material cost, material order
get order for raw material A
Y
order > 0 ?
N
get A material cost, quality
count - average input cost
- average input quality level
- total order no.
count - material in next stock
- material quality level in next stock
- material carrying cost
- average material cost in next period
reduce the cash with the A material carrying cost & A material order
repeat process for material B
add the material carrying cost to the A, B production cost
end


machining start
initialize A, B machine overhead
A, B machine cost in last period
A, B total machine cost in this period
A, B machine breakdown no.
A, B machine depreciation cost
A, B machine repair time
calculatethe A, B machine value in last period
A, B machine value in current period
no. of defect found in machining
no. of breakdown in machine A, B
no. of A machine for sale
sell the machine ?
N
Y
sale no. > exist machine no. ?
Y
N
choose the oldest machine to sell
update the machine information
reduce the total no. of A machine
increase the cash after selling
repeat the process for machine B
maintenance no. for A, B machine
count the maintenance cost
reduce the cash due to maintenance
A
A
get the no. of A, B machine to be bought
calculate the A, B machine total cost including the new machine
calculate the repair time range
count the total repair time for A, B machine
total repairing cost
no. of breakdown for different age machine
total machine t ime for A, B product
overtime for A, B machine
reduce the cash due to machine overtime payment
calculate A, B machine regular work hour
count the total no. of defect in machining
increase the production cost with -
machine overhead, factory overhead, rent rate,
repairing cost, maintenance cost, machine
overtime and machine depreciation cost
end


assembly start
initialize overtime of A, B worker
overtime cost of A, B worker
total hiring cost for A, B worker
no. of A
worker to be
fired
Y
fired no > exist
A worker ?
N
no. of A worker to
be hired, & salary
repeat process for B worker
calculate the total hiring & firing cost
reduce the cash for hiring & firing
count the no. of newworker due to
interchange & hiring
count the no of work leave
count the no. of exist worker
reduce the cash for worker's salary
calculate the total assembly time
count the assembly overtime
calculater the overtime cost
calculate the turnover index & defect
found in assembly
increase the production cost with the
worker overtime cost & salary
end


inspect start
count the no. of A, B product can be sold out
get no. of defect allowed & the sample no. for test
sample no. > defect allowed ?
Y
N
get the A, B product selling price
get the no. of product to be sold out
input no < finished product no ?
Y
N
count the total no. of defect in raw material, machining & assembly
count the inspection no. & inspection cost
reduce the cash duefor the inspection
calcualate the total AOQ
calculate the product remained in next stock
count the product carrying cost
repeat process for B product
reduce the cash for the product carryimg cost
increase the production cost due to product carrying cost
count the average product cost for product A, B
increase the cash after selling
B
B
cash <0 ?
N
Y
overdraft= negative cash
cash =0
end


update_env start
update the machine conditional age due to maintenace
update the machine information if new machine is bought
update the exist machine no.
update the no. of new worker
update the current material stock quatity
update the current stock quality level
update the average raw material cost in stock
update the no. of finished product in stock
increment the period no.
end

The production game - Program listing (part 1)

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#include<math.h>
#define TRUE 1
#define FALSE 0
#define E 2.718281828  /*---- declare global variables in parameter file --*/
float capital=0;              /* capital */
float cash=0;                /* cash on hand */
float loan_rate=0;             /* loan interest rate */
float overdraft_rate=0;         /* emergency over draft interest rate */
float pay_loan=0;          /* fund transfer from marketing */
float pay_overdraft=0;
float hour_per_period=0;        /* no. of working hours */
float firing_cost=0;           /* labour firing cost/person */
float hiring_overhead=0;        /* labour hiring cost/overhead */
float hire_cost=0;            /* labour hiring cost/person */
float average_wage=0;          /* average industrial wages */
float factory_overhead=0;       /* overhead cost of the factory */
float overtime_pay=0;          /* overtime pay per hour */
float rent_rate=0;            /* rent and rates  */
float A_product_level;              /* product A quality level in stock */
long int A_stock_material=0;       /* raw material A in stock */
float A_stock_material_level=0;  /* material quality level in stock */
long int A_stock_product=0;        /* product A in stock */
float A_stock_carry_cost=0;        /* stock carrying cost/unit/period (A) */
float A_stock_ave_raw_cost=0;      /* average cost of raw material A in stock */
float A_stock_ave_product_cost=0; /* average cost of product A in stock */
float A_machine_time=0;          /* A machine time per product */
float Amachine_info[10][6]={0};     /* Amachine_info[i][0]=no. of machine  */
/* Amachine_info[i][1]=machine actual age  Amachine_info[i][2]=machine conditional age */
/* Amachine_info[i][3]=no. of breakdown    Amachine_info[i][4]=no. of defects  */
/* Amachine_info[i][5]=depreciation cost */
float A_machine_price=0;         /* price for machine A */
float A_machine_deprec_rate=0;  /* machine A depreciation % */
float A_machine_overhead=0;     /* machine A overhead cost/unit */
float A_mainten_cost=0;         /* machine A maintenance cost */
float A_mainten_effect=0;        /* machine A maintenance effect */
float A_breakdn_factor=0;        /* machine A break down factor */
float A_quality_factor=0;        /* machine A quality factor */
float A_min_repair_time=0;       /* machine A min repairing time */
float A_max_repair_time=0;       /* machine A max repairing time */
float A_repair_cost=0;         /* machine A repairing cost per hour */
long int A_worker=0;              /* no. of worker for part A */
float A_assemble_time=0;         /* assembling time for part A */
float A_turnover_quality_factor=0; /* turnover quality factor for A */
float A_learn_curve=0;           /* learning curve for A workers */
int A_lot_size=0;              /* log size for product A */
float A_inspect_cost=0;          /* inspection cost/unit for A */
float B_product_level;              /* product B quality level in stock */
long int B_stock_material=0;       /* stock quantity of part B */
float B_stock_material_level=0;  /* material quality level in stock */
long int B_stock_product=0;        /* product B in stock */
float B_stock_carry_cost=0;        /* stock carrying cost/unit/period (B) */
float B_stock_ave_raw_cost=0;     /* average cost of raw material B in stock */
```

```
float B_stock_ave_product_cost=0; /* average cost of product B in stock */
float B_machine_time=0;      /* machine time for product B */
float Bmachine_info[10][6]={0}; /* Bmachine_info[i][0]=no. of machine */
/* Bmachine_info[i][1]=machine actual age  Bmachine_info[i][2]=machine conditional age */
/* Bmachine_info[i][3]=no. of breakdown   Bmachine_info[i][4]=no. of defects */
/* Bmachine_info[i][5]=depreciation cost */
float B_machine_price=0;      /* price for machine B */
float B_machine_deprec_rate=0; /* machine B depreciation % */
float B_machine_overhead=0;    /* machine B overhead cost/unit */
float B_mainten_cost=0;       /* machine B maintenance cost */
float B_mainten_effect=0;      /* machine B maintenace effect */
float B_breakdn_factor=0;      /* machine B break down factor */
float B_quality_factor=0;      /* machine B quality factor */
float B_min_repair_time=0;     /* machine B min repairing time */
float B_max_repair_time=0;     /* machine B max repairing time */
float B_repair_cost=0;        /* machine B repairing cost per hour */
long int B_worker=0;          /* no. of worker for part B */
float B_assemble_time=0;       /* assembling time for part B */
float B_turnover_quality_factor=0; /* turnover quality factor for B */
float B_learn_curve=0;        /* learning curve for B workers */
int B_lot_size=0;            /* log size for product B */
float B_inspect_cost=0;        /* inspection cost/unit for B */
int period=1;    /*--------- variables in report -----------*/
float premise=0;             /* from parameter file */
float A_T_machine_cost=0;       /* exist all A machines cost */
float A_T_machine_deprec_cost=0; /* accumulated depreciation */
float B_T_machine_cost=0;       /* exist all B machines cost */
float B_T_machine_deprec_cost=0; /* accumulated depreciation */
float overdraft=0;            /* over draft */
float loan=0;               /* approved loan */
float A_ave_cost=0;            /* average raw material A cost (input) */
float B_ave_cost=0;            /* average raw material B cost (input) */
float A_T_product_price=0;       /* product A total selling price */
float B_T_product_price=0;       /* product B total selling price */
float A_T_product_cost=0;    /* Total price for direct sell to marketing */
float B_T_product_cost=0;    /* Total price for direct sell to marketing */
long int A_sold_product=0;     /* no. of product A to be bought */
long int B_sold_product=0;     /* no. of product B to be bought */
long int A_order_material=0;        /* purchase of A raw material no. */
long int B_order_material=0;        /* purchase of B raw material no. */
float A_T_worker_wages=0;        /* total wages for worker A */
float B_T_worker_wages=0;        /* total wages for worker B */
float A_T_inspect_cost=0;       /* total inspection cost for product A */
float B_T_inspect_cost=0;       /* total inspection cost for product B */
int A_sold=0;               /* no. of A machine sold out */
int B_sold=0;               /* no. of B machine sold out */
int A_buy=0;              /* no. of machine A are bought */
int B_buy=0;              /* no. of machine B are bought */
float A_T_mainten_cost=0;       /* total maintenance cost for A machines */
float B_T_mainten_cost=0;       /* total maintenance cost for B machines */
float A_T_repair_cost=0;       /* total reparing cost for A machines */
float B_T_repair_cost=0;       /* total reparing cost for B machines */
float A_material_carry_cost=0;   /* carrying cost for raw material A */
float B_material_carry_cost=0;   /* carrying cost for raw material B */
float A_product_carry_cost=0;    /* carrying cost for product A */
float B_product_carry_cost=0;    /* carrying cost for product B */
float A_stock_product_defect=0; /*  defect of product in stock */
float B_stock_product_defect=0; /* defect of product in stock */
```

```
float A_T_fire_cost=0;        /* total firing cost for production line A */
float B_T_fire_cost=0;        /* total firing cost for production line B */
float A_T_hire_cost=0;        /* total hiring cost for production line A */
float B_T_hire_cost=0;        /* total hiring cost for production line B */
int A_repair_time=0;          /* machine A repair time */
int B_repair_time=0;          /* machine B repair time */
int A_exist_worker=0;         /* exist A worker for the current period */
int B_exist_worker=0;         /* exist B worker for the current period */
int A_new_worker=0;                  /* exist new worker for the current period */
int B_new_worker=0;                  /* exist new worker for the current period */
int A_breakdown=0;                   /* total no. of breakdown for machine A */
int B_breakdown=0;                   /* total no. of breakdown for machine B */
float A_total_machine_time=0;  /* total working hour for machine A */
float B_total_machine_time=0;  /* total working hour for machine B */
float A_machine_overtime=0;    /* overtime working hour for machine A */
float B_machine_overtime=0;    /* overtime working hour for machine B */
float A_machine_regular_hour=0; /* regular working hour for machine A */
float B_machine_regular_hour=0; /* regular working hour for machine B */
float A_total_worker_time=0;   /* total working hour for labour in line A */
float B_total_worker_time=0;   /* total working hour for labour in line B */
float A_worker_overtime=0;     /* overtime working hour for worker A */
float B_worker_overtime=0;     /* overtime working hour for worker B */
float A_worker_regular_hour=0; /* regular working hour for worker A */
float B_worker_regular_hour=0; /* regular working hour for worker B */
int A_ave_inspect_no=0;        /* average inspection no. of product A */
int B_ave_inspect_no=0;        /* average inspection no. of product B */
int A_sample=0;                /* sample no. of product A */
int B_sample=0;                /* sample no. of product B */
int A_defect_allow=0;          /* no. of defect allowed per lot of A */
int B_defect_allow=0;          /* no. of defect allowed per lot of B */
float A_AOQ=0;           /* product A average output quality (new produced) */
float B_AOQ=0;           /* product B average output quality (new produced) */
float A_T_AOQ=0;   /* overall AOQ of product A including the product in stock */
float B_T_AOQ=0;   /* overall AOQ of product B including the product in stock */
float A_sell_market_cost=0;    /* cash return from direct sell to marketing */
float B_sell_market_cost=0;    /* cash return from direct sell to marketing */
float A_sell_price=0;          /* product A selling price */
float B_sell_price=0;          /* product B selling price */
float A_salary=0;              /* worker A salary */
float B_salary=0;              /* worker B salary */
float NA_stock_material_level=0; /* next period material quality level in stock */
float NB_stock_material_level=0; /* next period material quality level in stock */
/*----------- variables in programming --------------*/
long int NB_stock_product=0;   /* product B in next stock */
long int NA_stock_product=0;   /* product A in next stock */
long int A_target=0;           /* product A production target */
long int B_target=0;           /* product B production target */
long int NA_material_stock=0;  /* A raw material in stock for next period */
long int NB_material_stock=0;  /* B raw material in stock for next period */
int A_assemble_defect=0;       /* total defects from in assembly A / lot size */
int B_assemble_defect=0;       /* total defects from in assembly B / lot size */
int A_machine_defect=0;        /* total defects from machining / lot size */
int B_machine_defect=0;        /* total defects from machining / lot size */
float NA_stock_ave_raw_cost=0; /* next period average material cost in stock */
float NB_stock_ave_raw_cost=0; /* next period average material cost in stock */
long int A_sell_market=0;          /* product A direct sell to marketing */
long int B_sell_market=0;          /* product B direct sell to marketing */
int A_maintenance=0;          /* maintenance frequency for A machine */
```

```
int B_maintenance=0;        /* maintenance frequency for B machine */
int A_hire=0;            /* no. of A workers are hired */
int B_hire=0;            /* no. of B workers are hired */
int A_fire=0;            /* no. of A workers are fired */
int B_fire=0;            /* no. of B workers are fired */
int A_leave=0;            /* no. of A workers left */
int B_leave=0;            /* no. of B workers left */
int A_change=0;            /* no. of A workers are changed to B */
int B_change=0;            /* no. of B workers are changed to A */
int A_total_machine=0;        /* no. of working A machine */
int B_total_machine=0;        /* no. of working B machine */
main()
{
        clrscr();
   printf("Input the starting period  :  ");
        scanf("%d",&period);
                readpara();
        readhis();
        cash=capital;        /* cash on hand = capital in parameter file */
        clrscr();
                scr_decision();
                scr_material();
                scr_machine();
        decision();
                raw_material();
                machining();
                clrscr();
                scr_labour();
        scr_inspect();
        scr_selling();
                assembly();
        inspect();
                prn_output();
        update_env();
                writehis();
}
readpara() /*-------- read the parameter file --------*/
{  FILE *para;
        char string[50];
   clrscr();
        printf("Initializing the system .......\n");
   if ((para=fopen("prodpara.txt","r"))==NULL)
   printf("<<ERROR : Parameter file open error !>>\n");
   fscanf(para,"%s %f",string,&premise);
   fscanf(para,"%s %f",string,&loan_rate);
   fscanf(para,"%s %f",string,&overdraft_rate);
   fscanf(para,"%s %f",string,&hour_per_period);
   fscanf(para,"%s %f",string,&firing_cost);
   fscanf(para,"%s %f",string,&hiring_overhead);
   fscanf(para,"%s %f",string,&hire_cost);
   fscanf(para,"%s %f",string,&average_wage);
   fscanf(para,"%s %f",string,&factory_overhead);
   fscanf(para,"%s %f",string,&overtime_pay);
        fscanf(para,"%s %f",string,&rent_rate);
                /*------ for production line A ---------*/
   fscanf(para,"%s %f",string,&A_stock_carry_cost);     /* stock carrying cost/unit/period (A) */
        fscanf(para,"%s %f",string,&A_machine_time);     /* machine time for production A */
   fscanf(para,"%s %f",string,&A_machine_price);      /* price for machine A */
```

```
        fscanf(para,"%s %f",string,&A_machine_deprec_rate);     /* machine A depreciation % */
   fscanf(para,"%s %f",string,&A_machine_overhead);        /* machine A overhead cost/unit */
   fscanf(para,"%s %f",string,&A_mainten_cost);      /* machine A maintenance cost */
   fscanf(para,"%s %f",string,&A_mainten_effect);     /* machine A maintenace effect */
   fscanf(para,"%s %f",string,&A_breakdn_factor);     /* machine A break down factor */
   fscanf(para,"%s %f",string,&A_quality_factor);     /* machine A quality factor */
   fscanf(para,"%s %f",string,&A_min_repair_time);     /* machine A min repairing time */
   fscanf(para,"%s %f",string,&A_max_repair_time);     /* machine A max repairing time */
   fscanf(para,"%s %f",string,&A_repair_cost);    /* machine A repairing cost */
   fscanf(para,"%s %f",string,&A_assemble_time);     /* assembling time for part A */
   fscanf(para,"%s %f",string,&A_turnover_quality_factor);  /* turnover quality factor for A */
   fscanf(para,"%s %f",string,&A_learn_curve);     /* learning curve for A workers */
   fscanf(para,"%s %ld",string,&A_lot_size);     /* log size for product A */
   fscanf(para,"%s %f",string,&A_inspect_cost);     /* inspection cost/unit for A */
              /*------ for production line B --------*/
        fscanf(para,"%s %f",string,&B_stock_carry_cost);    /* stock carrying cost/unit/period (B)
*/
        fscanf(para,"%s %f",string,&B_machine_time);    /* machine time for production B */
   fscanf(para,"%s %f",string,&B_machine_price);     /* price for machine B */
        fscanf(para,"%s %f",string,&B_machine_deprec_rate);     /* machine B depreciation % */
   fscanf(para,"%s %f",string,&B_machine_overhead);        /* machine B overhead cost/unit */
   fscanf(para,"%s %f",string,&B_mainten_cost);      /* machine B maintenance cost */
   fscanf(para,"%s %f",string,&B_mainten_effect);     /* machine B maintenace effect */
   fscanf(para,"%s %f",string,&B_breakdn_factor);     /* machine B break down factor */
   fscanf(para,"%s %f",string,&B_quality_factor);     /* machine B quality factor */
   fscanf(para,"%s %f",string,&B_min_repair_time);     /* machine B min repairing time */
   fscanf(para,"%s %f",string,&B_max_repair_time);     /* machine B max repairing time */
   fscanf(para,"%s %f",string,&B_repair_cost);    /* machine B repairing cost */
   fscanf(para,"%s %f",string,&B_assemble_time);     /* assembling time for part B */
   fscanf(para,"%s %f",string,&B_turnover_quality_factor);  /* turnover quality factor for B */
   fscanf(para,"%s %f",string,&B_learn_curve);     /* learning curve for B workers */
   fscanf(para,"%s %ld",string,&B_lot_size);     /* log size for product B */
   fscanf(para,"%s %f",string,&B_inspect_cost);     /* inspection cost/unit for B */
   fclose(para); }

/*-----------------------------------------------------------\
/      function : get user input with coordinates         \
/         input : y-coordinate, x-coordinate, input type   \
/ allow input type : integer, long integer and floating point \
/----------------------------------------------------------*/
float getin(int row,int col,char type)
{  char string[20];
   int length=0,error;
   float num;
   do
        {  error=0;
        gotoxy(col,row);  clreol();
        gotoxy(col,row);  gets(string);
        length=strlen(string);
        switch(type)
        {  case 'i' : num=atoi(string); break;
           case 'l' : num=atol(string); break;
           case 'f' : num=atof(string); break;   }
        if (num==0)
           if (string[0]!='0' || length>1)
                 error =1;
   }while (error);
   switch(type)
```

```
{  case 'i' : return((int)num);
       case 'l' : return((long int)num);
       case 'f' : return(num);    } }

/*---------------------------------------------*/
long fact(int num)     /* return the factorial of a input number */
{  int i;
   long ans=1;
   for(i=1;i<=num;i++)
          ans=ans*i;
   return ans;    }

/*------------------------------------\
/ int c - the no. of defects allowed  \
/------------------------------------*/
float pa(int c, float percent_defect)  /* calculate the pa */
{  int i;
   float pa=0;
   for(i=0;i<c;i++)
          pa+=pow(percent_defect,i)/(pow(E,percent_defect)*fact(i));
   return(pa);    }

/*-----------------------------------------------------------\
/ function : - get the value paid for loan                   \
/          - request or return the loan                      \
/          - set the quantity of direst selling to marketing  \
/          - set the production target                       \
/-----------------------------------------------------------*/
decision()
{  float fund, min_pay_loan, new_loan=0;
          float labour, machine ;
          int i;
          A_total_machine=0; B_total_machine=0; /* initialize the total machine no. */
   pay_overdraft=(1+overdraft_rate)*overdraft; /* autopay the overdraft interest */
   overdraft=0;
          cash=cash-pay_overdraft;
          cash=-(loan*loan_rate);
   do
          {  pay_loan=getin(1,55,'f');    /* get fund transfer from marketing */
          }while(pay_loan>loan);
          cash-=pay_loan;         /* add fund to the current capital */
          loan-=pay_loan;
          new_loan=getin(2,17,'f');       /* request for new loan */
   loan=loan+new_loan;    /* update the loan values */
   cash+=new_loan;              /* update the current capital */
   for(i=0;i<10;i++)          /* max. 10 different age of machine */
   {  A_total_machine+=Amachine_info[i][0];  /* count the total no. of machine A */
          B_total_machine+=Bmachine_info[i][0]; }  /* count the total no. of machine B */
   machine=1.5*A_total_machine*hour_per_period/A_machine_time;  /* A machine power */
   labour=1.5*A_worker*hour_per_period/A_assemble_time;  /* A labour power */
   do
   {  A_target=getin(3,35,'l');     /* get the production target for product A */
   }while(A_stock_material<A_target || A_target>min(machine,labour));
   machine=1.5*B_total_machine*hour_per_period/B_machine_time; /* B machine power */
   labour=1.5*B_worker*hour_per_period/B_assemble_time;  /* B labour power */
   do
   {  B_target=getin(4,35,'l');     /* get the production target for product B */
   }while(B_stock_material<B_target || B_target>min(machine,labour));
```

```
do
{   A_sell_market=getin(5,53,'l'); /* get the no. of direct sell to marketing */
}while (A_target<A_sell_market);
do
{   B_sell_market=getin(6,53,'l'); /* get the no. of direct sell to marketing */
}while (B_target<B_sell_market);
  /* count the total product cost in raw material */
A_T_product_cost=A_target*A_stock_ave_raw_cost;
B_T_product_cost=B_target*B_stock_ave_raw_cost;    }
/*-----------------------------------------------------------------------------\
/ function - order the raw material                                             \
/        - calculate the stock raw material remained in the next period          \
/        - calculate the average material cost in the next period stock          \
/        - calculate the average quality level of material in the next stock     \
/        - cash remained after ordering the raw material                         \
/-----------------------------------------------------------------------------*/
raw_material()
{  float cost=0, level=0, ave_level=0, temp=0;
        long int order=0;
        A_order_material=0; B_order_material=0; /* initialize the order  */
        A_ave_cost=0;  B_ave_cost=0;        /* initialize the order average cost */
   do     /* for raw material A */
   {  order=getin(8,36,'l');        /* get the quantity of raw material A */
        if (order>0)
        {  cost=getin(9,36,'f');    /* get the unit cost of raw material A */
           level=getin(10,36,'f');   /* get the quality level of material A */

        A_ave_cost=(A_ave_cost*A_order_material+order*cost)/(A_order_material+order);
                                         /* cal the average unit cost of input */

        ave_level=(ave_level*A_order_material+order*level)/(A_order_material+order);
                                         /* cal the average level of input */
        A_order_material+=order;  }        /* accumulate the input order */
        }while(order>0);
   NA_material_stock=A_stock_material+A_order_material-A_target;
        if(NA_material_stock>0)
   {  temp=A_order_material*ave_level;
                NA_stock_material_level=(temp+(A_stock_material-
A_target)*A_stock_material_level)/NA_material_stock;
                temp=(A_stock_material-A_target)*A_stock_ave_raw_cost;  /* remained average
unit cost */

        NA_stock_ave_raw_cost=(temp+A_ave_cost*A_order_material)/NA_material_stock;
        }
        else
   {  NA_stock_material_level=0;
        NA_stock_ave_raw_cost=0;     }
        A_material_carry_cost=(A_stock_material-
A_target/2+A_order_material/2)*A_stock_carry_cost;
  temp=A_order_material*A_ave_cost;    /* cost for order the raw material A */
  cash=cash-temp-A_material_carry_cost;
  do  /* for raw material B */
  {  order=getin(11,36,'l');       /* get the quantity of raw material A */
        if (order>0)
        {  cost=getin(12,36,'f');    /* get the unit cost of raw material A */
           level=getin(13,36,'f');   /* get the quality level of material A */

       B_ave_cost=(B_ave_cost*B_order_material+order*cost)/(B_order_material+order);
```

```
                                        /* cal the average unit cost of input */

        ave_level=(ave_level*B_order_material+order*level)/(B_order_material+order);
                                        /* cal the average level of input */
          B_order_material+=order; }        /* accumulate the order */
        }while(order>0);
   NB_material_stock=B_stock_material+B_order_material-B_target;
        if(NB_material_stock>0)
   {  temp=B_order_material*ave_level;
                NB_stock_material_level=(temp+(B_stock_material-
B_target)*B_stock_material_level)/NB_material_stock;
                temp=(B_stock_material-B_target)*B_stock_ave_raw_cost; /* remained average
unit cost */

        NB_stock_ave_raw_cost=(temp+B_ave_cost*B_order_material)/NB_material_stock;
        }
        else
   {  NB_stock_material_level=0;
        NB_stock_ave_raw_cost=0;   }
        B_material_carry_cost=(B_stock_material-
B_target/2+B_order_material/2)*B_stock_carry_cost;
   temp=B_order_material*B_ave_cost;   /* cost for order the raw material B */
   cash=cash-temp-B_material_carry_cost;
   A_T_product_cost+=A_material_carry_cost;
   B_T_product_cost+=B_material_carry_cost;   }
/*----------------------------------------------------*/
int old(char ch)     /* find out the oldest machine */
{  int location,j;   float max=0;
        if (ch=='a')
   {  for(j=0;j<10;j++)          /* check the whole array */
          if (max<Amachine_info[j][2])
          {  max=Amachine_info[j][2];    /* update the oldest conditional age */
               location=j;   }  }
        else
   {  for(j=0;j<10;j++)          /* check the whole array */
          if (max<Bmachine_info[j][2])
          {  max=Bmachine_info[j][2];    /* update the oldest conditional age */
               location=j;   } }
  return(location); }
/*-------------------------------------------------------------\
/ Amachine_info[i][0] or Bmachine_info[i][0] =no. of machine  \
/ Amachine_info[i][1] or Bmachine_info[i][1] =machine actual age       \
/ Amachine_info[i][2] or Bmachine_info[i][2] =machine conditional age  \
/ Amachine_info[i][3] or Bmachine_info[i][3] =no. of breakdown        \
/ Amachine_info[i][4] or Bmachine_info[i][4] =no. of defects          \
/ Amachine_info[i][5] or Bmachine_info[i][5] =salvage value of the machine \
/--------------------------------------------------------------------*/
machining()
{  float b,a=0;
  int i,j,sold=0, range=0;
  float repair_time=0;
        float A_last_period_machine_value=0;  /* salvage value of machine in last period */
        float B_last_period_machine_value=0;
        int A_machine_defect=0,    B_machine_defect=0;
        A_machine_overtime=0;      B_machine_overtime=0;
        A_T_machine_cost=0;        B_T_machine_cost=0;
        A_T_machine_deprec_cost=0; B_T_machine_deprec_cost=0;
        A_last_period_machine_value=0;  B_last_period_machine_value=0;
```

```
        A_breakdown=0;      B_breakdown=0;
        A_repair_time=0;    B_repair_time=0;
   for(i=0;i<10;i++)
   {  Amachine_info[i][5]=A_machine_price*pow((1-A_machine_deprec_rate),Amachine_info[i][1]);
        if(Amachine_info[i][1]!=0)
        {  A_last_period_machine_value=A_machine_price*pow((1-
A_machine_deprec_rate),Amachine_info[i][1]-1);
                        A_T_machine_deprec_cost+=(A_last_period_machine_value-
Amachine_info[i][5])*Amachine_info[i][0];
        }
        Bmachine_info[i][5]=B_machine_price*pow((1-
B_machine_deprec_rate),Amachine_info[i][1]);
        if(Bmachine_info[i][1]!=0)
        {  B_last_period_machine_value=B_machine_price*pow((1-
B_machine_deprec_rate),Bmachine_info[i][1]-1);
                        B_T_machine_deprec_cost+=(B_last_period_machine_value-
Bmachine_info[i][5])*Bmachine_info[i][0];
        }        /* calculate the depreciation value of the machine */
        Amachine_info[i][4]=pow(A_quality_factor,Amachine_info[i][1]);
        Bmachine_info[i][4]=pow(B_quality_factor,Bmachine_info[i][1]);
                /* calculate the no. of defects found per lot */
            Amachine_info[i][3]=floor(pow(A_breakdn_factor,Amachine_info[i][2])-1);
            Bmachine_info[i][3]=floor(pow(B_breakdn_factor,Bmachine_info[i][2])-1);
   }          /* calculate the no. of breakdown */
   for(i=0;i<10;i++){
   if(Amachine_info[i][1]==10) sold+=Amachine_info[i][0];
   }
   A_total_machine-=sold;
   do
   {  A_sold=getin(15,47,'i');    /* get the no. of machine A to be sold out */
   }while(A_sold>A_total_machine);
   A_total_machine-=A_sold;      /* update no. of A machine after selling */
   sold+=A_sold;
   while(sold>0)  /* sell the machine A and raise the capital */
   {  i=old('a');
        if(Amachine_info[i][0]>=sold)
        {  Amachine_info[i][0]-=sold;  /* update the exist machine A no. */
                        cash+=sold*Amachine_info[i][5];  /* update the capital after selling */
          sold=0;      }
        else
        {  sold=sold-Amachine_info[i][0];
                        cash+=Amachine_info[i][0]*Amachine_info[i][5];
          for(j=0;j<5;j++)
          Amachine_info[i][j]=0; } }  /* clear the machine information */
   for(i=0;i<10;i++){
   if(Bmachine_info[i][1]==10) sold+=Bmachine_info[i][0];
   }
   B_total_machine-=sold;
   do
   {  B_sold=getin(16,47,'i');    /* get the no. of machine B to be sold out */
   }while(B_sold>B_total_machine);
   B_total_machine-=B_sold;   /* update the no. of B machine after selling */
   sold+=B_sold;
   while(sold>0)  /* sell the machine B and raise the capital */
   {  i=old('b');
        if(Bmachine_info[i][0]>=sold)
        {  Bmachine_info[i][0]-=sold;  /* update the exist machine B no. */
                        cash+=sold*Bmachine_info[i][5];  /* update the capital after selling */
```

```
            sold=0;      }
          else
          {  sold=sold-Bmachine_info[i][0];
                         cash+=Bmachine_info[i][0]*Bmachine_info[i][5];
            for(j=0;j<5;j++)
                Bmachine_info[i][j]=0; } } /* clear the machine information */
   A_maintenance=getin(17,36,'i'); /* get the maintenance frequency */
   B_maintenance=getin(18,36,'i'); /* get the maintenance frequency */
   A_T_mainten_cost=A_maintenance*A_mainten_cost*A_total_machine;
   B_T_mainten_cost=B_maintenance*B_mainten_cost*B_total_machine;
   cash=cash-A_T_mainten_cost-B_T_mainten_cost;
   A_buy=getin(19,46,'i');
   B_buy=getin(20,46,'i');
   cash=cash-A_buy*A_machine_price-B_buy*B_machine_price;
   for(i=0;i<10;i++)
   {  A_T_machine_cost+=Amachine_info[i][5]*Amachine_info[i][0]; /* value of exist A machine */
          B_T_machine_cost+=Bmachine_info[i][5]*Bmachine_info[i][0]; } /* value of exist B
machine */
   A_T_machine_cost+=A_buy*A_machine_price;    /* capital on new A machine */
          B_T_machine_cost+=B_buy*B_machine_price;      /* capital on new B machine */
   if(A_total_machine>0)
   {  range=1+A_max_repair_time-A_min_repair_time;
                for(i=0;i<10;i++)
          {  A_breakdown+=Amachine_info[i][3];  /* total no. of breakdown for A */
                         repair_time=(random(range)+A_min_repair_time); /* random number in
hour */

          A_repair_time+=(Amachine_info[i][0]/A_total_machine)*repair_time*Amachine_info[i][3];
                }
                A_T_repair_cost=A_breakdown*A_repair_cost; /* total repair cost */
                cash-=A_T_repair_cost;
          A_total_machine_time=(A_machine_time*A_target)/A_total_machine;   }
   if(B_total_machine>0)
   {  range=B_max_repair_time-B_min_repair_time;
                for(i=0;i<10;i++)
          {  B_breakdown+=Bmachine_info[i][3];
                         repair_time=(random(range)+B_min_repair_time); /* random number in
hour */

          B_repair_time+=(Bmachine_info[i][0]/B_total_machine)*repair_time*Bmachine_info[i][3];
                }
                B_T_repair_cost=B_breakdown*B_repair_cost;  /* total repair cost */
                cash-=B_T_repair_cost;
          B_total_machine_time=(B_machine_time*B_target)/B_total_machine;   }
   if (A_total_machine_time>hour_per_period)
          A_machine_overtime=A_total_machine_time-hour_per_period;
   if (B_total_machine_time>hour_per_period)
          B_machine_overtime=B_total_machine_time-hour_per_period;
             /* update the capital after overtime work */
   cash-=(A_machine_overtime+B_machine_overtime)*overtime_pay;
   A_machine_regular_hour=A_total_machine_time-A_machine_overtime;
   B_machine_regular_hour=B_total_machine_time-B_machine_overtime;
   for(i=0;i<10;i++)     /* total defects found for A & B per lot size */
   {  A_machine_defect+=Amachine_info[i][4];
          B_machine_defect+=Bmachine_info[i][4];
   }       /* count the total product cost in machining */
   A_T_product_cost+=A_machine_overhead*A_total_machine;
          a=factory_overhead+rent_rate;
```

```
        a=a*A_target/(A_target+B_target);
        A_T_product_cost+=a;
  A_T_product_cost+=A_T_repair_cost;
  A_T_product_cost+=A_T_mainten_cost;
  A_T_product_cost+=A_machine_overtime*overtime_pay;
  A_T_product_cost+=A_T_machine_deprec_cost;
  B_T_product_cost+=B_machine_overhead*B_total_machine;
        b=factory_overhead+rent_rate;
        b=b*B_target/(A_target+B_target);
        B_T_product_cost+=b;
  B_T_product_cost+=B_T_repair_cost;
  B_T_product_cost+=B_T_mainten_cost;
  B_T_product_cost+=B_machine_overtime*overtime_pay;
  B_T_product_cost+=B_T_machine_deprec_cost; }
/*------------------------------------------------*/
assembly()
{  float time_new, time_skill,temp;
  float B_overtime_cost, A_overtime_cost=0;
  float A_turnover_index=0, B_turnover_index=0;
        A_T_hire_cost=0;
        B_T_hire_cost=0;
        A_worker_overtime=0;
        B_worker_overtime=0;
        do
        {   A_fire=getin(3,58,'i');        /* no. of A workers to be fired */
        }while(A_fire>A_worker);            /* if fire more than current no. of A worker */
        A_hire=getin(4,42,'i');        /* no. of A workers to be hired */
        A_salary=getin(5,36,'f');      /* salary of each A worker */
  A_change=getin(6,42,'i');       /* no. of A worker changed to B */
        do
        {   B_fire=getin(8,58,'i');        /* no. of B workers to be fired */
        }while(B_fire>B_worker);            /* if fire more than current no. of B worker */
        B_hire=getin(9,42,'i');        /* no. of B workers to be hired */
        B_salary=getin(10,36,'f');     /* salary of each B worker */
  B_change=getin(11,42,'i');      /* no. of B worker changed to A */
  if (A_hire>0)
                A_T_hire_cost=hiring_overhead+A_hire*hire_cost;  /* total hiring cost */
        if (B_hire>0)
                B_T_hire_cost=hiring_overhead+B_hire*hire_cost;  /* total hiring cost */
  A_T_fire_cost=firing_cost*A_fire;          /* total firing cost */
  B_T_fire_cost=firing_cost*B_fire;          /* total firing cost */
  cash-=(A_T_hire_cost+A_T_fire_cost);
  cash-=(B_T_hire_cost+B_T_fire_cost);
  A_new_worker+=B_change;  /* total current new worker=hire from last period+come from B line*/
  B_new_worker+=A_change;  /* total current new worker=hire from last period+come from A
line*/
        /* total wages for B workers */
  temp=A_salary/average_wage;
        if (temp>1)
                A_leave=ceil(A_worker*0.005);  /* round up the no. */
  else
                A_leave=ceil(A_worker*(1-temp));       /* round up the no. */
  A_exist_worker=A_worker+A_new_worker-A_fire-A_leave-A_change;   /* exist A worker to work
*/
  if (A_exist_worker<0)
        A_exist_worker=0;         /* when salary is too low, all worker leave */
  A_T_worker_wages=A_exist_worker*A_salary;        /* total wages for A workers */
  if(A_exist_worker>0)
```

```
{ A_total_worker_time=A_target*A_assemble_time/((A_exist_worker-
A_new_worker)+A_learn_curve*A_new_worker);
			A_turnover_index=(A_new_worker+A_fire+A_leave)/A_exist_worker;
	}
  temp=B_salary/average_wage;
	if (temp>1)
			B_leave=ceil(B_worker*0.005);  /* round up the no. */
  else
			B_leave=ceil(B_worker*(1-temp));  /* round up the no. */
  B_exist_worker=B_worker+B_new_worker-B_fire-B_leave-B_change;   /* exist B worker to work
*/
  if (B_exist_worker<0)
	B_exist_worker=0;		/* when salary is too low, all worker leave */
  B_T_worker_wages=B_exist_worker*B_salary;
  if(B_exist_worker>0)
  {  B_total_worker_time=B_target*B_assemble_time/((B_exist_worker-
B_new_worker)+B_learn_curve*B_new_worker);
	B_turnover_index=(B_new_worker+B_fire+B_leave)/B_exist_worker; }
  if(A_total_worker_time>hour_per_period)
			A_worker_overtime=A_total_worker_time-hour_per_period;
  if(B_total_worker_time>hour_per_period)
			B_worker_overtime=B_total_worker_time-hour_per_period;
  A_worker_regular_hour=A_total_worker_time-A_worker_overtime;
  B_worker_regular_hour=B_total_worker_time-B_worker_overtime;
  A_overtime_cost=(A_worker_overtime/hour_per_period)*A_salary*A_exist_worker;
		/* overtime payment for the worker A */
  cash-=A_overtime_cost;		/* overtime payment for the worker B */
  B_overtime_cost=(B_worker_overtime/hour_per_period)*B_salary*B_exist_worker;
  cash-=B_overtime_cost;  /* count the total product cost in assembly */
  A_assemble_defect=A_turnover_index*A_turnover_quality_factor;
	B_assemble_defect=B_turnover_index*B_turnover_quality_factor;
  A_T_product_cost+=A_overtime_cost;
  A_T_product_cost+=A_T_worker_wages;
  B_T_product_cost+=B_overtime_cost;
  B_T_product_cost+=B_T_worker_wages; }

/*-----------------------------------*/
inspect()
{  float A1,A2, B1,B2=0;
  float A_percent_defect,B_percent_defect,total_defect=0;
  long int A_can_sold=0, B_can_sold=0;
  A_can_sold=A_target+A_stock_product-A_sell_market; /* no. of product exist */
  B_can_sold=B_target+B_stock_product-B_sell_market; /* no. of product exist */
  A_defect_allow=getin(13,41,'i');	/* defects allowed */
  do
  {  A_sample=getin(14,51,'i');	/* samples selected for testing */
	}while(A_sample<A_defect_allow);
  B_defect_allow=getin(15,41,'i');	/* defects allowed */
  do
  {  B_sample=getin(16,51,'i');	/* samples selected for testing */
	}while(B_sample<B_defect_allow);
  A_sell_price=getin(18,30,'f'); /* selling price */
  B_sell_price=getin(19,30,'f');
  do
  {  A_sold_product=getin(20,39,'l'); /* no. of product A to be bought */
  }while(A_sold_product>A_can_sold);
  do
  {  B_sold_product=getin(21,39,'l');  /* no. of product B to be bought */
```

```
   }while(B_sold_product>B_can_sold);

total_defect=(A_stock_material_level+A_assemble_defect+A_machine_defect)*A_target/A_lot_size;
   A_percent_defect=total_defect*100/A_lot_size;
   A_ave_inspect_no=ceil(A_sample+(1-pa(A_defect_allow,A_percent_defect))*(A_lot_size-
A_sample));  /* per lot */
        A_T_inspect_cost=A_ave_inspect_no*(A_target/A_lot_size)*A_inspect_cost;
   cash-=A_T_inspect_cost;
             /* average outgoing quality */
        A_AOQ=pa(A_defect_allow,A_percent_defect)*(A_lot_size-
A_sample)*A_percent_defect/100;
           /* overall AOQ including the product in stock */
   A_T_AOQ=(A_AOQ*A_target+A_stock_product*A_T_AOQ)/(A_target+A_stock_product);
   NA_stock_product=A_stock_product+A_target-A_sold_product-A_sell_market;   /* product
remained */
   if(NA_stock_product>0)
        {
        if(NA_stock_product>A_stock_product)

        A_product_level=((A_product_level*A_stock_product)+((A_defect_allow/A_sample*1000)*
(NA_stock_product-A_stock_product)))/NA_stock_product;
        else
                  A_product_level=A_defect_allow/A_sample*1000;
        }
        else
        A_product_level=0;

A_product_carry_cost=((NA_stock_product)+((A_sold_product+A_sell_market)*0.5))*A_stock_carry
_cost;   /* for this period */

total_defect=(B_stock_material_level+B_assemble_defect+B_machine_defect)*B_target/B_lot_size;
   B_percent_defect=total_defect*100/B_lot_size;
   B_ave_inspect_no=ceil(B_sample+(1-pa(B_defect_allow,B_percent_defect))*(B_lot_size-
B_sample)); /* per lot */
        B_T_inspect_cost=B_ave_inspect_no*(B_target/B_lot_size)*B_inspect_cost;
   cash-=B_T_inspect_cost;
           /* average outgoing quality */
        B_AOQ=pa(B_defect_allow,B_percent_defect)*(B_lot_size-
B_sample)*B_percent_defect/100;
           /* overall AOQ including the product in stock */
   B_T_AOQ=(B_AOQ*B_target+B_stock_product*B_T_AOQ)/(B_target+B_stock_product);
   NB_stock_product=B_stock_product+B_target-B_sold_product-B_sell_market; /* product remained
*/
   if(NB_stock_product>0)
        {
        if(NB_stock_product>B_stock_product)

        B_product_level=((B_product_level*B_stock_product)+((B_defect_allow/B_sample*1000)*(
NB_stock_product-B_stock_product)))/NB_stock_product;
        else
                  B_product_level=B_defect_allow/B_sample*1000;
        }
        else
        B_product_level=0;

B_product_carry_cost=((NB_stock_product)+((B_sold_product+B_sell_market)*0.5))*B_stock_carry
_cost;   /* for this period */
   cash-=B_product_carry_cost;
```

```
A_T_product_price=A_sold_product*A_sell_price; /* cash from selling product */
        B_T_product_price=B_sold_product*B_sell_price; /* cash from selling product */
cash+=(A_T_product_price+B_T_product_price);
        /* count the total product cost in inpsection */
A_T_product_cost+=A_T_inspect_cost;
B_T_product_cost+=B_T_inspect_cost;
/* average product unit cost in stock after at the end of period */
            A1=A_stock_ave_product_cost*A_stock_product;
            A2=A_T_product_cost;
A_stock_ave_product_cost=(A1+A2)/(A_stock_product+A_target);
            B1=B_stock_ave_product_cost*B_stock_product;
            B2=B_T_product_cost;
B_stock_ave_product_cost=(B1+B2)/(B_stock_product+B_target);
/* return from selling to marketing */
A_sell_market_cost=A_sell_market*A_stock_ave_product_cost;
B_sell_market_cost=B_sell_market*B_stock_ave_product_cost;
cash+=A_sell_market_cost+B_sell_market_cost;
if (cash<0)
{  overdraft+=cash*(-1);
        cash=0; } }

/*-------------------------------------------------------------\
/ update the environment for the next period                    \
/ update - conditional age of machines    - no. of worker after hiring \
/      - no. of machine after purchasing - period counter        \
/-------------------------------------------------------------*/
update_env()
{  int times=1, i=0;  float effect=0;
   if(A_maintenance>0)
   {  do
         {  effect+=pow(1-A_mainten_effect,times)/A_maintenance;
            times++;
         }while(times<A_maintenance);  }
   else
         effect=1;
   for(i=0;i<10;i++)
   {  Amachine_info[i][2]+=effect; /* update the conditional age for next period */
         Amachine_info[i][1]++;  }   /* update the actual age */
   if(B_maintenance>0)
   {  times=1;
         do
         {  effect+=pow(1-B_mainten_effect,times)/B_maintenance;
            times++;
         }while(times<B_maintenance);  }
   else
         effect=1;
   for(i=0;i<10;i++)
   {  Bmachine_info[i][2]+=effect; /* update the conditional age for next period */
         Bmachine_info[i][1]++; }    /* update the actual age */
         if (A_buy>0)
         for(i=0;i<10;i++)
                        if(Amachine_info[i][0]==0)
           {  Amachine_info[i][0]=A_buy;  /* add the no. of machine bought */
                 Amachine_info[i][1]=0;      /* all new machine are 1-year old */
                 break;          }
         if (B_buy>0)
         for(i=0;i<10;i++)
                        if(Bmachine_info[i][0]==0)
```

```
            {  Bmachine_info[i][0]=B_buy;  /* add the no. of machine bought */
                                  Bmachine_info[i][1]=1;      /* all new machine are 1-year old */
                 break;        }
          A_worker=A_exist_worker;  B_worker=B_exist_worker;
   A_new_worker+=A_hire;  /* update the new worker no. for the next period */
   B_new_worker+=B_hire;
   A_stock_material=NA_material_stock;
   B_stock_material=NB_material_stock;
   A_stock_material_level=NA_stock_material_level;
   B_stock_material_level=NB_stock_material_level;
   A_stock_ave_raw_cost=NA_stock_ave_raw_cost;
   B_stock_ave_raw_cost=NB_stock_ave_raw_cost;
   A_stock_product=NA_stock_product;
   B_stock_product=NB_stock_product;
}

writehis()
{
   FILE *hisout;
   int i;
   char filename[11];

  clrscr();
  printf("Input the new history file name : ");
  scanf("%s",filename);
  strcat(filename,".txt");
  if((hisout=fopen(filename,"w"))==NULL)
    printf("<<ERROR : File open error !>>\n");
  else
    {
          fprintf(hisout,"loan %f\n",loan);
          fprintf(hisout,"overdraft %f\n",overdraft);
          fprintf(hisout,"A_new_worker %d\n",A_new_worker);
          fprintf(hisout,"B_new_worker %d\n",B_new_worker);
          fprintf(hisout,"capital %f\n",cash);
          fprintf(hisout,"A_product_level_stock %f\n",A_product_level);  /* stock product A quality
level */
          fprintf(hisout,"A_material_in_stock %ld\n",NA_material_stock);  /* stock quantity of part A
*/
          fprintf(hisout,"A_stock_material_quality_level %f\n",NA_stock_material_level);  /* material
quality level */
          fprintf(hisout,"A_stock_product %ld\n",NA_stock_product);  /* no. of product A in stock */
          fprintf(hisout,"A_stock_ave_raw_cost %f\n",NA_stock_ave_raw_cost);  /* stock material
average cost */
          fprintf(hisout,"A_stock_ave_product_cost %f\n",A_stock_ave_product_cost);  /* average
product A cost */
          for(i=0;i<10;i++)
             {
                   fprintf(hisout,"A_%d_type_no_of_machine %f\n",i,Amachine_info[i][0]);
                   fprintf(hisout,"A_%d_type_actual_age %f\n",i,Amachine_info[i][1]);
                   fprintf(hisout,"A_%d_type_conditional_age %f\n",i,Amachine_info[i][2]);
             }
          fprintf(hisout,"A_worker %ld\n",A_worker);        /* no. of worker for part A */

          fprintf(hisout,"B_product_level_stock %f\n",B_product_level);  /* stock product B quality
level */
          fprintf(hisout,"B_material_in_stock %ld\n",NB_material_stock);  /* stock quantity of part B
*/
```

```
        fprintf(hisout,"B_stock_material_quality_level %f\n",NB_stock_material_level); /* material
quality level */
        fprintf(hisout,"B_stock_product %ld\n",NB_stock_product); /* no. of product B in stock */
        fprintf(hisout,"B_stock_ave_raw_cost %f\n",NB_stock_ave_raw_cost); /* stock material
average cost */
        fprintf(hisout,"B_stock_ave_product_cost %f\n",B_stock_ave_product_cost); /* average
product B cost */

        for(i=0;i<10;i++)
          {
                fprintf(hisout,"B_%d_type_no_of_machine %f\n",i,Bmachine_info[i][0]);
                fprintf(hisout,"B_%d_type_actual_age %f\n",i,Bmachine_info[i][1]);
                fprintf(hisout,"B_%d_type_conditional_age %f\n",i,Bmachine_info[i][2]);
                        }
        fprintf(hisout,"B_worker %ld\n",B_worker);
  }
fclose(hisout);
}

readhis()
{
  FILE *hisin;
  int i,j;
  char filename[11];
  char string[80];
        float temp;

 clrscr();
 printf("Input the history file name : ");
 scanf("%s",filename);
 strcat(filename,".txt");
 if((hisin=fopen(filename,"r"))==NULL)
   printf("<<ERROR : File open error !>>\n");
 else
   {
        fscanf(hisin,"%s %f",string,&loan);
        fscanf(hisin,"%s %f",string,&overdraft);
        fscanf(hisin,"%s %d",string,&A_new_worker);
        fscanf(hisin,"%s %d",string,&B_new_worker);
        fscanf(hisin,"%s %f",string,&capital);
        fscanf(hisin,"%s %f",string,&A_product_level); /* stock product A quality level */
        fscanf(hisin,"%s %ld",string,&A_stock_material); /* stock quantity of part A */
        fscanf(hisin,"%s %f",string,&A_stock_material_level); /* material quality level */
        fscanf(hisin,"%s %ld",string,&A_stock_product); /* no. of product A in stock */
        fscanf(hisin,"%s %f",string,&A_stock_ave_raw_cost); /* stock material average cost */
        fscanf(hisin,"%s %f",string,&A_stock_ave_product_cost); /* average product A cost */
        for(i=0;i<10;i++)
          for(j=0;j<3;j++)          /* for machine A */
          {                       /* number of machine, */
                fscanf(hisin,"%s %f",string,&temp); /* actual age of machine */
                Amachine_info[i][j]=temp;   }    /* conditional age */
        fscanf(hisin,"%s %ld",string,&A_worker);      /* no. of worker for part A */
        fscanf(hisin,"%s %f",string,&B_product_level); /* stock product B quality level */
        fscanf(hisin,"%s %ld",string,&B_stock_material); /* stock quantity of part B */
        fscanf(hisin,"%s %f",string,&B_stock_material_level); /* material quality level */
        fscanf(hisin,"%s %ld",string,&B_stock_product); /* no. of product B in stock */
        fscanf(hisin,"%s %f",string,&B_stock_ave_raw_cost); /* stock material average cost */
        fscanf(hisin,"%s %f",string,&B_stock_ave_product_cost); /* average product B cost */
```

```
        for(i=0;i<10;i++)
          for(j=0;j<3;j++)              /* for machine B */
          {                           /* number of machine, */
                fscanf(hisin,"%s %f",string,&temp); /* actual age of machine */
                Bmachine_info[i][j]=temp;      }   /* conditional age */
        fscanf(hisin,"%s %ld",string,&B_worker);        /* no. of worker for part B */
  }
fclose(hisin);
}
```

The production game - program listing (part 2)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define E 2.718281828

extern float capital;            /* capital */
extern float cash;               /* cash on hand */
extern float loan_rate;          /* loan interest rate */
extern float overdraft_rate;     /* emergency over draft interest rate */
extern float pay_loan;           /* fund transfer from marketing */
extern float pay_overdraft;
extern float hour_per_period;    /* no. of working hours */
extern float firing_cost;        /* labour firing cost/person */
extern float hiring_overhead;    /* labour hiring cost/overhead */
extern float hire_cost;          /* labour hiring cost/person */
extern float average_wage;       /* average industrial wages */
extern float factory_overhead;   /* overhead cost of the factory */
extern float overtime_pay;       /* overtime pay per hour */
extern float rent_rate;          /* rent and rates  */
extern float A_product_level;        /* product A quality level in stock */
extern long int A_stock_material;    /* raw material A in stock */
extern float A_stock_material_level; /* material quality level in stock */
extern long int A_stock_product;     /* product A in stock */
extern float A_stock_carry_cost;     /* stock carrying cost/unit/period (A) */
extern float A_stock_ave_raw_cost;   /* average cost of raw material A in stock */
extern float A_stock_ave_product_cost; /* average cost of product A in stock */
extern float A_machine_time;         /* A machine time per product */
extern float Amachine_info[10][6];   /* Amachine_info[i][0]=no. of machine */
/* Amachine_info[i][1]=machine actual age  Amachine_info[i][2]=machine conditional age */
/* Amachine_info[i][3]=no. of breakdown   Amachine_info[i][4]=no. of defects */
/* Amachine_info[i][5]=depreciation cost */
extern float A_machine_price;        /* price for machine A */
extern float A_machine_deprec_rate;  /* machine A depreciation % */
extern float A_machine_overhead;     /* machine A overhead cost/unit */
extern float A_mainten_cost;         /* machine A maintenance cost */
extern float A_mainten_effect;       /* machine A maintenance effect */
extern float A_breakdn_factor;       /* machine A break down factor */
extern float A_quality_factor;       /* machine A quality factor */
extern float A_min_repair_time;      /* machine A min repairing time */
extern float A_max_repair_time;      /* machine A max repairing time */
extern float A_repair_cost;          /* machine A repairing cost per hour */
extern long int A_worker;            /* no. of worker for part A */
extern float A_assemble_time;        /* assembling time for part A */
extern float A_turnover_quality_factor; /* turnover quality factor for A */
extern float A_learn_curve;          /* learning curve for A workers */
extern int A_lot_size;               /* log size for product A */
extern float A_inspect_cost;         /* inspection cost/unit for A */
extern float B_product_level;        /* product B quality level in stock */
extern long int B_stock_material;    /* stock quantity of part B */
extern float B_stock_material_level; /* material quality level in stock */
extern long int B_stock_product;     /* product B in stock */
extern float B_stock_carry_cost;     /* stock carrying cost/unit/period (B) */
extern float B_stock_ave_raw_cost;   /* average cost of raw material B in stock */
extern float B_stock_ave_product_cost; /* average cost of product B in stock */
extern float B_machine_time;         /* machine time for product B */
```

```
extern float Bmachine_info[10][6]; /* Bmachine_info[i][0]=no. of machine */
/* Bmachine_info[i][1]=machine actual age  Bmachine_info[i][2]=machine conditional age */
/* Bmachine_info[i][3]=no. of breakdown   Bmachine_info[i][4]=no. of defects */
/* Bmachine_info[i][5]=depreciation cost */
extern float B_machine_price;       /* price for machine B */
extern float B_machine_deprec_rate; /* machine B depreciation % */
extern float B_machine_overhead;    /* machine B overhead cost/unit */
extern float B_mainten_cost;        /* machine B maintenance cost */
extern float B_mainten_effect;      /* machine B maintenace effect */
extern float B_breakdn_factor;      /* machine B break down factor */
extern float B_quality_factor;      /* machine B quality factor */
extern float B_min_repair_time;     /* machine B min repairing time */
extern float B_max_repair_time;     /* machine B max repairing time */
extern float B_repair_cost;         /* machine B repairing cost per hour */
extern long int B_worker;           /* no. of worker for part B */
extern float B_assemble_time;       /* assembling time for part B */
extern float B_turnover_quality_factor; /* turnover quality factor for B */
extern float B_learn_curve;         /* learning curve for B workers */
extern int B_lot_size;              /* log size for product B */
extern float B_inspect_cost;        /* inspection cost/unit for B */
extern int period;    /*--------- variables in report -----------*/
extern float premise;               /* from parameter file */
extern float A_T_machine_cost;      /* exist all A machines cost */
extern float A_T_machine_deprec_cost; /* accumulated depreciation */
extern float B_T_machine_cost;      /* exist all B machines cost */
extern float B_T_machine_deprec_cost; /* accumulated depreciation */
extern float overdraft;             /* over draft */
extern float loan;                  /* approved loan */
extern float A_ave_cost;            /* average raw material A cost (input) */
extern float B_ave_cost;            /* average raw material B cost (input) */
extern float A_T_product_price;     /* product A total selling price */
extern float B_T_product_price;     /* product B total selling price */
extern float A_T_product_cost;  /* Total price for direct sell to marketing */
extern float B_T_product_cost;  /* Total price for direct sell to marketing */
extern long int A_sold_product;     /* no. of product A to be bought */
extern long int B_sold_product;     /* no. of product B to be bought */
extern long int A_order_material;       /* purchase of A raw material no. */
extern long int B_order_material;       /* purchase of B raw material no. */
extern float A_T_worker_wages;      /* total wages for worker A */
extern float B_T_worker_wages;      /* total wages for worker B */
extern float A_T_inspect_cost;      /* total inspection cost for product A */
extern float B_T_inspect_cost;      /* total inspection cost for product B */
extern int A_sold;                  /* no. of A machine sold out */
extern int B_sold;                  /* no. of B machine sold out */
extern int A_buy;                   /* no. of machine A are bought */
extern int B_buy;                   /* no. of machine B are bought */
extern float A_T_mainten_cost;      /* total maintenance cost for A machines */
extern float B_T_mainten_cost;      /* total maintenance cost for B machines */
extern float A_T_repair_cost;       /* total reparing cost for A machines */
extern float B_T_repair_cost;       /* total reparing cost for B machines */
extern float A_material_carry_cost; /* carrying cost for raw material A */
extern float B_material_carry_cost; /* carrying cost for raw material B */
extern float A_product_carry_cost;  /* carrying cost for product A */
extern float B_product_carry_cost;  /* carrying cost for product B */
extern float A_stock_product_defect; /*  defect of product in stock */
extern float B_stock_product_defect; /* defect of product in stock */
extern float A_T_fire_cost;         /* total firing cost for production line A */
extern float B_T_fire_cost;         /* total firing cost for production line B */
```

```
extern float A_T_hire_cost;        /* total hiring cost for production line A */
extern float B_T_hire_cost;        /* total hiring cost for production line B */
extern int A_repair_time;          /* machine A repair time */
extern int B_repair_time;          /* machine B repair time */
extern int A_exist_worker;         /* exist A worker for the current period */
extern int B_exist_worker;         /* exist B worker for the current period */
extern int A_new_worker;           /* exist new worker for the current period */
extern int B_new_worker;           /* exist new worker for the current period */
extern int A_breakdown;            /* total no. of breakdown for machine A */
extern int B_breakdown;            /* total no. of breakdown for machine B */
extern float A_total_machine_time;  /* total working hour for machine A */
extern float B_total_machine_time;  /* total working hour for machine B */
extern float A_machine_overtime;    /* overtime working hour for machine A */
extern float B_machine_overtime;    /* overtime working hour for machine B */
extern float A_machine_regular_hour; /* regular working hour for machine A */
extern float B_machine_regular_hour; /* regular working hour for machine B */
extern float A_total_worker_time;   /* total working hour for labour in line A */
extern float B_total_worker_time;   /* total working hour for labour in line B */
extern float A_worker_overtime;     /* overtime working hour for worker A */
extern float B_worker_overtime;     /* overtime working hour for worker B */
extern float A_worker_regular_hour; /* regular working hour for worker A */
extern float B_worker_regular_hour; /* regular working hour for worker B */
extern int A_ave_inspect_no;       /* average inspection no. of product A */
extern int B_ave_inspect_no;       /* average inspection no. of product B */
extern int A_sample;               /* sample no. of product A */
extern int B_sample;               /* sample no. of product B */
extern int A_defect_allow;         /* no. of defect allowed per lot of A */
extern int B_defect_allow;         /* no. of defect allowed per lot of B */
extern float A_AOQ;           /* product A average output quality (new produced) */
extern float B_AOQ;           /* product B average output quality (new produced) */
extern float A_T_AOQ;  /* overall AOQ of product A including the product in stock */
extern float B_T_AOQ;  /* overall AOQ of product B including the product in stock */
extern float A_sell_market_cost;    /* cash return from direct sell to marketing */
extern float B_sell_market_cost;    /* cash return from direct sell to marketing */
extern float A_sell_price;         /* product A selling price */
extern float B_sell_price;         /* product B selling price */
extern float A_salary;             /* worker A salary */
extern float B_salary;             /* worker B salary */
extern float NA_stock_material_level; /* next period material quality level in stock */
extern float NB_stock_material_level; /* next period material quality level in stock */
/*---------- variables in programming --------------*/
extern long int NB_stock_product;   /* product B in next stock */
extern long int NA_stock_product;   /* product A in next stock */
extern long int A_target;          /* product A production target */
extern long int B_target;          /* product B production target */
extern long int NA_material_stock; /* A raw material in stock for next period */
extern long int NB_material_stock; /* B raw material in stock for next period */
extern int A_assemble_defect;      /* total defects from in assembly A / lot size */
extern int B_assemble_defect;      /* total defects from in assembly B / lot size */
extern int A_machine_defect;       /* total defects from machining / lot size */
extern int B_machine_defect;       /* total defects from machining / lot size */
extern float NA_stock_ave_raw_cost; /* next period average material cost in stock */
extern float NB_stock_ave_raw_cost; /* next period average material cost in stock */
extern long int A_sell_market;     /* product A direct sell to marketing */
extern long int B_sell_market;     /* product B direct sell to marketing */
extern int A_maintenance;          /* maintenance frequency for A machine */
extern int B_maintenance;          /* maintenance frequency for B machine */
extern int A_hire;              /* no. of A workers are hired */
```

```
fprintf(out,"\nInspection\n");
fprintf(out,"      Lot size : %d\n",A_lot_size);
fprintf(out,"      Sample size : %d\n",A_sample);
fprintf(out,"      No. of defectives allowed : %d\n",A_defect_allow);
fprintf(out,"      Average no. of inspection/lot : %d\n",A_ave_inspect_no);
fprintf(out,"\nFinished product\n");
    fprintf(out,"      No. of items produced : %ld\n",A_target);
fprintf(out,"      Average quality level : %.2f\n",A_T_AOQ);
fprintf(out,"      No. of items transferred to Marketing : %ld\n",A_sell_market);
fprintf(out,"      Production cost/unit : %.2f\n",A_T_product_cost/A_target);
fprintf(out,"      Existing stock : %d\n",NA_stock_product);
/*--------------- Production summary (product B) ---------- */
fprintf(out,"\n\n              Production Summary (Product B)\n");
fprintf(out,"\nRaw material\n");
fprintf(out,"      Part B used : %ld\n",B_target);
fprintf(out,"      Average quality level : %.2f\n",B_stock_material_level);
fprintf(out,"      Stock carryover : %ld\n",NB_material_stock);
fprintf(out,"      Average quality level : %.2f\n",NB_stock_material_level);
fprintf(out,"\nCapacity\n");
fprintf(out,"      Manpower\n");
fprintf(out,"          Newly hired : %d\n",B_new_worker);
fprintf(out,"          Left or transferred : %d\n",B_change+B_fire+B_leave);
fprintf(out,"      Total no. of labours : %d\n",B_exist_worker);
fprintf(out,"\n     Machine\n");
fprintf(out,"          Newly purchased : %d\n",B_buy);
fprintf(out,"          Sold : %d\n",B_sold);
fprintf(out,"      Total no. of machine : %d\n",B_total_machine);
fprintf(out,"\nAssembling\n");
fprintf(out,"      Regular man-hours used : %.2f\n",B_worker_regular_hour);
fprintf(out,"      Overtime man-hours used : %.2f\n",B_worker_overtime);
fprintf(out,"\nMachining\n");
fprintf(out,"      Regular machine-hours used : %.2f\n",B_machine_regular_hour);
fprintf(out,"      Overtime machine-hours used : %.2f\n",B_machine_overtime);
fprintf(out,"\nMaintenance\n");
fprintf(out,"      Number of maintenance/period : %d\n",B_maintenance);
fprintf(out,"      Number of breakdowns : %d\n",B_breakdown);
fprintf(out,"      Time lost in production : %d\n",B_repair_time);
fprintf(out,"\nInspection\n");
fprintf(out,"      Lot size : %d\n",B_lot_size);
fprintf(out,"      Sample size : %d\n",B_sample);
fprintf(out,"      No. of defectives allowed : %d\n",B_defect_allow);
fprintf(out,"      Average no. of inspection/lot : %d\n",B_ave_inspect_no);
fprintf(out,"\nFinished product\n");
    fprintf(out,"      No. of items produced : %ld\n",B_target);
fprintf(out,"      Average quality level : %.2f\n",B_T_AOQ);
fprintf(out,"      No. of items transferred to Marketing : %ld\n",B_sell_market);
fprintf(out,"      Production cost/unit : %.2f\n",B_T_product_cost/B_target);
fprintf(out,"      Existing stock : %d\n",NB_stock_product);
/*------ Production's Trading & Profit & Loss ---------*/
fprintf(out,"\n-----------------------------------------------------------------");
fprintf(out,"\nProduction's Trading & Profit & Loss A/C for the Year Ended Period
%d\n",period);
out1=A_sell_price*A_sold_product;
out2=A_sell_market_cost;
fprintf(out,"\nProduct A - Direct Sales at selling price            %10.0f\n",out1);
fprintf(out,"        Transfer to Branch(Marketing) at cost price %10.0f\n",out2);
fprintf(out,"                                            ----------");
fprintf(out,"   %10.0f\n",out1+out2);
```

```
out3=B_sell_price*B_sold_product;
out4=B_sell_market_cost;
fprintf(out,"\nProduct B - Direct Sales at selling price          %10.0f\n",out3);
fprintf(out,"          Transfer to Branch(Marketing) at cost price %10.0f\n",out4);
fprintf(out,"                                    ");
fprintf(out,"     ----------  %10.0f\n",out3+out4);
gain=out1+out2+out3+out4;
fprintf(out,"                          ");
fprintf(out,"                          ----------  %10.0f\n",gain);
com_data[9]=gain; /* for conbine report */
com_data[12]=out2+out4;
fprintf(out,"Less: Cost of Good Sold :\n\n");
out1=A_stock_material*A_stock_ave_raw_cost;
out2=A_order_material*A_ave_cost;
out3=NA_material_stock*NA_stock_ave_raw_cost;
out4=out1+out2-out3;
fprintf(out,"Part A for : Opening stock of Raw Material          %10.0f\n",out1);
fprintf(out,"Product A: + Purchase of Raw Material              %10.0f\n",out2);
fprintf(out,"          - Closing Stock of Raw Material          %10.0f\n",out3);
fprintf(out,"                                         ----------\n");
fprintf(out,"Raw Material of Part A Consumed                              %10.0f\n\n",out4);
out5=B_stock_material*B_stock_ave_raw_cost;
out6=B_order_material*B_ave_cost;
out7=NB_material_stock*NB_stock_ave_raw_cost;
out8=out5+out6-out7;
fprintf(out,"Part B for : Opening stock of Raw Material          %10.0f\n",out5);
fprintf(out,"Product B: + Purchase of Raw Material              %10.0f\n",out6);
fprintf(out,"          - Closing Stock of Raw Material          %10.0f\n",out7);
fprintf(out,"                                         ----------\n");
fprintf(out,"Raw Material of Part B Consumed                              %10.0f\n",out8);
fprintf(out,"                                                ----------\n");
com_data[10]=out1+out5; /* for combine report*/
com_data[11]=out2+out6; /* for conbine report*/
com_data[13]=out3+out7; /* for combine report*/
fprintf(out,"                                                %10.0f\n",out8+out4);
loss=out8+out4;
fprintf(out,"\n             Manufacturing Overhead\n");
out1=A_total_machine*A_machine_overhead;
out2=A_exist_worker*A_salary;
out3=A_T_inspect_cost;
fprintf(out,"\nProduct A - Machinery Overhead (per unit)          %10.0f\n",out1);
fprintf(out,"           Wages (indirect)                  %10.0f\n",out2);
fprintf(out,"           Inspection cost                   %10.0f\n",out3);
fprintf(out,"                                         ----------  %10.0f\n",out1+out2+out3);
loss+=out1+out2+out3;
out4=B_total_machine*B_machine_overhead;
out5=B_exist_worker*B_salary;
out6=B_T_inspect_cost;
fprintf(out,"Product B - Machinery Overhead (per unit)          %10.0f\n",out4);
fprintf(out,"           Wages (indirect)                  %10.0f\n",out5);
fprintf(out,"           Inspection cost                   %10.0f\n",out6);
fprintf(out,"                                    ");
fprintf(out,"      ----------  %10.0f\n",out4+out5+out6);
com_data[14]=out1+out4; /* for combine report*/
com_data[15]=out2+out5; /* for conbine report*/
com_data[16]=out3+out6; /* for conbine report*/
com_data[17]=com_data[14]+com_data[15]+com_data[16];/*combine report data*/
loss+=out4+out5+out6;
```

```
    fprintf(out,"                              ");
    fprintf(out,"                    ---------  %10.0f\n",out1+out2+out3+out4+out5+out6);
    fprintf(out,"        ");
    fprintf(out,"                                    ----------\n");
    fprintf(out,"                              ");
    fprintf(out,"                   %10.0f\n",gain-loss);
    com_data[18]=gain-loss; /* combine reprot data*/
    fprintf(out,"          Gross Profit of Product A & B\n\n");
    fprintf(out,"          Expenses\n");
    fprintf(out,"\nProduct A:    Depreciation of Machinery
%10.0f\n",A_T_machine_deprec_cost);
    com_data[19]=A_T_machine_deprec_cost+B_T_machine_deprec_cost;
    fprintf(out,"          Maintenance cost              %10.0f\n",A_T_mainten_cost);
    com_data[20]=A_T_mainten_cost+B_T_mainten_cost;
    fprintf(out,"          Repairing cost               %10.0f\n",A_T_repair_cost);
    com_data[21]=A_T_repair_cost+B_T_repair_cost;
    fprintf(out,"          Carrying cost of Raw Material      %10.0f\n",A_material_carry_cost);
    fprintf(out,"          Carrying cost of Finished Product   %10.0f\n",A_product_carry_cost);
    fprintf(out,"          Firing cost                  %10.0f\n",A_T_fire_cost);
    fprintf(out,"          Hiring cost                  %10.0f\n",A_T_hire_cost);
    out1=A_T_machine_deprec_cost+A_T_mainten_cost+A_T_repair_cost;
    out1+=A_material_carry_cost+A_product_carry_cost+A_T_fire_cost+A_T_hire_cost;
    fprintf(out,"                              ---------- %10.0f\n",out1);
    fprintf(out,"\nProduct B:    Depreciation of Machinery
%10.0f\n",B_T_machine_deprec_cost);
    fprintf(out,"          Maintenance cost              %10.0f\n",B_T_mainten_cost);
    fprintf(out,"          Repairing cost               %10.0f\n",B_T_repair_cost);
    fprintf(out,"          Carrying cost of Raw Material      %10.0f\n",B_material_carry_cost);
    fprintf(out,"          Carrying cost of Finished Product   %10.0f\n",B_product_carry_cost);
    fprintf(out,"          Firing cost                  %10.0f\n",B_T_fire_cost);
    com_data[26]=A_material_carry_cost+B_material_carry_cost;/* for combine report*/
    com_data[27]=A_product_carry_cost+B_product_carry_cost;/* for combine report*/
    com_data[28]=A_T_fire_cost+B_T_fire_cost; /* for combine report */
    fprintf(out,"          Hiring cost                  %10.0f\n",B_T_hire_cost);
    com_data[29]=A_T_hire_cost+B_T_hire_cost; /* for combine report */
    out2=B_T_machine_deprec_cost+B_T_mainten_cost+B_T_repair_cost;
    out2+=B_material_carry_cost+B_product_carry_cost+B_T_fire_cost+B_T_hire_cost;
    fprintf(out,"                 ");
    fprintf(out,"          ----------  %10.0f\n\n",out2);
    fprintf(out,"General Expenses (Overhead)              %10.0f\n",factory_overhead);
    com_data[23]=factory_overhead;
    fprintf(out,"Paid for Over Draft                  %10.0f\n",pay_overdraft);
    com_data[24]=pay_overdraft;
    fprintf(out,"Paid for Loan                        %10.0f\n",pay_loan);
    com_data[25]=pay_loan;
    out3=factory_overhead+loan*loan_rate;
    out3+=rent_rate+overdraft_rate*overdraft;
    com_data[30]=rent_rate;/* for combine report*/
    out4=out1+out2+out3;
    fprintf(out,"Rent & Rates                         %10.0f   %10.0f
%10.0f\n",rent_rate,out3,out4);
    fprintf(out,"                       ");
    fprintf(out," ----------  ----------  ----------\n");
    profit=gain-loss-out4;
    fprintf(out,"Net Profit                           ");
    fprintf(out,"          %10.0f\n",profit);
  /*---------- Production's Balance Sheet at peiod XX -------*/
    fprintf(out,"\n\n          Production's Balance Sheet As at Period %d\n",period);
```

```
fprintf(out,"\nFixed Assest :\n");
fprintf(out,"Premise :                          %10.0f\n",premise);
com_data[0]=premise;             /* for combined report */
fprintf(out,"Machinery A (Cost) :          %10.0f\n",A_T_machine_cost);
com_data[1]=A_T_machine_cost;       /* for combined report */
fprintf(out,"Less: Accumulated Depreciation : %10.0f\n",A_T_machine_deprec_cost);
com_data[2]=A_T_machine_deprec_cost;  /* for combined report */
outa=A_T_machine_cost-A_T_machine_deprec_cost;
fprintf(out,"                          ----------  %10.0f\n",outa);
fprintf(out,"\nMachinery B (Cost) :          %10.0f\n",B_T_machine_cost);
com_data[1]+=B_T_machine_cost;
fprintf(out,"Less: Accumulated Depreciation : %10.0f\n",B_T_machine_deprec_cost);
com_data[2]+=B_T_machine_deprec_cost;
outb=B_T_machine_cost-B_T_machine_deprec_cost;
fprintf(out,"                          ----------  %10.0f\n",outb);
fprintf(out,"                                      ----------\n");
fprintf(out,"                                      %10.0f\n",premise+outa+outb);
fprintf(out,"\nCurrent Assets :\n");
out1=NA_material_stock*NA_stock_ave_raw_cost;
out2=NB_material_stock*NB_stock_ave_raw_cost;
out3=NA_stock_product*A_stock_ave_product_cost;
out4=NB_stock_product*B_stock_ave_product_cost;
if (cash>0)
  out5=cash;     /* as cash on hand */
else
  out5=0;
fprintf(out,"Raw Material of Product A :     %10.0f\n",out1);
fprintf(out,"Raw Material of Product B :     %10.0f\n",out2);
com_data[3]=out1+out2;          /* for combined report */
fprintf(out,"Finished Product A :          %10.0f\n",out3);
fprintf(out,"Finished Product B :          %10.0f\n",out4);
com_data[4]=out3+out4;          /* for combined report */
fprintf(out,"Cash on Hand :               %10.0f\n",out5);
com_data[5]=out5;            /* for combined report */
out6=out1+out2+out3+out4+out5;
fprintf(out,"                          ----------\n");
fprintf(out,"                          %10.0f\n",out6);
fprintf(out,"Less: Current Liability\n");
fprintf(out,"Over Draft :                 %10.0f\n",overdraft);
com_data[6]=overdraft;        /* for combined report */
com_data[7]=loan;           /* for combined report */
fprintf(out,"Loan :                    %10.0f  %10.0f\n",loan,loan+overdraft);
fprintf(out,"                          ----------  ----------\n");
out7=outa+outb+premise+out6-loan-overdraft;
fprintf(out,"                                %10.0f\n",out7);
fprintf(out,"\nFinanced by\n");
fprintf(out,"Capital :                          %10.0f\n",out7-profit);
fprintf(out,"Net Profit :                        %10.0f\n",profit);
com_data[8]=profit;
com_data[31]=profit;          /* for combined report */
com_data[22]=com_data[19]+com_data[20]+com_data[21]+com_data[23]+com_data[24];

com_data[22]+=com_data[25]+com_data[26]+com_data[27]+com_data[28]+com_data[29]+com_data
[30];
  /*-------- start printing the combine file -----*/
strcat(filename,".d");        /* output file XXXXXX.d */
if ((out=fopen(filename,"w"))==NULL)
  printf("<<ERROR : Output file open error !>>\n");
```

```
    else
      for(i=0;i<32;i++)
          fprintf(out,"%f \n",com_data[i]); } }  /* write to combine files*/

/*-------------- input format ------------------*/
scr_decision()
{  printf("Input the value for pay for loan (loan=%7.2f):",loan);
   printf("\nApproved loan :");
   printf("\nProduction target for product A :");
   printf("\nProduction target for product B :");
   printf("\nQuantity of direct selling to marketing (product A):");
   printf("\nQuantity of direct selling to marketing (product B):\n");
}
scr_material()
{  printf("\nOrder for raw material A quantity:");
   printf("\n       Raw material A unit cost :");
   printf("\n    Raw material A quality level :");
   printf("\nOrder for raw material B quantity:");
   printf("\n       Raw material B unit cost :");
   printf("\n    Raw material B quality level :\n");
}
scr_machine()
{  printf("\nInput the number of machine A to be sold out:");
   printf("\nInput the number of machine B to be sold out:");
   printf("\nNo. of maintenance for Machine A :");
   printf("\nNo. of maintenance for Machine B :");
   printf("\nInput the number of machine A to be bought:");
   printf("\nInput the number of machine B to be bought:\n");
}
scr_labour()
{  printf("\nProduction line A -");
        printf("\n        Number of worker to be fired (worker=%ld) : ",A_worker);
        printf("\n        Number of worker to be hired : ");
   printf("\n        Salary for each worker :");
   printf("\n        Number of worker interchange :");
   printf("\nProduction line B :");
        printf("\n        Number of worker to be fired (worker=%ld) : ",B_worker);
        printf("\n        Number of worker to be hired : ");
   printf("\n        Salary for each worker :");
   printf("\n        Number of worker interchange :\n");
}
scr_inspect()
{   printf("\nProduct A - no. of defectives allowed :");
   printf("\n          - no. of samples selected for testing :");
   printf("\nProduct B - no. of defectives allowed :");
   printf("\n          - no. of samples selected for testing :\n");
}
scr_selling()
{  printf("\nSelling price of product A :");
   printf("\nSelling price of product B :");
        printf("\nNumber of product A direct sold out :");
        printf("\nNumber of product B direct sold out :\n");
}
```

Appendix XVII

Production Summary (Product A)

Raw material
- Part A used : 50000
- Average quality level : 1.00
- Stock carryover : 150000
- Average quality level : 1.67

Capacity
- Manpower
  - Newly hired : 0
  - Left or transferred : 4
- Total no. of labours : 46

- Machine
  - Newly purchased : 0
  - Sold : 0
- Total no. of machine : 31

Assembling
- Regular man-hours used : 108.70
- Overtime man-hours used : 0.00

Machining
- Regular machine-hours used : 180.00
- Overtime machine-hours used : 29.68

Maintenance
- Number of maintenance/period : 2
- Number of breakdowns : 33
- Time lost in production : 4

Inspection
- Lot size : 1000
- Sample size : 100
- No. of defectives allowed : 5
- Average no. of inspection/lot : 604

Finished product
- No. of items produced : 50000
- Average quality level : 19.82
- No. of items transferred to Marketing : 50000
- Production cost/unit : 31.15
- Existing stock : 0

Production Summary (Product B)

Raw material
- Part B used : 50000
- Average quality level : 1.00
- Stock carryover : 150000
- Average quality level : 1.00

Capacity
- Manpower
  - Newly hired : 0
  - Left or transferred : 4
- Total no. of labours : 46

- Machine
  - Newly purchased : 0
  - Sold : 0
- Total no. of machine : 31

Appendix XVII

Assembling
- Regular man-hours used : 108.70
- Overtime man-hours used : 0.00

Machining
- Regular machine-hours used : 180.00
- Overtime machine-hours used : 61.94

Maintenance
- Number of maintenance/period : 3
- Number of breakdowns : 33
- Time lost in production : 5

Inspection
- Lot size : 1000
- Sample size : 100
- No. of defectives allowed : 3
- Average no. of inspection/lot : 888

Finished product
- No. of items produced : 50000
- Average quality level : 5.61
- No. of items transferred to Marketing : 50000
- Production cost/unit : 48.34
- Existing stock : 0

Production's Trading & Profit & Loss A/C for the Year Ended Period 1

Product A - Direct Sales at selling price	0		
Transfer to Branch(Marketing) at cost price	1557546		
	----------	1557546	
Product B - Direct Sales at selling price	0		
Transfer to Branch(Marketing) at cost price	2417019		
	----------	2417019	
		----------	3974565
Less: Cost of Good Sold :			
Part A for : Opening stock of Raw Material	600000		
Product A: + Purchase of Raw Material	100000		
- Closing Stock of Raw Material	400000		

Raw Material of Part A Consumed		300000	
Part B for : Opening stock of Raw Material	600000		
Product B: + Purchase of Raw Material	200000		
- Closing Stock of Raw Material	500000		

Raw Material of Part B Consumed		300000	

		600000	
Manufacturing Overhead			
Product A - Machinery Overhead (per unit)	3100		
Wages (indirect)	322000		
Inspection cost	30200		
	----------	355300	
Product B - Machinery Overhead (per unit)	3720		
Wages (indirect)	322000		
Inspection cost	222000		
	----------	547720	
		----------	903020

Gross Profit of Product A & B

Expenses

Product A:	Depreciation of Machinery	300972		
	Maintenance cost	62000		
	Repairing cost	165000		
	Carrying cost of Raw Material	12500		
	Carrying cost of Finished Product	2500		
	Firing cost	0		
	Hiring cost	0		
		----------	542972	
Product B:	Depreciation of Machinery	601944		
	Maintenance cost	93000		
	Repairing cost	165000		
	Carrying cost of Raw Material	25000		
	Carrying cost of Finished Product	5000		
	Firing cost	0		
	Hiring cost	0		
		----------	889944	
General Expenses (Overhead)		30000		
Paid for Over Draft		0		
Paid for Loan		0		
Rent & Rates		100000	131000	1563916
		----------	----------	----------
Net Profit				907629

Production's Balance Sheet As at Period 1

Fixed Assest :		
Premise :		100000
Machinery A (Cost) :	702268	
Less: Accumulated Depreciation :	300972	
	----------	401296
Machinery B (Cost) :	1404536	
Less: Accumulated Depreciation :	601944	
	----------	802592

		1303888
Current Assets :		
Raw Material of Product A :	400000	
Raw Material of Product B :	500000	
Finished Product A :	0	
Finished Product B :	0	
Cash on Hand :	1977736	

	2877736	
Less: Current Liability		
Over Draft :	0	
Loan :	10000	10000
	----------	----------
		4171624
Financed by		
Capital :		3263995
Net Profit :		907629

The marketing game - program overview

Introduction

In the marketing game, it can support up to six firms. During the period, each firm has its own decision. At the end of each period, three kinds of report will be given automatically. They are the market summary, marketing's balance sheet and the marketing's trading & profit & loss A/C for the year ended of the period. Besides, a competitive profile would be generated if the player requests for it.

Factors concerned in the marketing game :

1. Firm
- initial capital
- approved loan
- rent and rate
- factory overhead
- no. of branch
- no. of salespeople per branch

2. Salespeople
- hire salespeople
- fire salespeople
- new salespeople efficiency
- salary
- commission

3. Product quality
- product quality index

4. Service quality
- branch index
- commission index
- salesforce index
- fraction of non-selling time
-service index

5. Product price
- price index

6. Product Familiarity
- amount of advertising
- familiarity index

Input variables :

1. No. of firms in the game
   - no. of firms share the market in the game
   (range : 1-6)

2. Pay loan
   - return for the last period loan
   (>= 0 < loan value)

3. Approved loan
   - loan approved by the instructor
   (>= 0)

4. No. of branch to be opened
   (>= 0)

5. No. of branch to be closed
   (< exit no. of branch in the firm and >= 0)

6. No. of salespeople to be hired
   (>= 0)

7. No. of salespeople to be fired
   (< exit no. of salespeople in the branch and >= 0)

8. Commission rate (%)
   - the percentage of commission rate given to the salespeople
   (0 < commission rate < 100)

9. Fraction of non-selling time
   - percentage of time for during the customer service
   (0 < non-selling time < 100)

10. Product quantity order
    - product A & B to be ordered (0 to exit)
    (>= 0)

11. Product unit cost
    - unit cost of product A & B ordered
    (> 0)

12. Product quality level
    - quality level of the product ordered percentage of defective
    (0 < quality level =< 100)

13. Selling price of product A & B
    (> 0)

14. Advertising amount
    - advertising amount for product A & B
    (>= 0)

15. Need for the competitor profile
    - output of the competitor profile at the end of period
    (Y/N, if 'y' request profile cost)

16. Output file name
    - files contain the output reports
    ('XXXXXXXX' without extension)

Source file :market.c
Execution file : market.exe
Parameter file : mktpara.txt
History file : mkthis1.txt
Output files : report file for all the six firms 'XXXX' (where XXXX is user defined)
: report file for each firm 'XXXX.n'
(where XXXX is same as the report file and 'n' is the firm no.)
: data file 'XXXX.nd' (where XXXX is same as the report file)

Report type :
1. Marketing's summary for all firms.
2. Competitor profile for all firms (optional)
3. Marketing's balance sheet for each firm
4. Marketing's trading & profit & loss A/C for the end of period

The production game - User's manual

User manual of market.exe

Input the starting period : 'XX' - period number of the marketing

Input the no of firms in the game (1-6) : 'X' - number of firms in the game (minimum is 1 and maximum is 6)

Input the history file name :'XXXX' - name of the history file (e.g. mkthis1)

For each firm, you should input the below:-

Input the value for paid for loan : (0 ≤ paid for loan ≤ approved loan)
Approved loan : (from the instructor approved)
Branch - open new branch(es) : (0 ≤)
- close branch(es) : (0 ≤ close branch(es) ≤ existing number of branch(es))
Salesman - hire salespeople per branch : (0 ≤)
- fire salespeople per branch : (0 ≤ fire salespeople ≤ existing number of salespeople)
- commission rate(%) : (0 ≤ commission rate ≤ 100)
- fraction of non-selling time(%) : : (0 ≤ non-selling time ≤ 100)
Product A - quantity ordered : (0 ≤) (Remark : It will loop back until input = 0)
- unit price : (0 ≤)
- quality level (%) defective : (0 ≤ quality level ≤ 100)
- selling price : (0 ≤)
- Price for advertising : (0 ≤)
Product B - quantity ordered : (0 ≤) (Remark : It will loop back until input = 0)
- unit price : (0 ≤)
- quality level (%) defective : (0 ≤ quality level ≤ 100)
- selling price : (0 ≤)
- Price for advertising : (0 ≤)

After completed all firms data entry:-

1. Do you want the competitor profile ? : - output of the competitor profile at the end of period (Y/N, if 'y' request profile cost)
2. Input a file name for the output file : - files contain the marketing output reports ('XXXX' without extension)
3. Input the new history file name : - files contain the new marketing history file ('XXXX' without extension)

The marketing report included the following:-

1. Marketing's summary for all firms.
2. Competitor profile for all firms (optional)
3. Marketing's balance sheet for each firm
4. Marketing's trading & profit & loss A/C for the end of period

The Marketing Game - Programmer's manual

Marketing

Maximum number of firms is six.

Familiarity

Familiarity index (FI), maximum value is 99.9%

$$\text{Reminding cost} = \frac{an[2 + (n - 1)d]}{2}$$

Where a is the market reminding cost for the first 1% of market,

n = standard FI × (1 - coe) × 100 and d is the advertising factor

Carrying over effect (coe), the population needed to be reminded about the products

If the amount for advertising is greater than the reminding cost, the money will be left for increasing the familiarity index of the product.

$$\text{Remained advertising cost} = \sum_{n=0}^{m} \frac{g}{(1 - \text{standard FI}) - 0.01n}$$

Where g is the growth advertising cost for the first 1% of market, and

m is the increasing of the familiarity index.

FI = standard FI + m when the amount for advertising is greater than the reminding cost.

If more than one product is selling in the market, the reminding cost of the second product is:

$$\text{Second product reminding cost} = \frac{an[2 + (n - 1)d]}{2} \times \text{asoe}$$

Advertising spill over effect (asoe), if a company sells more than one product, there is an advertising spill over effect from one product to the other.

The remained advertising cost is the same to the above.

Product quality

Product quality index (PQI) is between 0 to 1.

$$PQI = \frac{\text{Product quality - standard product quality}}{10}$$

If PQI is smaller than zero, PQI is equal to zero.

Service quality

Service quality index (SQI), maximum value is 99.9%

SQI = standard SQI + growth

growth = $(1 - cs)(1 - 2^{-(ASI - 1)})$

Where cs is the customer satisfaction index from the parameter file.

$$\text{After sales index (ASI)} = \frac{\text{standard after sales service}}{\text{number of product A sold} \times \text{pasu} + \text{number of product B sold} \times \text{p}}$$

Where pasu is the service unit required for product A when one unit is sold, and

pbsu is the service unit required for product B when one unit is sold.

Price

$$\text{Price index (PI)} = \frac{\text{the lowest firm selling price}}{\text{firm's selling price}}$$

Salesforce

Branch weight factor (BWF) denotes the importance of the number of branches and cover factor (CF) is the additional plant can cover the remaining market.
e.g. for 3 branches, the branch index (BI) is:

$$BI = BWF + CF(1 - BWF) + CF[1 - BWF - CF(1 - BWF)]$$

$$\text{Salesman index (SI)} = \frac{\text{number of salesman} \times (1 - \text{non-selling time \%})}{\text{standard number of salesman per branch}}$$

$$\text{Commission index (CI)} = \frac{\text{firm's commission rate}}{\text{average commission rate}}$$

$$\text{Motivation index (MI)} = ssi + (1 - ssi)(1 - 3^{1-CI})$$

Where ssi is the salesman satisfaction index.

$$\text{Salesforces index (SFI)} = BI \times SI \times MI$$

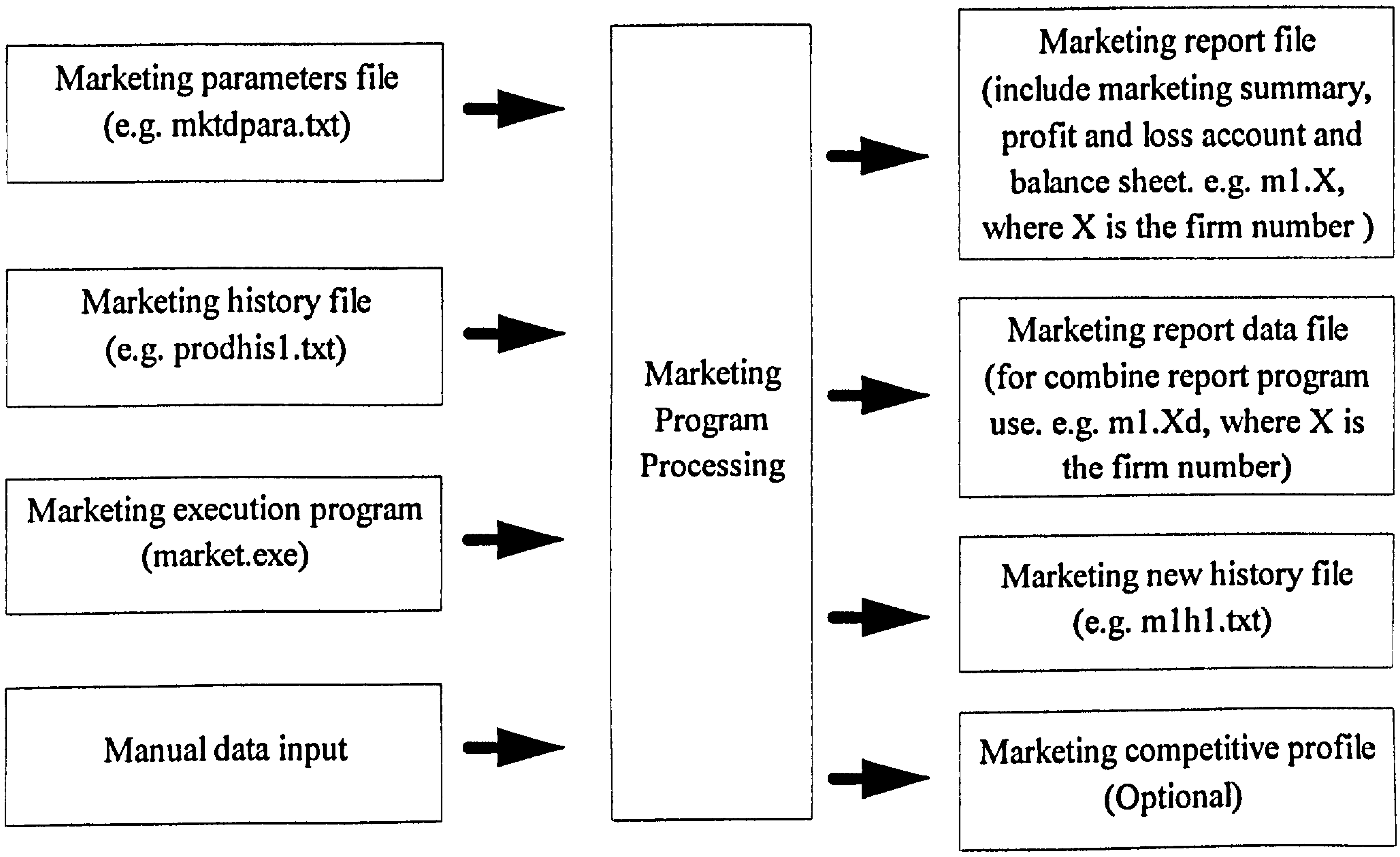
Market share

Firm's potential market = d × firm index + (1 - familiarity factor)(1 - d × firm index)
Where d is the demand of the market
Firm's market index = PFW × PI + SQFW × SQI + PQFW × PQI + SFW × SFI
Where PFW is the price factor weight,
SQFW is the service quality factor weight,
PQFW is the product quality factor weight, and
SFW is the saleforces factor weight.

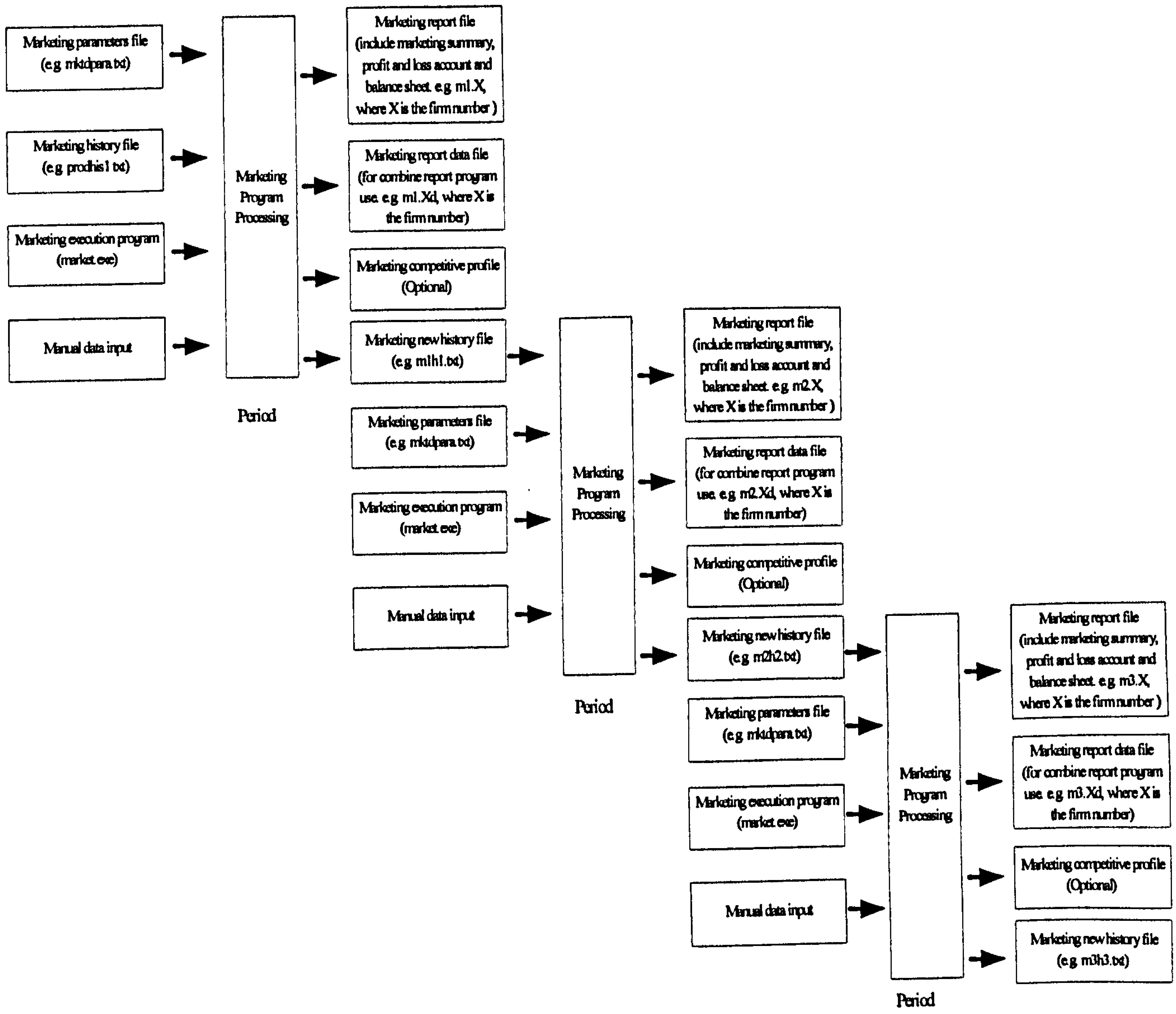
$$\text{Firm's market share} = \frac{\text{firm's market index}}{\text{sum of the market index}}$$

Firm's market size = firm's market share × firm's potential market
If the total market size is smaller than the demand. That means not all the potential customers have purchased the products. If the company do not have enough stock to sell to the customers, half sales will be lost and the other half will be distributed among the others according to the ratio. If the customers still cannot purchase the product for the second time, the sales will be lost.

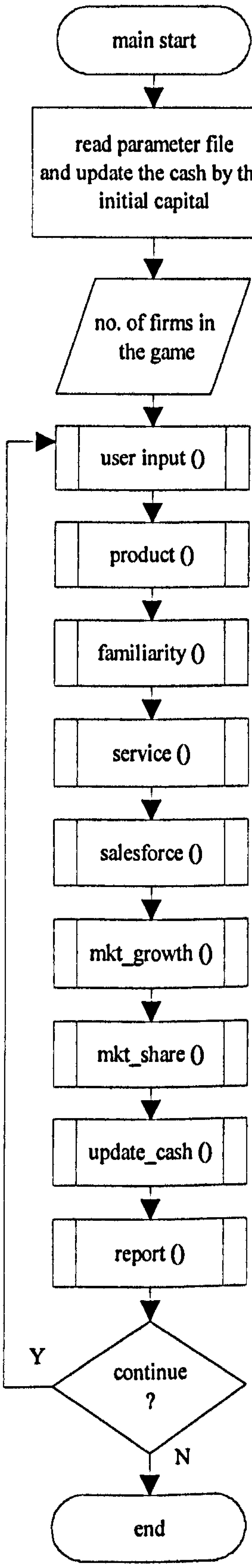
Marketing program data flow chart

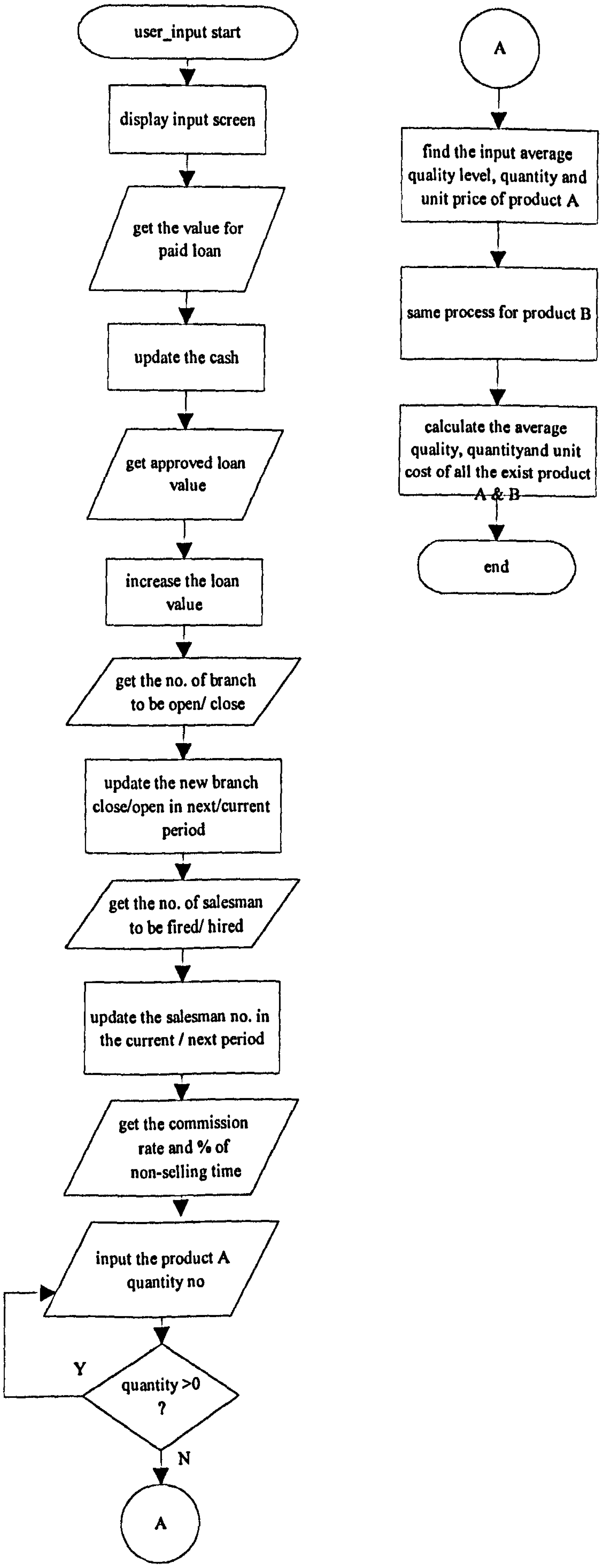


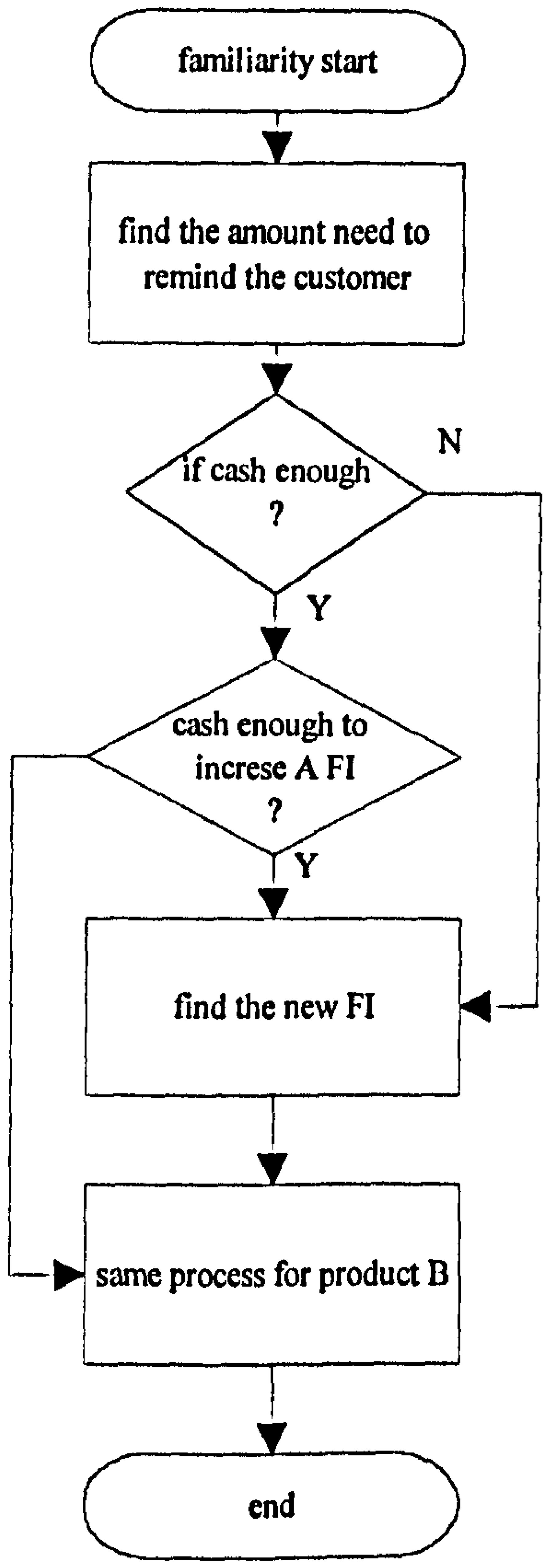
Marketing program in chain data flow chart

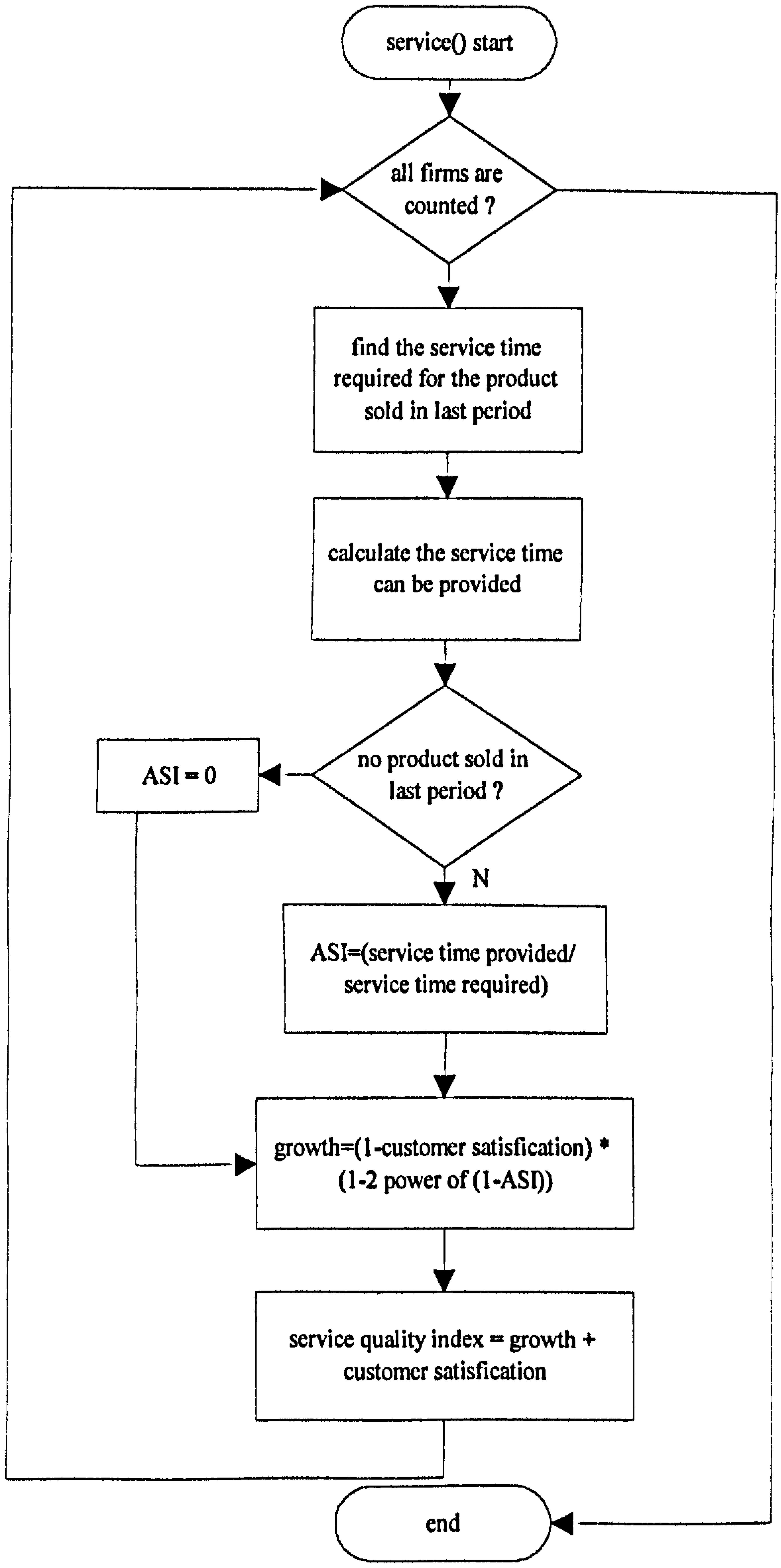


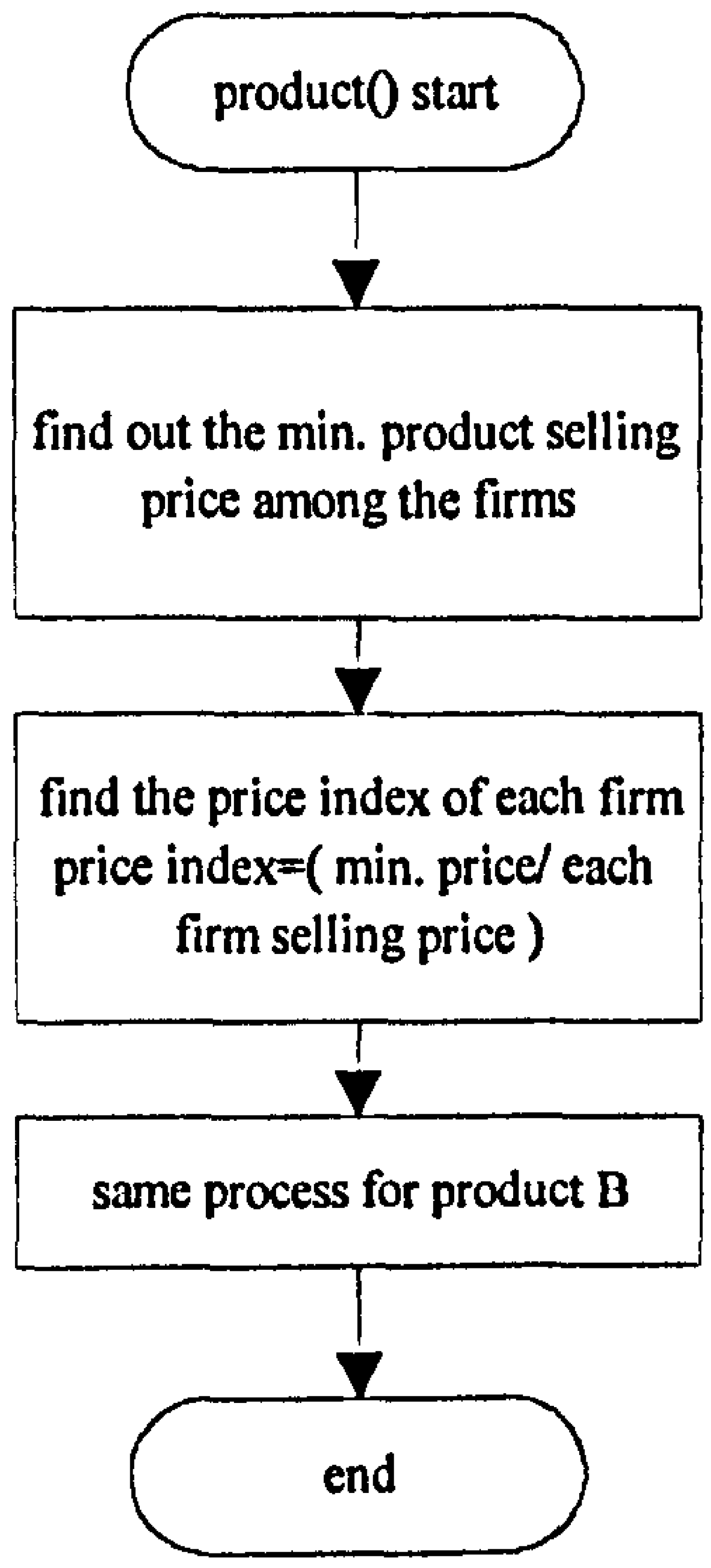
The Marketing Game - Flow Chart



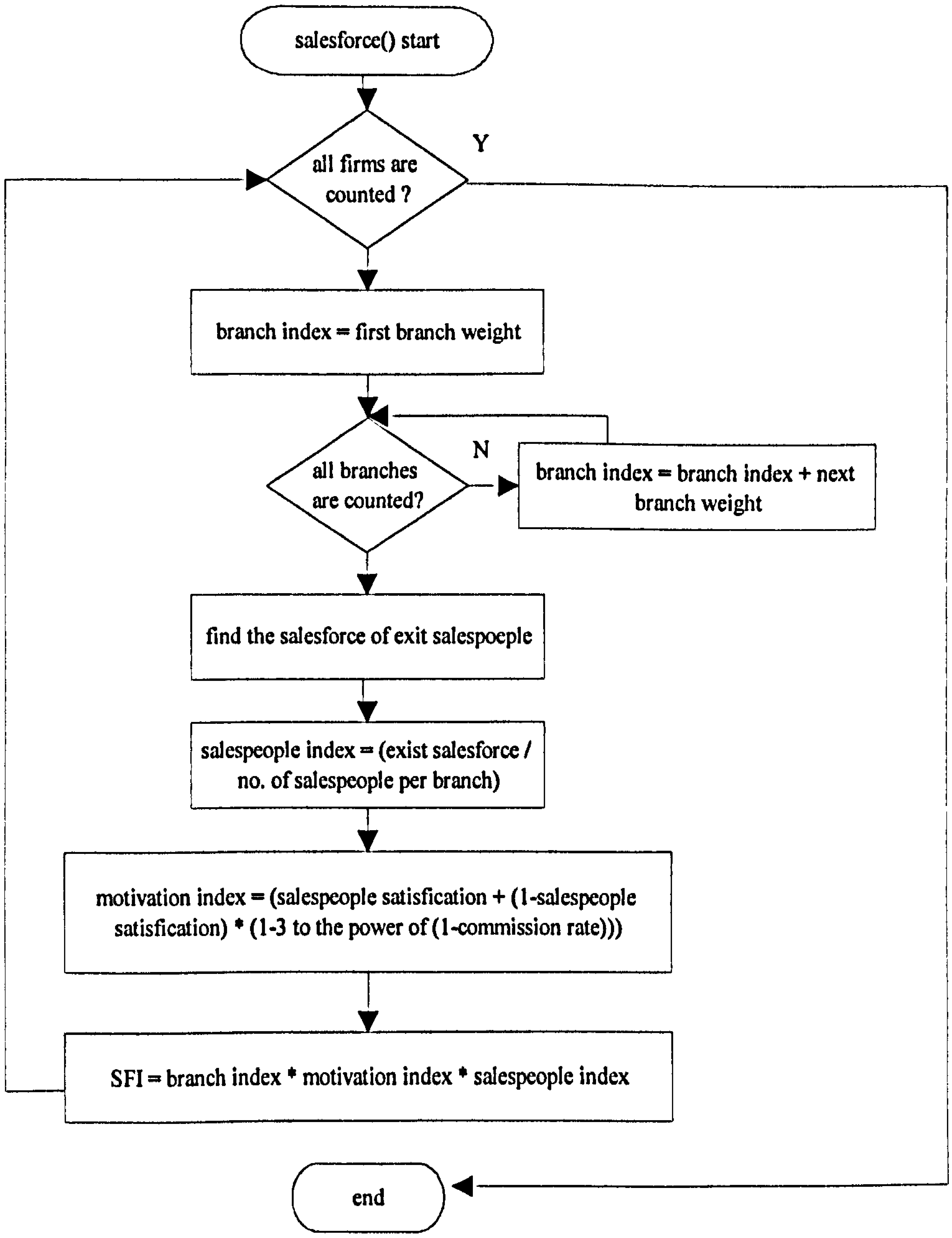

user_input start
display input screen
get the value for paid loan
update the cash
get approved loan value
increase the loan value
get the no. of branch to be open/ close
update the new branch close/open in next/current period
get the no. of salesman to be fired/ hired
update the salesman no. in the current / next period
get the commission rate and % of non-selling time
input the product A quantity no
quantity >0 ?
Y
N
A
A
find the input average quality level, quantity and unit price of product A
same process for product B
calculate the average quality, quantityand unit cost of all the exist product A & B
end

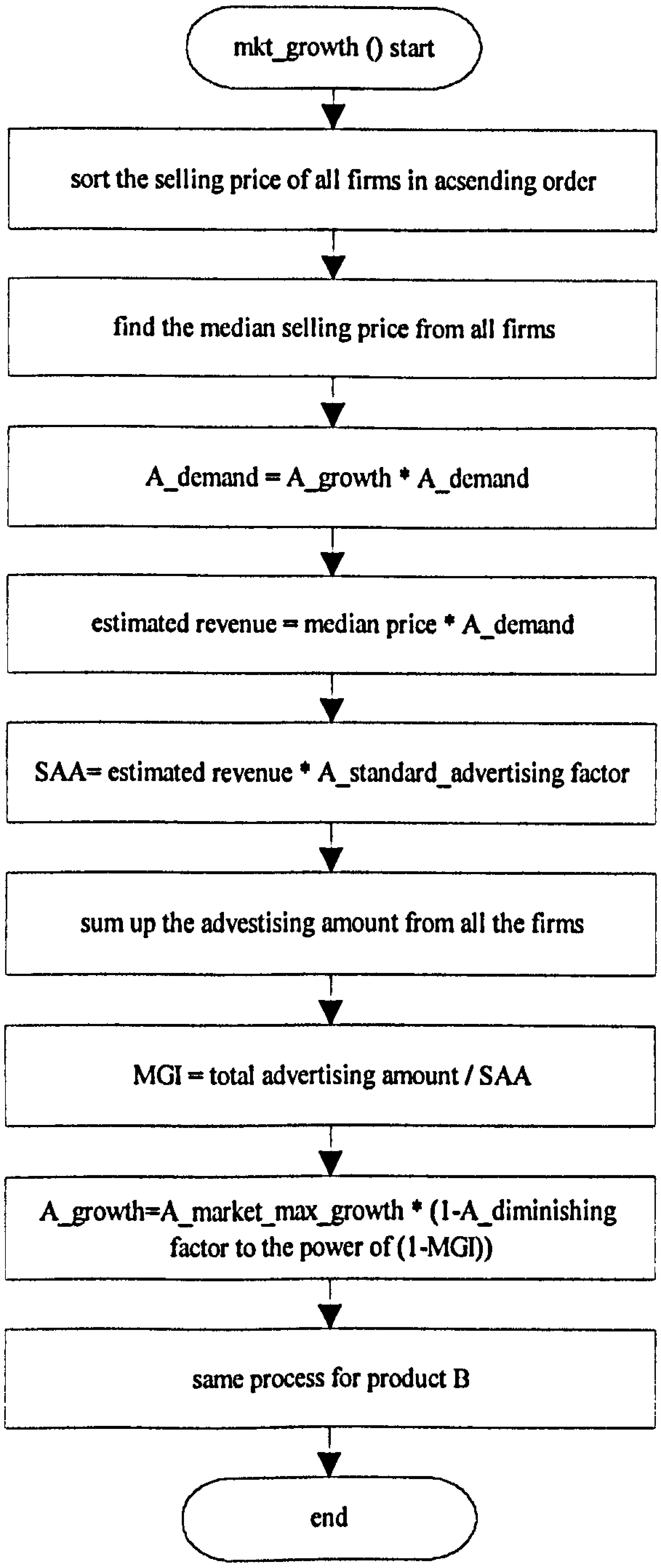

familiarity start
find the amount need to remind the customer
if cash enough ?
N
Y
cash enough to increse A FI ?
Y
find the new FI
same process for product B
end

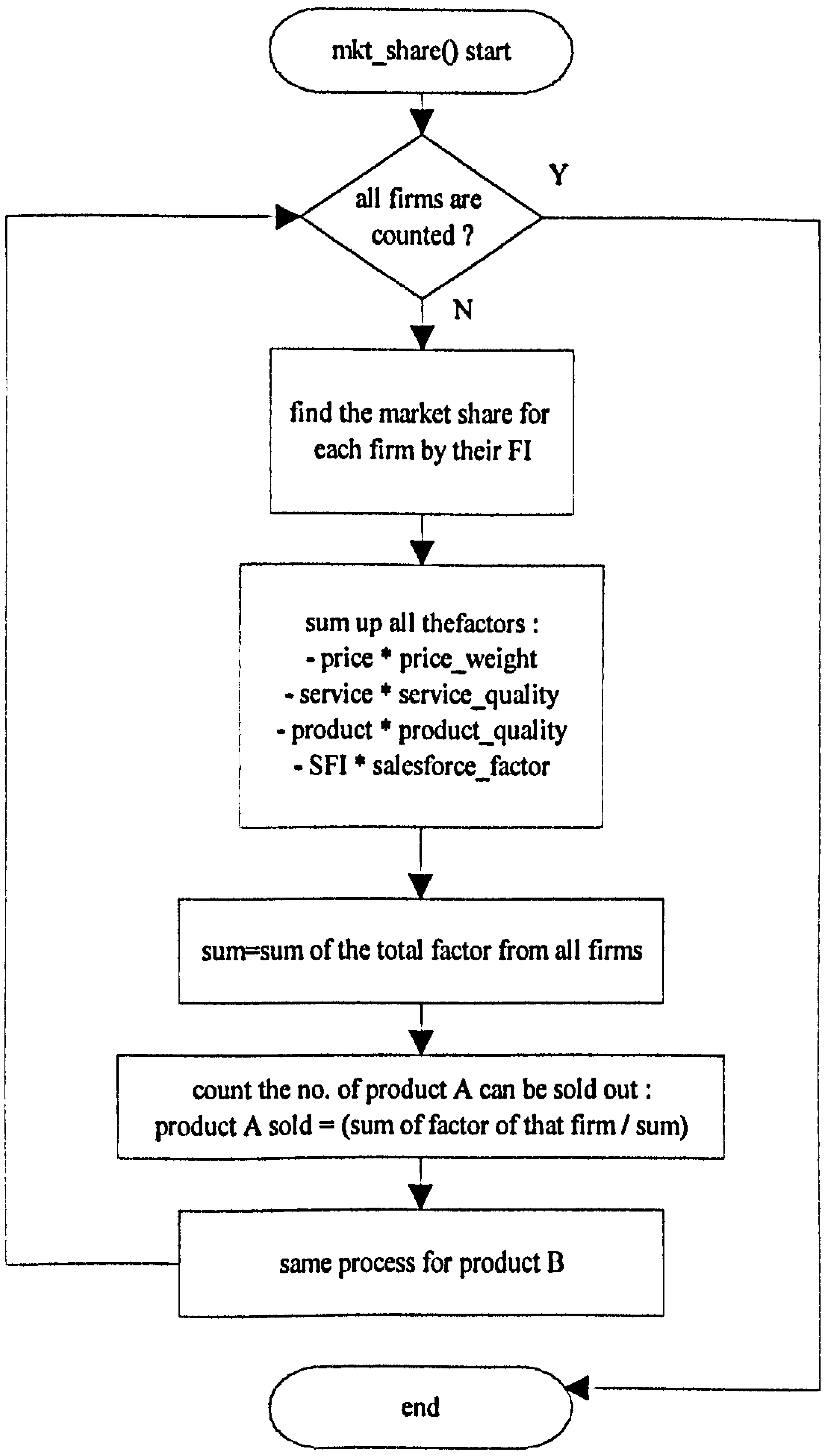

service() start
all firms are counted ?
find the service time required for the product sold in last period
calculate the service time can be provided
ASI = 0
no product sold in last period ?
N
ASI=(service time provided/ service time required)
growth=(1-customer satisfication) * (1-2 power of (1-ASI))
service quality index = growth + customer satisfication
end

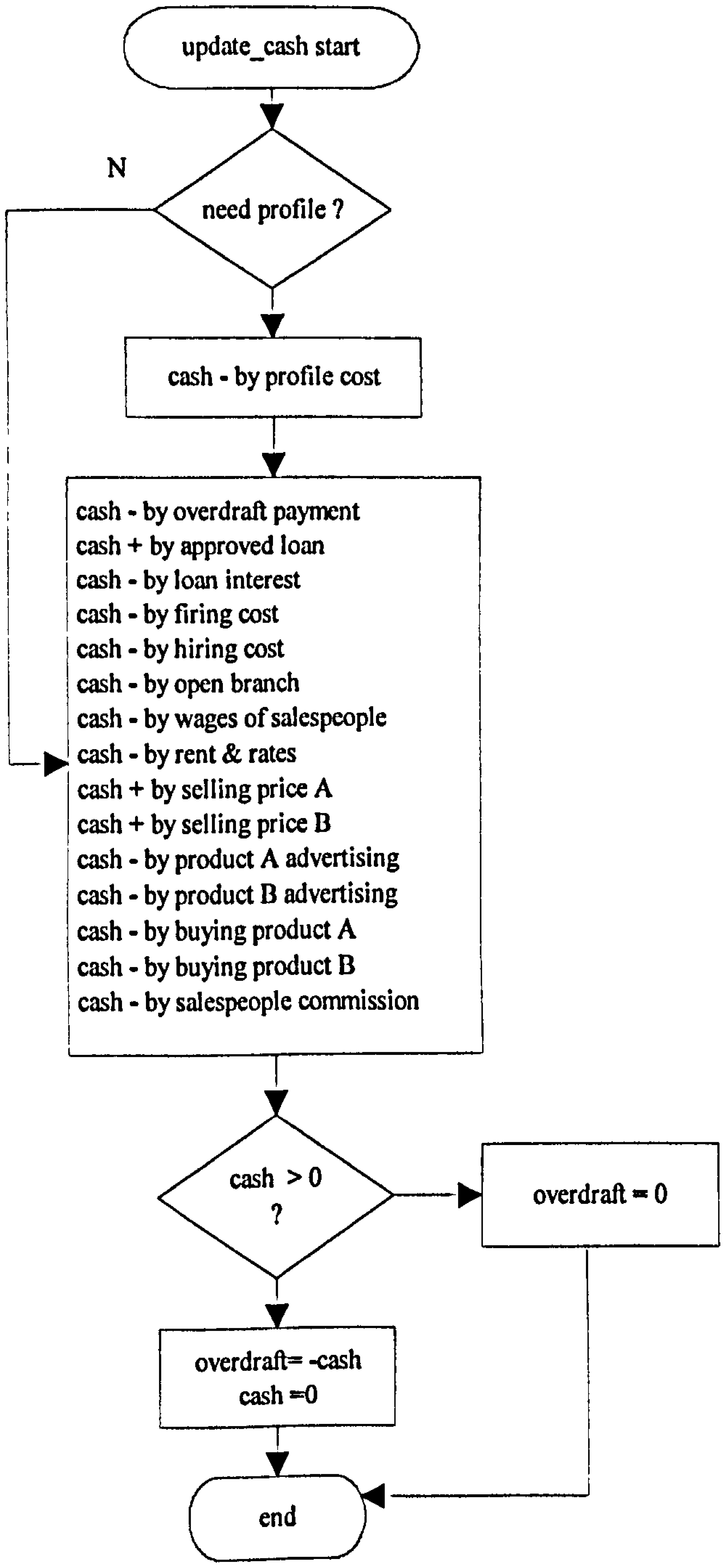


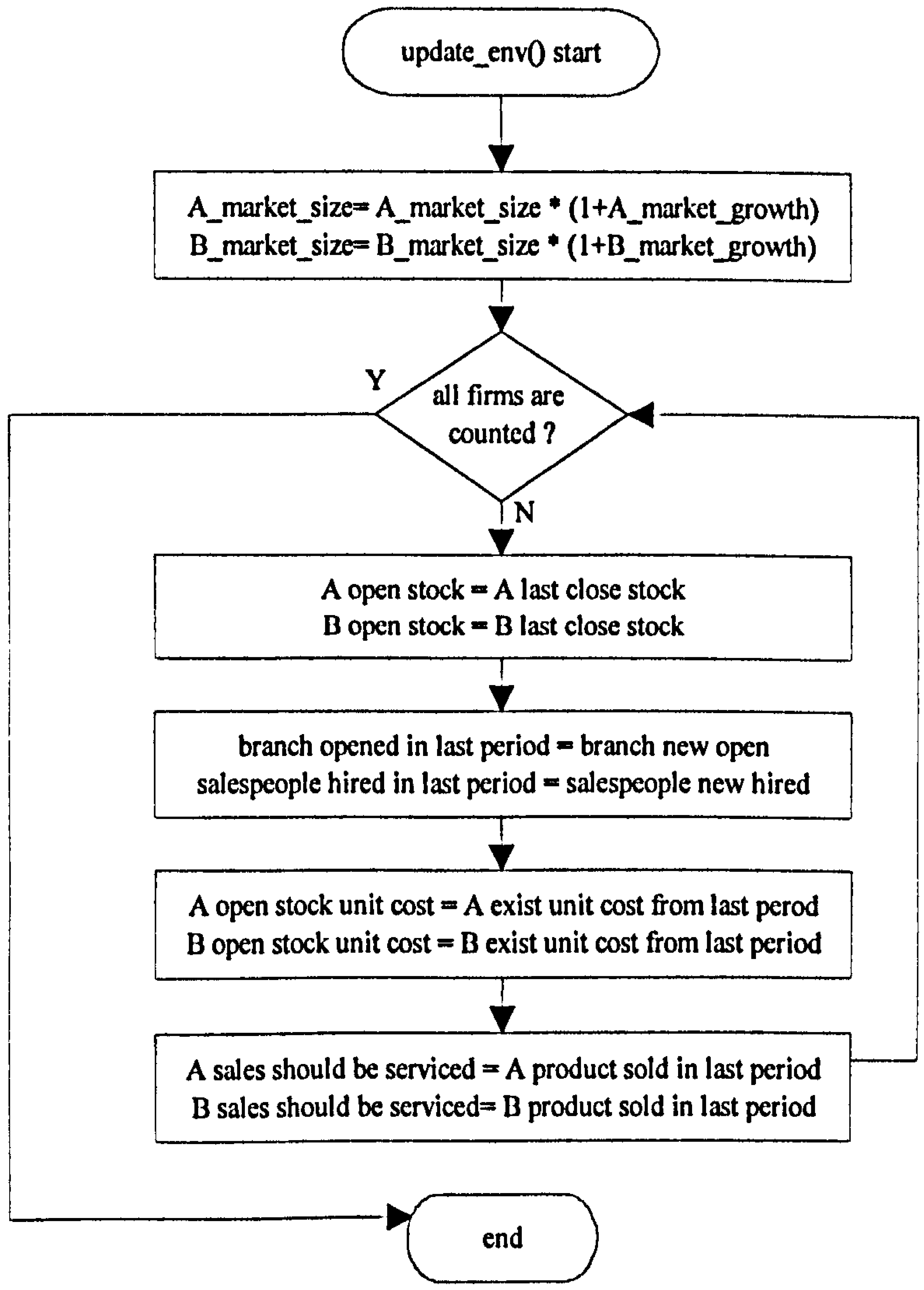
product() start
find out the min. product selling price among the firms
find the price index of each firm price index=(min. price/ each firm selling price)
same process for product B
end


salesforce() start
all firms are counted ?
Y
branch index = first branch weight
all branches are counted?
N
branch index = branch index + next branch weight
find the salesforce of exit salespoeple
salespeople index = (exist salesforce / no. of salespeople per branch)
motivation index = (salespeople satisfication + (1-salespeople satisfication) * (1-3 to the power of (1-commission rate)))
SFI = branch index * motivation index * salespeople index
end


mkt_growth () start
sort the selling price of all firms in acsending order
find the median selling price from all firms
A_demand = A_growth * A_demand
estimated revenue = median price * A_demand
SAA= estimated revenue * A_standard_advertising factor
sum up the advestising amount from all the firms
MGI = total advertising amount / SAA
A_growth=A_market_max_growth * (1-A_diminishing factor to the power of (1-MGI))
same process for product B
end


mkt_share() start
all firms are counted ?
Y
N
find the market share for each firm by their FI
sum up all thefactors :
- price * price_weight
- service * service_quality
- product * product_quality
- SFI * salesforce_factor
sum=sum of the total factor from all firms
count the no. of product A can be sold out :
product A sold = (sum of factor of that firm / sum)
same process for product B
end


update_cash start
N
need profile ?
cash - by profile cost
cash - by overdraft payment
cash + by approved loan
cash - by loan interest
cash - by firing cost
cash - by hiring cost
cash - by open branch
cash - by wages of salespeople
cash - by rent & rates
cash + by selling price A
cash + by selling price B
cash - by product A advertising
cash - by product B advertising
cash - by buying product A
cash - by buying product B
cash - by salespeople commission
cash > 0 ?
overdraft = 0
overdraft= -cash
cash =0
end


update_env() start
A_market_size= A_market_size * (1+A_market_growth)
B_market_size= B_market_size * (1+B_market_growth)
all firms are counted ?
Y
N
A open stock = A last close stock
B open stock = B last close stock
branch opened in last period = branch new open
salespeople hired in last period = salespeople new hired
A open stock unit cost = A exist unit cost from last perod
B open stock unit cost = B exist unit cost from last period
A sales should be serviced = A product sold in last period
B sales should be serviced= B product sold in last period
end

The Marketing game - Program listing

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#include<math.h>
#define YES 1
#define NO 1
/* declare variables in parameter file */

float branch_overhead=0;      /* overhead cost per branch */
float loan_interest=0;        /* loan interest rate */
float overdraft_interest=0;   /* overdraft interest rate */
float branch_setup_cost=0;    /* cost for setup a new branch */
float branch_close_cost=0;    /* cost for close a branch */
float rent_rate=0;            /* rent and rate per firm */
float wages=0;                /* wages persalespeople */
float fire_cost=0;            /* firing cost per head */
float hire_cost=0;            /* hiring cost per head */
float profile_cost=0;         /* cost for making competitors' profile */
float adv_carryover_effect=0; /* advertising carry over effect */
float std_service_unit=0;     /* standard after sales service unit/salespeople */
float cust_satisfy=0;         /* customer satisfaction % */
float salespeople_eff=0;      /* new salespeople efficiency */
float basic_motive=0;         /* basic motivation % */
float salespeople_overhead=0; /* salespeople overhead per branch */

float service_quality=0;      /* % of service quality factor */
float branch_weight=0;        /* branch weight factor */
float other_branch_weight=0;  /* % of additional branch cover the market */
float salesforce=0;           /* % of salesforce factor */
float price=0;                /* % of price factor */
float product_quality=0;      /* % of product quality factor */
float man_satisfy=0;          /* salespeople satisfaction % */

                /* for product A parameters */
long int A_market_size=0;         /* product A market size */
float A_growth=0;                 /* product A growth % per period 1 */
long int A_demand=0;              /* basic demand at period 0 */
float A_product_carry_cost=0;     /* stock product carrying cost */
float A_ff=0;                     /* familiarity factor */
float A_remind_adv=0;             /* cost of reminding advertising per unit */
float A_remind_factor=0;          /* reminding advertising factor */
float A_growth_adv=0;             /* cost of growth advertising per unit */
float A_std_adv_factor=0;         /* standard advertising amount % */
float A_max_growth=0;             /* max. market growth % */
float A_service_unit=0;           /* service unit per unit sold */
float A_df=0;                     /* diminishing factor */
                /* for product B parameters */
long int B_market_size=0;         /* product B market size */
float B_growth=0;                 /* product B growth % per period 1 */
long int B_demand=0;              /* basic demand at period 0 */
float B_product_carry_cost=0;     /* stock product carrying cost */
float B_ff=0;                     /* familiarity factor */
float B_remind_adv=0;             /* cost of reminding advertising per unit */
float B_remind_factor=0;          /* reminding advertising factor */
```

```
float B_growth_adv=0;         /* cost of growth advertising per unit */
float B_std_adv_factor=0;     /* standard advertising amount % */
float B_max_growth=0;         /* max. market growth % */
float B_service_unit=0;       /* service unit per unit sold */
float B_df=0;                 /* diminishing factor */

              /* for firms parameters */
float firm[44][6]={0};   /* i : the firm no. */
                         /* [0][i]  : cash on hand in each firm */
                         /* [1][i]  : product A opening stock */
                         /* [2][i]  : product A familiarity index */
                         /* [3][i]  : product A advertising amount */
                         /* [4][i]  : service quality index */
                         /* [5][i]  : salesforce index */
                         /* [6][i]  : product A market share */
                         /* [7][i]  : product A selling price */
                         /* [8][i]  : product A price index */
                         /* [9][i]  : product B opening stock */
                         /* [10][i] : product B familiarity index */
                         /* [11][i] : product B advertising amount */

                         /* [12][i] : product A can be sold out */
                         /* [13][i] : product B can be sold out */

                         /* [14][i] : product B market share */
                         /* [15][i] : product B selling price */
                         /* [16][i] : product B price index */
                         /* [17][i] : branch no. per firm   */
                         /* [18][i] : salespeople no. per branch */
                         /* [19][i] : commission rate */
                         /* [20][i] : % of service time */

                         /* [21][i] : A input average unit price */
                         /* [22][i] : A input order quantity */
                         /* [23][i] : A input average quality level */
                         /* [24][i] : A product sold last period */
                         /* [25][i] : B average unit price */
                         /* [26][i] : B order quantity */
                         /* [27][i] : B average quality level */
                         /* [28][i] : B product sold last period */

                         /* [29][i] : A closing stock */
                         /* [30][i] : B closing stock */
                         /* [31][i] : sales of product A */
                         /* [32][i] : sales of product B */
                         /* [33][i] : non-selling time */
                         /* [34][i] : product A open stock unit cost */
                         /* [35][i] : product B open stock unit cost */
                         /* [36][i] : product A stock average quality level */
                         /* [37][i] : product B stock average quality level */
                         /* [38][i] : product A close stock unit cost */
                         /* [39][i] : product B close stock unit cost */
                         /* [40][i] :  net profit */
                         /* [41][i] : product A quality */
                         /* [42][i] : product B quality */
                         /* [43][i] : firm index */

/* global variables in program */
```

```
float overdraft[6]={0};        /* overdraft when negative cash */
float loan[6]={0};             /* approved loan */
float pay_over[6]={0};         /* paid for overdraft */
float pay_loan[6]={0};         /* paid loan */


int new[6][6]={0};       /* new[0][i]  : no. of branch to be opened */
                             /* new[1][i]  : no. of branch to be closed */
                             /* new[2][i]  : hire no. of salespeople */
                             /* new[3][i]  : fire no. of salespeople */
                             /* new[4][i]  : salespeople hired in last period */
                             /* new[5][i]  : branch opened in last period */


int firm_no=0;           /* total no. of firms in game */
float fire_man_cost[6]={0};       /* cost for firing salespeople */
float hire_man_cost[6]={0};       /* cost for hiring salespeople */
float open_cost[6]={0};       /* cost for open branch */
float close_cost[6]={0};      /* cost for close branch */
int profile=1;           /* 1 = require for profile */
int period=1;            /* period counter */


float getin(int row,int col,char type)
{
        char string[20];
        int length=0,error;
        float num;
        do
        {
                error=0;
                gotoxy(col,row);
                clreol();
                gotoxy(col,row);
                gets(string);
                length=strlen(string);
                switch(type)
                {
                        case 'i' : num=atoi(string); break;
                        case 'l' : num=atol(string); break;
                        case 'f' : num=atof(string); break;
                }
                if (num==0)
                        if (string[0]!='0' || length>1)
                                error =1;
        }while (error);
                /*    return(num);*/
        switch(type)
        {
                case 'i' : return((int)num);
                case 'l' : return((long int)num);
                case 'f' : return(num);
        }
}

/*------------------------------------------------------\
/ function : goto coordinate and clear the rest of line \
/    input : y-coordincate, x-coordinate               \
```

```
/------------------------------------------------------*/
goxy(int row,int col)
{
        gotoxy(col,row);
        clreol();
        gotoxy(col,row);
}


main()
{

        readfile();
        printf("\nInput the starting period  :");
   scanf("%d",&period);

   printf("Input the no. of firms in the game (1-6):");
        firm_no=getin(0,42,'i');
        readhis();
        userinput();
                product();
                familiarity();
                service();
                sales_force();
                mkt_growth();
                mkt_share();
                update_cash();
                report();
        update_env();
        writehis();
   }


readfile()
{
        char string[50];
   FILE *para;
   int i=0;

   clrscr();
   printf("Initializing the system ........\n");
   if ((para=fopen("mktpara.txt","r"))==NULL)
        printf("<<ERROR : Parameter file open error !>>\n");

        fscanf(para,"%s %f",string,&branch_overhead);
   fscanf(para,"%s %f",string,&loan_interest);
   fscanf(para,"%s %f",string,&overdraft_interest);
   fscanf(para,"%s %f",string,&branch_setup_cost);
   fscanf(para,"%s %f",string,&branch_close_cost);
   fscanf(para,"%s %f",string,&rent_rate);
   fscanf(para,"%s %f",string,&wages);
   fscanf(para,"%s %f",string,&fire_cost);
   fscanf(para,"%s %f",string,&hire_cost);
   fscanf(para,"%s %f",string,&profile_cost);
   fscanf(para,"%s %f",string,&adv_carryover_effect);
        fscanf(para,"%s %f",string,&std_service_unit);
   fscanf(para,"%s %f",string,&cust_satisfy);
   fscanf(para,"%s %f",string,&salespeople_eff);
```

```
    fscanf(para,"%s %f",string,&basic_motive);
    fscanf(para,"%s %f",string,&salespeople_overhead);
    fscanf(para,"%s %f",string,&service_quality);
    fscanf(para,"%s %f",string,&branch_weight);
    fscanf(para,"%s %f",string,&other_branch_weight);
    fscanf(para,"%s %f",string,&salesforce);
    fscanf(para,"%s %f",string,&price);
    fscanf(para,"%s %f",string,&product_quality);
    fscanf(para,"%s %f",string,&man_satisfy);

            /* for product A parameters */
    fscanf(para,"%s %ld",string,&A_market_size);
    fscanf(para,"%s %f",string,&A_product_carry_cost);
    fscanf(para,"%s %f",string,&A_ff);
    fscanf(para,"%s %f",string,&A_remind_adv);
    fscanf(para,"%s %f",string,&A_remind_factor);
    fscanf(para,"%s %f",string,&A_growth_adv);
    fscanf(para,"%s %f",string,&A_std_adv_factor);
    fscanf(para,"%s %f",string,&A_max_growth);
    fscanf(para,"%s %f",string,&A_service_unit);
    fscanf(para,"%s %f",string,&A_df);

            /* for product B parameters */
    fscanf(para,"%s %ld",string,&B_market_size);
    fscanf(para,"%s %f",string,&B_product_carry_cost);
    fscanf(para,"%s %f",string,&B_ff);
    fscanf(para,"%s %f",string,&B_remind_adv);
    fscanf(para,"%s %f",string,&B_remind_factor);
    fscanf(para,"%s %f",string,&B_growth_adv);
    fscanf(para,"%s %f",string,&B_std_adv_factor);
    fscanf(para,"%s %f",string,&B_max_growth);
    fscanf(para,"%s %f",string,&B_service_unit);
    fscanf(para,"%s %f",string,&B_df);


    for(i=0;i<6;i++)            /* get firm index */
        fscanf(para,"%s %f",string,&firm[43][i]);

    fclose(para);
}

/*--------------------------------------------------------\
/     function : get user input with coordinates           \
/       input : y-coordinate, x-coordinate, input type     \
/ allow input type : integer, long integer and floating point \
/---------------------------------------------------------*/

userinput()
{
    int i;
        float quantity, level, unit_price=0;

    clrscr();
    for(i=0;i<firm_no;i++)
    {
        clrscr();
        printf("\nFirm %d:",i+1);
        /* input_scr() */
```

```
printf("   Input the value for paid for loan (loan=%7.2f):\n\n",loan[i]);
printf("   Approved loan :\n");
printf("\n   Branch - open new branch(es) :\n");
printf("          - close branch(es) :\n");
printf("   Salesman - hire salespeople per branch :\n");
printf("            - fire salespeople per branch :\n");
printf("            - commission rate(%)  :\n");
printf("            - fraction of non-selling time(%) :\n");
printf("   Product A - quantity ordered :\n");
printf("             - unit cost :\n");
printf("             - quality level (defective/lot) :\n");
printf("             - selling price :\n");
printf("             - Price for advertising :\n");
printf("   Product B - quantity ordered :\n");
printf("             - unit cost :\n");
printf("             - quality level (defective/lot) :\n");
printf("             - selling price :\n");
printf("             - Price for advertising :\n");
/* end of input scr() */
firm[0][i]-=(loan[i]*loan_interest);
do
    {
        pay_loan[i]=getin(2,66,'f'); /* Get the value for paid the load */
    }while(pay_loan[i]>loan[i]); /* check the input enough or too much */
    loan[i]-=pay_loan[i];  /*loan- by paid loan*/
    firm[0][i]-=pay_loan[i]; /* cash- by paid loan*/

    loan[i]+=getin(4,25,'f');          /* approved loan */
    new[0][i]=getin(6,40,'i');    /* open branch no. */
            /* new branch setup at next period */
    do
    {
            new[1][i]=getin(7,37,'i');     /* close branch no. */
    }while(new[1][i]>firm[17][i]); /* check the input not over exist branch*/

    new[2][i]=getin(8,50,'i');   /* hire salespeople no. */
    /* all new salespeople arrive at next period */
    do
    {
            new[3][i]=getin(9,50,'i');    /* fire salespeople no. */
    }while(new[3][i]>firm[18][i]); /* check the input not ver exist salespeople */

    open_cost[i]=new[0][i]*branch_setup_cost;  /* new[0][i]:no. of new branch */
    close_cost[i]=new[1][i]*branch_close_cost;  /* new[1][i]:no. of branch is closed */
                    /* new[2][i]:no.of salespeople hired */
                    /* new[3][i]:no.of salespeople fired */
                                        /* firm[17][i]:no.of branch exist */
                    /* firm[18][i]:no.of salespeople/branch exist */

if(new[1][i]>0)     /* branch is closed, so salespeople are also fired */
                    fire_man_cost[i]=new[1][i]*firm[18][i]*fire_cost;
    else
                    fire_man_cost[i]=0;

    firm[17][i]-=new[1][i];      /* branch closed immediately */
firm[18][i]-=new[3][i];      /* all fired salespeople leave immediately */

    fire_man_cost[i]+=new[3][i]*firm[17][i]*fire_cost;
```

```
        hire_man_cost[i]=(new[2][i]*firm[17][i]+(firm[18][i]+new[2][i])*new[0][i])*hire_cost;
/* when new branch is opened, it has the same no. of salespeople as other branches */

        firm[19][i]=(getin(10,41,'f')/100);  /* commission rate */
        firm[20][i]=(getin(11,54,'f')/100);  /* % of service time */
        quantity=getin(12,39,'f');  /* product A quantity ordered */
        while(quantity>0)
        {                        /* firm[23][i]=A average quality level */
          unit_price=getin(13,34,'f');     /* firm[21][i]=A unit price  */
          level=getin(14,49,'f');       /* firm[22][i]=A quantity */
          level=level/1000;
          firm[21][i]=(firm[22][i]*firm[21][i]+unit_price*quantity)/(firm[22][i]+quantity);
          firm[23][i]=(firm[22][i]*firm[23][i]+level*quantity)/(firm[22][i]+quantity);
          firm[22][i]+=quantity;
          quantity=getin(12,39,'f');  /* product quantity ordered */
        }
                /* [22][i]  : A input average unit price */
                /* [21][i]  : A input total order quantity */
                          /* [23][i]  : A input average quality level */
                          /* [36][i]  : product A stock average quality level */
                          /* [37][i]  : product B stock average quality level */
                          /* [38][i]  : product A close stock unit cost */
                          /* [1][i]   : product A opening stock */
                          /* [9][i]   : product B opening stock */
                          /* [34][i]  : product A open stock unit cost */
                          /* [35][i]  : product B open stock unit cost */

          /*-- calculate the average A product quality level among the input & stock */
        if((firm[1][i] && firm[22][i])>0)
        {
          firm[36][i]=(firm[1][i]*firm[36][i]+firm[22][i]*firm[23][i])/(firm[1][i]+firm[22][i]);
                firm[41][i]=(10-firm[36][i])*.1;
                if (firm[41][i]<0)
                        firm[41][i]=0;
          /*-- calculate the average A product unit cost among the input & stock */
          firm[38][i]=(firm[1][i]*firm[34][i]+firm[22][i]*firm[21][i])/(firm[1][i]+firm[22][i]);
        }

                firm[7][i]=getin(15,37,'f');  /* product selling price */
                firm[3][i]=getin(16,44,'f');   /* advertising amount */

                quantity=getin(17,39,'f');  /* product B quantity ordered */
        while(quantity>0)
        {                        /* firm[27][i]=B average quality level */
        unit_price=getin(18,34,'f');     /* firm[26][i]=B quantity */
        level=(getin(19,49,'f')/1000);         /* firm[25][i]=B average unit price */
        firm[25][i]=(firm[25][i]*firm[26][i]+unit_price*quantity)/(firm[26][i]+quantity);
        firm[27][i]=(firm[27][i]*firm[26][i]+level*quantity)/(firm[26][i]+quantity);
         firm[26][i]+=quantity;
          quantity=getin(17,39,'f');  /* product quantity ordered */
        }
          /*-- calculate the average B product quality level among the input & stock */
        if((firm[9][i] && firm[26][i])>0)
        {
          firm[37][i]=(firm[9][i]*firm[37][i]+firm[26][i]*firm[27][i])/(firm[9][i]+firm[26][i]);
        firm[42][i]=(10-firm[37][i])*.1;
                if (firm[42][i]<0)
                        firm[42][i]=0;
```

```
  /*-- calculate the average B product unit cost among the input & stock */
            firm[39][i]=(firm[9][i]*firm[35][i]+firm[26][i]*firm[25][i])/(firm[9][i]+firm[26][i]);
         }
                  firm[15][i]=getin(20,37,'f');  /* product selling price */
                  firm[11][i]=getin(21,44,'f');   /* advertising amount */
  }
}

product()     /* calculate the price index */
{
  float min_price=0;
  int i=0;

  /* -------- for product A ----------*/
        min_price=firm[7][0];        /* firm[7][i]= product A selling price */
  for(i=1;i<firm_no;i++)
        if(firm[7][i]<min_price)
          min_price=firm[7][i];

        for(i=0;i<firm_no;i++)            /* firm[8][i]= product A price index */
        firm[8][i]=min_price/firm[7][i];

  /* -------- for product B ----------*/
        min_price=firm[15][0];       /* firm[15][i]= product B selling price */
  for(i=1;i<firm_no;i++)
        if(firm[15][i]<min_price)
          min_price=firm[15][i];

        for(i=0;i<firm_no;i++)            /* firm[16][i]= product B price index */
        firm[16][i]=min_price/firm[15][i];

}

familiarity()     /* calculate the new FI */
{
        int i;
  float remind=0;    /* percentage to be remined */
  float remind_cost=0;    /* cost used for reminding */
  float growth_cost=0;   /* cost remain for growth */
  float a,d=0;
        float A,B,C=0;        /* used for quadratic equation */
        float temp=0;
        float ans=0;

  /*-------------- for product A ---------------*/
  for(i=0;i<firm_no;i++)          /* firm[3][i]:A's advertising amount */
  {
        remind=(1-adv_carryover_effect)*firm[2][i]*100;   /* firm[2][i]:A's FI */
  a=A_market_size*A_remind_adv/100;   /* cost to remind the first 1% */
        d=a*A_remind_factor;            /* cost required for later 1 % */
        remind_cost=0.5*remind*(2*a+(remind-1)*d);

        if(remind_cost<firm[3][i])  /* advertising amount can support reminding */
        {
          growth_cost=firm[3][i]-remind_cost; /* cost for increse the FI */
        a=A_market_size*A_growth_adv/100;
                       d=firm[2][i];                 /* current FI level */
          while(1)
```

```
        {
                        if(a/(1-d)>growth_cost)    /* cannot grow 1 % more */
            break;
          else
          {
            if(d>=0.98)
                break;          /* reach the max. FI */
            else
            {
                                    growth_cost-=a/(1-d);
                                    d+=0.01;          /* grow 1 % more */
            }
          }
        }
                firm[2][i]=d;       /* update the new FI after grow */
      }
      else              /* can't support FI growth */
      {
                A=d;                /* use quadratic equation to find */
        B=d-2*a;            /* max pertcentage that the advertising */
        C=firm[3][i]*(-1);      /* amount can support */
                temp=sqrt(B*B-4*A*C);
                ans=(temp-B)/(2*A);  /* remind=max can remind % */
                if (!(ans>(-1)*remind && ans<remind))
                    ans=(-1)*(B+temp)/(2*a);

                remind=ans;
        firm[2][i]=firm[2][i]*adv_carryover_effect+remind/100;  /* new FI after reminding */
      }
  }

  /*------------- for product B ---------------*/
  for(i=0;i<firm_no;i++)          /* firm[11][i]:B's advertising amount */
  {
      remind=(1-adv_carryover_effect)*firm[10][i]*100;   /* firm[10][i]:B's FI */
  a=B_market_size*B_remind_adv/100;   /* cost to remind the first 1% */
      d=a*B_remind_factor;          /* cost required for later 1 % */
      remind_cost=0.5*remind*(2*a+(remind-1)*d);

      if(remind_cost<firm[11][i])  /* advertising amount can support reminding */
      {
        growth_cost=firm[11][i]-remind_cost; /* cost for increse the FI */
      a=B_market_size*B_growth_adv/100;
        d=firm[10][i];              /* current FI level */
        while(1)
        {
if(a/(1-d)>growth_cost)    /* cannot grow 1 % more */
            break;
          else
          {
            if(d>=0.98)
                break;          /* reach the max. FI */
            else
            {
                                    growth_cost-=a/(1-d);
                                    d+=0.01;          /* grow 1 % more */
            }
          }
```

```
            }
            firm[10][i]=d;          /* update the new FI after grow */
         }
         else                    /* can't support FI growth */
         {
            A=d;                    /* use quadratic equation to find */
                           B=d-2*a;             /* max pertcentage that the advertising */
            C=firm[11][i]*(-1);        /* amount can support */
            temp=sqrt(B*B-4*A*C);
                           ans=(temp-B)/(2*A);  /* remind=max can remind % */
                           if (!(ans>(-1)*remind && ans<remind))
                                 ans=(-1)*(B+temp)/(2*a);

            remind=ans;
            firm[10][i]=firm[10][i]*adv_carryover_effect+remind/100;  /* new FI after reminding
*/
         }
   }
}

service()     /* product A & B have the same service quality level */
{
         int i;
   float R_service_time=0;    /* required service time for product A+B */
   float S_service_time=0;    /* service time can affort */
   float service_man=0;       /* no. of salespeople do non-selling */
   float ASI=0;               /* sales service index */
   float growth=0;            /* growth % of the ASI */

   for(i=0;i<firm_no;i++)     /* firm[24][i]:no.of A product sold last period */
   {                   /* firm[28][i]:no.of B product sold last period */

         R_service_time=firm[24][i]*A_service_unit+firm[28][i]*B_service_unit;
   service_man=(firm[18][i]-new[2][i]+new[2][i]*salespeople_eff-
salespeople_overhead)*firm[17][i]*firm[20][i];
                                    /* firm[17][i]:no.of branch */
                    /* firm[18][i]:no.of salespeople per branch */
                                    /* firm[20][i]: % of non-selling time */
                    /* new[2][i] :no.of salespeople hired in last period */
         S_service_time=service_man*std_service_unit;
   firm[33][i]=S_service_time;
   if(firm[24][i]==0 && firm[28][i]==0)
            ASI=0;
         else
            ASI=S_service_time/R_service_time;

         growth=(1-cust_satisfy)*(1-pow(2,(1-ASI)));
         firm[4][i]=cust_satisfy+growth;  /* firm[4][i]:service quality index */
   }
}

sales_force()   /* calculate the SFI, product A & B have the same SFI */
{
         int i,j;
   float BI=0;       /* branch index */
   float SI=0;       /* salespeople index */
   float MI=0;       /* motivation index */
   float salespeople=0;  /* no. of salespeople work for selling */
```

```
for(i=0;i<firm_no;i++)
        {                   /* firm[17][i]:no.of branch in each firm */
        BI=branch_weight;
                for(j=0;j<firm[17][i]-1;j++)
           BI+=(1-BI)*other_branch_weight;

                /* new[2][i]:no.of salespeople hired in last period */
                        /* firm[20][i]: % of non-selling time */
                /* firm[18][i]:no.of salespeople per branch */
salespeople=(firm[18][i]-new[2][i]+new[2][i]*salespeople_eff)*(1-firm[20][i]);
SI=(salespeople-salespeople_overhead)/firm[18][i];

MI=man_satisfy+(1-man_satisfy)*(1-pow(3,(1-firm[19][i])));
                                /* firm[19][i]:commission rate */

        firm[5][i]=BI*SI*MI;   /* firm[5][i] : SFI */
  }
}

mkt_growth() /* find the market growth for the next period */
{
        int i,j;
  float price_list[6];      /* list for sorting to find the median */
  float temp=0;
  float min=0;
  float median=0;           /* the median selling price among the firms */
        float est_revenue=0;     /* estimated revenue */
  float SAA=0;              /* standard advertising amount */
  float total_adv=0;        /* total amount for advertising */
  float MGI=0;              /* market growth index */

  /*------- for product A -----------*/

  for(i=0;i<6;i++)
        price_list[i]=firm[7][i];  /* firm[7][i]:product A selling price */

  for(j=0;j<firm_no;j++)  /* arrange the price in ascending order */
        for(i=j;i<firm_no;i++)
        {
          min=j;
          if(price_list[min]>price_list[i])
          {
                min=i;
                temp=price_list[j];
                price_list[j]=price_list[min];
                price_list[min]=temp;
          }
        }

  if(firm_no%2==0)
                median=(price_list[firm_no/2-1]+price_list[firm_no/2])/2;
  else
                median=price_list[((firm_no+1)/2)-1];

  est_revenue=median*A_demand;
  SAA=est_revenue*A_std_adv_factor;
```

```
for(i=0;i<firm_no;i++)
        total_adv+=firm[3][i];

MGI=total_adv/SAA;

A_growth=A_max_growth*(1-pow(A_df,(1-MGI)));
                                    /* the market growth % for next period */
                                    /* A_df is the diminishing factor */

/*------- for product B -----------*/

for(i=0;i<6;i++)
            price_list[i]=firm[15][i];  /* firm[16][i]:product B selling price */

for(j=0;j<firm_no;j++)  /* arrange the price in ascending order */
        for(i=j;i<firm_no;i++)
        {
          min=j;
          if(price_list[min]>price_list[i])
          {
                min=i;
                temp=price_list[j];
                price_list[j]=price_list[min];
                price_list[min]=temp;
          }
        }

if(firm_no%2==0)
            median=(price_list[firm_no/2-1]+price_list[firm_no/2])/2;
else
            median=price_list[((firm_no+1)/2)-1];

est_revenue=median*B_demand;
SAA=est_revenue*B_std_adv_factor;

for(i=0;i<firm_no;i++)
        total_adv+=firm[11][i];

MGI=total_adv/SAA;

B_growth=B_max_growth*(1-pow(B_df,(1-MGI)));
                                    /* the market growth % for next period */
                                    /* B_df is the diminishing factor */
}

mkt_share()
{
        int i;
   float a=0;   /* a=FI*demand */
   float b=0;   /* b=(1-FF)(demand-FI*demand)=(1-FF)*demand(1-FI) */
   float total[6]={0}; /* total of weighting factors */
   float sum=0;   /* sum of the total from all the firms */

   /*-------- for product A ----------*/
   for(i=0;i<firm_no;i++)
   {
   a=firm[43][i]*A_demand;              /* firm[2][i]:FI */
   b=(1-A_ff)*A_demand*(1-firm[43][i]);
```

```
        firm[6][i]=a+b;              /* A market share for each firm */

        /*----- calculate the weights ----------*/
        total[i]=firm[8][i]*price;          /* firm[8][i]: product A price index */
        total[i]+=firm[4][i]*service_quality; /* firm[4][i]:service quality index */
        /* calculate the quality index */
   total[i]+=firm[41][i]*product_quality; /* firm[36][i]:A stock quality level */
   total[i]+=firm[5][i]*salesforce;      /* firm[5][i]: SFI */
   }
   for(i=0;i<firm_no;i++)
        sum+=total[i];

   for(i=0;i<firm_no;i++)
        firm[12][i]=firm[6][i]*total[i]/sum; /* firm[6][i]: market size after shared */
                                             /* firm[12][i]: product A can be sold */

   /*-------- for product B ----------*/
   for(i=0;i<firm_no;i++)
   {
   a=firm[43][i]*B_demand;              /* firm[2][i]:FI */
   b=(1-B_ff)*B_demand*(1-firm[43][i]);
        firm[14][i]=a+b;              /* B market share for each firm */

        /*----- calculate the weights ----------*/
        total[i]=firm[16][i]*price;          /* firm[16][i]: product B price index */
        total[i]+=firm[4][i]*service_quality; /* firm[4][i]:service quality index */
        total[i]+=firm[42][i]*product_quality; /* firm[42][i]: B stock quality level */
        total[i]+=firm[5][i]*salesforce;      /* firm[5][i]: SFI */
   }
   sum=0;
   for(i=0;i<firm_no;i++)
        sum+=total[i];

   for(i=0;i<firm_no;i++)
        firm[13][i]=firm[14][i]*total[i]/sum; /* firm[14][i]: market size after shared */
                                              /* firm[13][i]: product B can be sold */

}

update_cash()
{                       /* firm[0][i] : cash on hand */
   int i;

   printf("Do you want the competitor profile ? ");
   if(toupper(getche())!='Y')
         profile=NO;
        else
   {
              profile=YES;
              for(i=0;i<firm_no;i++)
                     firm[0][i]-=profile_cost;  /* cost for profile */
   }

   for(i=0;i<firm_no;i++)
   {
        /* fund transferred in & out are calculated in userinput() */
        /* initial capital are added in userinput() */
```

```
pay_over[i]=overdraft[i]*(1+overdraft_interest); /* overdraft in previous period */
firm[0][i]-=pay_over[i];
firm[0][i]+=loan[i];            /* cash+ by approved loan */
firm[0][i]-=fire_man_cost[i];   /* cash- by firing */
firm[0][i]-=hire_man_cost[i];   /* cash- by hiring */
firm[0][i]-=open_cost[i];       /* cash- by open new branch */
firm[0][i]-=close_cost[i];      /* cash- by close branch */
firm[0][i]-=firm[18][i]*firm[17][i]*wages; /* cash- by salespeople wages */
firm[0][i]-=rent_rate*firm[17][i];   /* cash- by rent rate */
                    /* firm[12][i]:product A can be sold */
                    /* firm[13][i]:product B can be sold */
                    /* firm[1][i]: product A opening stock */
                    /* firm[9][i]: product B opening stock */
                    /* firm[22][i]: product A order quantity */
                    /* firm[26][i]: product B order quantity */
                    /* firm[7][i]: product A selling price */
                    /* firm[15][i]: product B selling price */
                    /* firm[29][i]  : A closing stock */
                    /* firm[30][i]  : B closing stock */
                    /* firm[31][i]  : sales of product A */
                    /* firm[32][i]  : sales of product B */
if((firm[1][i]+firm[22][i])>firm[12][i])
{
   firm[31][i]=firm[12][i];   /* sales of product A */
   firm[0][i]+=firm[31][i]*firm[7][i]; /* cash+ by selling product A */
   firm[29][i]=firm[1][i]+firm[22][i]-firm[12][i]; /* product A closing stock */
}
else
{
   firm[31][i]=firm[1][i]+firm[22][i]; /* sales of product A */
   firm[0][i]+=firm[31][i]*firm[7][i]; /* cash+ by selling */
   firm[29][i]=0;    /* product A closing stock */
}

if((firm[9][i]+firm[26][i])>firm[13][i])
{
   firm[32][i]=firm[13][i];      /* sales of product B */
   firm[0][i]+=firm[32][i]*firm[15][i]; /* cash+ by selling product B */
   firm[30][i]=firm[9][i]+firm[26][i]-firm[13][i]; /* product B closing stock */
}
else
{
   firm[32][i]=firm[9][i]+firm[26][i]; /* sales of product B */
   firm[0][i]+=firm[32][i]*firm[15][i]; /* cash+ by selling */
   firm[30][i]=0;    /* product B closing stock */
}
firm[0][i]-=firm[3][i];   /* cash- by product A advertising */
firm[0][i]-=firm[11][i];  /* cash- by product B advertising */
firm[0][i]-=firm[21][i]*firm[22][i];   /* cash- by buy product A */
firm[0][i]-=firm[25][i]*firm[26][i];   /* cash- by buy product B */
          firm[0][i]-=firm[19][i]*(firm[31][i]+firm[32][i]);  /* cash- by commission */

if(firm[0][i]>0)
   overdraft[i]=0;
else
{
                overdraft[i]=firm[0][i]*(-1);
                firm[0][i]=0;
```

```
        }
    }
}

report()
{
        int i,j;
    FILE *out;
    char filename[8];
        char temp_name[8];
        char temp_ext[3];
        char ext[3]=".";
        char no[2];
        float sum_sales,A_cost,B_cost,gross,carry_product_cost,total_wages;
        float commis,expense,market;
        float cur_asset,remain;
        float com_data[24];     /* for the combined output */

        printf("\nInput a file name for the output file : ");
        scanf("%s",filename);
        strcpy(temp_name,filename);

        for(i=0;i<firm_no;i++)
        {
                strcpy(temp_ext,ext);
                strcpy(temp_name,filename);
                itoa(1+i,no,10);
                strcat(temp_ext,no);
                if((out=fopen(strcat(temp_name,temp_ext),"w"))==NULL)
                        printf("<<ERROR : Output file open error !>>\n");
                else
                {           /*---------- trading & profit & loss ---------*/
                        fprintf(out,"\Marketing's Trading & Profit & Loss A/C for the Year
Ended of period %d (firm #%d)",period,i+1);
          fprintf(out,"\n   Sales of Product A
%11.0f",firm[31][i]*firm[7][i]);
          fprintf(out,"\n   Sales of Product B
%11.0f",firm[32][i]*firm[15][i]);

                        sum_sales=firm[31][i]*firm[7][i]+firm[32][i]*firm[15][i];
                        com_data[5]=sum_sales; /* for combine report*/
          fprintf(out,"\n                                          ----------- %11.0f",sum_sales);
          fprintf(out,"\n   Less Cost of Good Sold:");
          fprintf(out,"\nProduct A:  Opening Stock
%11.0f",firm[1][i]*firm[34][i]);
                        com_data[6]=firm[1][i]*firm[34][i]+(firm[9][i]*firm[35][i]); /* for
combine report */
          fprintf(out,"\n        +Purchase or Transferred(from Production)
%11.0f",firm[21][i]*firm[22][i]);
                        com_data[7]=firm[21][i]*firm[22][i]+(firm[25][i]*firm[26][i]); /* for
combine report*/
          fprintf(out,"\n        -Closing Product A
%11.0f",firm[29][i]*firm[38][i]);
                        com_data[8]=firm[29][i]*firm[38][i]+(firm[30][i]*firm[39][i]); /* for
combine report */
                        A_cost=firm[1][i]*firm[34][i]+firm[21][i]*firm[22][i]-
firm[29][i]*firm[38][i];
          fprintf(out,"\n                                     ----------- %11.0f\n",A_cost);
```

```
        fprintf(out,"\nProduct B:  Opening Stock
%11.0f",firm[9][i]*firm[35][i]);
        fprintf(out,"\n        +Purchase or Transferred(from Production)
%11.0f",firm[25][i]*firm[26][i]);
        fprintf(out,"\n        -Closing Product B
%11.0f",firm[30][i]*firm[39][i]);
                B_cost=firm[9][i]*firm[35][i]+firm[25][i]*firm[26][i]-
firm[30][i]*firm[39][i];
                com_data[9]=A_cost+B_cost; /* for combine rport */
        fprintf(out,"\n                                   ----------- %11.0f
%11.0f",B_cost,A_cost+B_cost);
        fprintf(out,"\n                                       ----------- -----------");

                gross=sum_sales-A_cost-B_cost;
        fprintf(out,"\nGross Profit                                        %11.0f",gross);

        fprintf(out,"\n   Less Expenses:");
        fprintf(out,"\n     Advertising
%11.0f",firm[3][i]+firm[11][i]);
                com_data[16]=firm[3][i]+firm[11][i]; /* for combine report */
       fprintf(out,"\n     Cost for set up branches                 %11.0f",open_cost[i]);
                com_data[17]=open_cost[i]; /* for combine report */
        fprintf(out,"\n     Cost for closing branches
%11.0f",close_cost[i]);
                com_data[18]=close_cost[i]; /* for combine report */
        fprintf(out,"\n     Cost for hiring salespeople
%11.0f",hire_man_cost[i]);
                com_data[12]=hire_man_cost[i]; /* for combine report */
        fprintf(out,"\n     Cost for firing salespeople
%11.0f",fire_man_cost[i]);
                com_data[11]=fire_man_cost[i]; /* for combine report */

       carry_product_cost=firm[1][i]*A_product_carry_cost+firm[9][i]*B_product_carry_cost;
                com_data[10]=carry_product_cost; /* for combine report */
        fprintf(out,"\n     Carrying cost of Finished product
%11.0f",carry_product_cost);
        fprintf(out,"\n     paid for loan
%11.0f",(pay_loan[i]+(loan[i]*loan_interest)));
        com_data[22]=pay_loan[i]+(loan[i]*loan_interest); /* for combine report */
                com_data[21]=pay_over[i]; /* for combine report */
        fprintf(out,"\n     Paid for overdraft                       %11.0f",pay_over[i]);
                total_wages=firm[17][i]*firm[18][i]*wages;
                com_data[14]=total_wages; /* for combine eport */
        fprintf(out,"\n     Wages of salespeople                         %11.0f",total_wages);
                commis=firm[19][i]*(firm[31][i]+firm[32][i]);
                com_data[15]=commis; /* for combine report */
        fprintf(out,"\n     Commission of salespeople                    %11.0f",commis);
        fprintf(out,"\n     Overhead of all branches
%11.0f",branch_overhead*firm[17][i]);
        com_data[13]=rent_rate*firm[17][i];  /* for combine reprot */
                com_data[23]=branch_overhead*firm[17][i];

       expense=firm[3][i]+firm[11][i]+open_cost[i]+close_cost[i]+hire_man_cost[i];

       expense+=fire_man_cost[i]+carry_product_cost+loan[i]*loan[i]*loan_interest;
```

```
        expense+=(pay_over[i]+(loan[i]*loan_interest))+total_wages+commis+(branch_overhea
d+rent_rate)*firm[17][i];
          fprintf(out,"\n      Rent and Rates                              %11.0f
%11.0f",rent_rate*firm[17][i],expense);
          fprintf(out,"\n                                         ------------- -------------");

                    firm[40][i]=gross-expense;
          fprintf(out,"\n      Net profit                                          %11.0f",firm[40][i]);

                    /*---------- Marketing's Balance sheet ----------*/
          fprintf(out,"\n\n      Marketing's Balance Sheet As at period %d(firm
#%d)",period,i+1);
                    fprintf(out,"\nCurrent Assets");
          fprintf(out,"\nProduct A                    %11.0f",firm[29][i]*firm[38][i]);
                    com_data[0]=firm[29][i]*firm[38][i];      /* for the combined data */
          fprintf(out,"\nProduct B                    %11.0f",firm[30][i]*firm[39][i]);
                    com_data[0]+=firm[30][i]*firm[39][i];     /* for the combined data */
          fprintf(out,"\nCash on hand                   %11.0f",firm[0][i]);
                    com_data[1]=firm[0][i];      /* for the combined data */
          fprintf(out,"\n                         ----------");
                    cur_asset=firm[29][i]*firm[38][i]+firm[30][i]*firm[39][i]+firm[0][i];
          fprintf(out,"\n                         %11.0f",cur_asset);

          fprintf(out,"\nOver Draft             %11.0f",overdraft[i]);
                    com_data[2]=overdraft[i];      /* for the combined data */
                    com_data[3]=loan[i];         /* for the combined data */
          fprintf(out,"\nLoan                   %11.0f %11.0f",loan[i],loan[i]+overdraft[i]);
          fprintf(out,"\n                       ---------- ----------");
                    remain=cur_asset-loan[i]-overdraft[i];
          fprintf(out,"\n                         %11.0f",remain);

                    fprintf(out,"\nFinanced By");
          fprintf(out,"\nCapital                      %11.0f",remain-firm[40][i]);
          fprintf(out,"\nNet profit                    %11.0f",firm[40][i]);
                    com_data[4]=firm[40][i];       /* for the combined data */
                    com_data[20]=firm[40][i];       /* for the combined data */

        com_data[19]=com_data[23]+com_data[21]+com_data[22]+com_data[10]+com_data[11
]+com_data[18];

        com_data[19]+=com_data[12]+com_data[13]+com_data[14]+com_data[15]+com_data[1
6]+com_data[17];
          fprintf(out,"\n                         %11.0f",remain);
                    fclose(out);

              }
          /*--------- for the combined report ---------*/
        strcat(temp_name,"d");        /* output file XXXXXXX.Xd */
        if((out=fopen(temp_name,"w"))==NULL)
           printf("<<ERROR : Output file open error !>>\n");
        else
              {     for (j=0;j<24;j++)
                                        fprintf(out,"%f \n",com_data[j]);  /* write the data to
 combine report */
                              fclose(out);
              }
```

```
        }

/*------------- same file for all the firms ---------------*/
if ((out=fopen(filename,"w"))==NULL)
        printf("<<ERROR : Output file open error !>>\n");
else
{
        /*------ Market summary ------ for product A --------------*/
        fprintf(out,"\nFor product A:\n");
fprintf(out,"\nFirm no.     ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"       firm %d",i+1);

fprintf(out,"\nPrice         ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.2f",firm[7][i]);

fprintf(out,"\nQuantity sold  ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.0f",firm[31][i]);

fprintf(out,"\nMarket share(%)");
        for(i=0;i<firm_no;i++)
                {
          market=(firm[31][i]/A_demand)*100;
                        fprintf(out,"    %11.0f",market);
                }
fprintf(out,"\nRevenue       ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.0f",firm[31][i]*firm[7][i]);

        /*---- Market summary -------- for product B ----------*/
        fprintf(out,"\n\nFor product B:\n");
fprintf(out,"\nFirm no.     ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"       firm %d",i+1);

fprintf(out,"\nPrice         ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.2f",firm[15][i]);

fprintf(out,"\nQuantity sold  ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.0f",firm[32][i]);

fprintf(out,"\nMarket share(%)");
        for(i=0;i<firm_no;i++)
                {
          market=(firm[32][i]/B_demand)*100;
                        fprintf(out,"    %11.0f",market);
                }
fprintf(out,"\nRevenue       ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"    %11.0f",firm[32][i]*firm[15][i]);

        /*--------- salesforce ----------------*/

fprintf(out,"\n\nFirm no.          ");
```

```
        for(i=0;i<firm_no;i++)
        fprintf(out,"    firm %d",i+1);

    fprintf(out,"\nNo.of salespeople  ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"%11.0f ",firm[18][i]*firm[17][i]);

    fprintf(out,"\nNo.of branches     ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"%11.0f ",firm[17][i]);

    fprintf(out,"\nCommission rate(%) ");
        for(i=0;i<firm_no;i++)
        fprintf(out,"%11.0f ",(firm[19][i]*100));

        /*-------------- profile -----------------*/
                if(profile)
                {
          fprintf(out,"\n\nFirm no.        ");
                        for(i=0;i<firm_no;i++)
        fprintf(out,"    firm %d",i+1);

                        fprintf(out,"\nFamiliarity (A)    ");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.2f",firm[2][i]);

                        fprintf(out,"\nFamiliarity (B)    ");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.2f",firm[10][i]);

        fprintf(out,"\nNon-selling time(%)");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.0f",(firm[20][i]*100));

                        fprintf(out,"\nAdvertising amount ");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.0f",firm[3][i]+firm[11][i]);

                        fprintf(out,"\nProduct quality (A)");
                        for(i=0;i<firm_no;i++)
                                fprintf(out," %11.2f",firm[41][i]);

                        fprintf(out,"\nProduct quality (B)");
                        for(i=0;i<firm_no;i++)
                                fprintf(out," %11.2f",firm[42][i]);

                        fprintf(out,"\nService quality    ");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.2f",firm[4][i]);

                        fprintf(out,"\nNet profit         ");
                        for(i=0;i<firm_no;i++)
                fprintf(out," %11.0f",firm[40][i]);
                }
        fclose(out);
        }

}
```

```
update_env()
{
        int i;
  A_demand=A_demand*(1+A_growth);   /* calculate the next market size */
  B_demand=B_demand*(1+B_growth);   /* calculate the next market size */
        for(i=0;i<firm_no;i++)
        {
                firm[1][i]=firm[29][i];     /* A open stock == close stock in last period */
                firm[9][i]=firm[30][i];     /* B open stock == close stock in last period */

                new[5][i]=new[0][i];        /* branch opened in last period */
        new[4][i]=new[2][i];         /* salespeople hired in last period */

        firm[34][i]=firm[38][i];  /* A open stock unit cost = exist unit cost in last period */
        firm[35][i]=firm[39][i];  /* B open stock unit cost = exist unit cost in last period */

        firm[24][i]=firm[31][i];  /* A sales should be serviced in next period */
        firm[28][i]=firm[32][i];  /* B sales should be serviced in next period */
  }
}

readhis()
{
        int i;
  FILE *hisin;
        char filename[11],string[50];
  clrscr();

  printf("\nInput the history file name : ");
        scanf("%s",filename);

        strcat(filename,".txt");
                if((hisin=fopen(filename,"r"))==NULL)
          printf("<<ERROR : Input file open error !>>\n");
        else
  {
        for(i=0;i<6;i++)
          fscanf(hisin,"%s %f",string,&firm[0][i]); /* capital as the initial cash on hand */
        for(i=0;i<6;i++)
                        fscanf(hisin,"%s %f",string,&firm[17][i]);
        for(i=0;i<6;i++)
                        fscanf(hisin,"%s %f",string,&firm[18][i]);
        fscanf(hisin,"%s %ld",string,&A_demand);
        fscanf(hisin,"%s %f",string,&A_growth);
        fscanf(hisin,"%s %ld",string,&B_demand);
        fscanf(hisin,"%s %f",string,&B_growth);
        /* for firms hisinmeters */
        for(i=0;i<6;i++)             /* get product A sold last period */
          fscanf(hisin,"%s %f",string,&firm[24][i]);
        for(i=0;i<6;i++)             /* get product A opening stock */
          fscanf(hisin,"%s %f",string,&firm[1][i]);
        for(i=0;i<6;i++)             /* get product A stock unit cost */
          fscanf(hisin,"%s %f",string,&firm[34][i]);
        for(i=0;i<6;i++)             /* get product A stock quality level */
          {
          fscanf(hisin,"%s %f",string,&firm[36][i]);
          firm[36][i]=firm[36][i]/1000;
```

```
            }
        for(i=0;i<6;i++)            /* get product A familiarity index */
            fscanf(hisin,"%s %f",string,&firm[2][i]);
        for(i=0;i<6;i++)            /* get product B sold last period */
            fscanf(hisin,"%s %f",string,&firm[28][i]);
        for(i=0;i<6;i++)            /* get product B opening stock */
            fscanf(hisin,"%s %f",string,&firm[9][i]);
        for(i=0;i<6;i++)            /* get product B stock unit cost */
            fscanf(hisin,"%s %f",string,&firm[35][i]);
        for(i=0;i<6;i++)            /* get product B stock quality level */
            {
            fscanf(hisin,"%s %f",string,&firm[37][i]);
            firm[37][i]=firm[37][i]/1000;
            }
        for(i=0;i<6;i++)            /* get product B familiarity index */
            fscanf(hisin,"%s %f",string,&firm[10][i]);
        for(i=0;i<6;i++)
            fscanf(hisin,"%s %f",string,&loan[i]);
        for(i=0;i<6;i++)
            fscanf(hisin,"%s %f",string,&overdraft[i]);
        for(i=0;i<6;i++)
            fscanf(hisin,"%s %d",string,&new[5][i]);
        for(i=0;i<6;i++)
            fscanf(hisin,"%s %d",string,&new[4][i]);

    fclose(hisin);
    }
}
writehis()
{

        int i;
    FILE *hisout;
        char filename[11];
    clrscr();
        printf("\nInput the new history file name : ");
        scanf("%s",filename);

        strcat(filename,".txt");
                if((hisout=fopen(filename,"w"))==NULL)
                        printf("<<ERROR : Output file open error !>>\n");
        else
    {
        for(i=0;i<6;i++)
                        fprintf(hisout,"initial_capital_firm_%d %f\n",i,firm[0][i]); /* capital as
the initial cash on hand */
        for(i=0;i<6;i++)
            fprintf(hisout,"no_of_branch_firm_%d %.0f\n",i,firm[17][i]);
        for(i=0;i<6;i++)
            fprintf(hisout,"no_of_salepeople_firm_%d %.0f\n",i,firm[18][i]);
                fprintf(hisout,"A_basic_demand %ld\n",A_demand);
                fprintf(hisout,"A_market_growth %f\n",A_growth);
                fprintf(hisout,"B_basic_demand %ld\n",B_demand);
                fprintf(hisout,"B_market_growth %f\n",B_growth);
        /* for firms hisoutmeters */
        for(i=0;i<6;i++)            /* put product A sold last period */
            fprintf(hisout,"product_A_sold_last_period_firm_%d %.0f\n",i,firm[24][i]);
        for(i=0;i<6;i++)            /* put product A opening stock */
```

```
            fprintf(hisout,"product_A_opening_stock_firm_%d %.0f\n",i,firm[1][i]);
        for(i=0;i<6;i++)            /* put product A stock unit cost */
                        fprintf(hisout,"product_A_stock_unit_cost_firm_%d
%f\n",i,firm[34][i]);
        for(i=0;i<6;i++)            /* put product A stock quality level */
                        fprintf(hisout,"product_A_quality_level_firm_%d %f\n",i,firm[36][i]);
        for(i=0;i<6;i++)            /* put product A familiarity index */
                        fprintf(hisout,"product_A_familiayity_index_firm_%d
%f\n",i,firm[2][i]);
        for(i=0;i<6;i++)            /* put product B sold last period */
            fprintf(hisout,"product_B_sold_last_period_firm_%d %.0f\n",i,firm[28][i]);
        for(i=0;i<6;i++)            /* put product B opening stock */
            fprintf(hisout,"product_B_opening_stock_firm_%d %.0f\n",i,firm[9][i]);
        for(i=0;i<6;i++)            /* put product B stock unit cost */
                        fprintf(hisout,"product_B_stock_unit_cost_firm_%d
%f\n",i,firm[35][i]);
        for(i=0;i<6;i++)            /* put product B stock quality level */
                        fprintf(hisout,"product_B_stock_quality_level_firm_%d
%f\n",i,firm[37][i]);
        for(i=0;i<6;i++)            /* put product B familiarity index */
                        fprintf(hisout,"product_B_familiarty_index_firm_%d
%f\n",i,firm[10][i]);
        for(i=0;i<6;i++)
            fprintf(hisout,"Loan_of_firm_%d %f\n",i,loan[i]);
        for(i=0;i<6;i++)
            fprintf(hisout,"Overdraft_of_firm_%d %f\n",i,overdraft[i]);
        for(i=0;i<6;i++)
            fprintf(hisout,"Branch_open_of_firm_%d %d\n",i,new[5][i]);
        for(i=0;i<6;i++)
                        fprintf(hisout,"New_salepeople_hire_of_firm_%d %d\n",i,new[4][i]);
  }
fclose(hisout);
}
```

For product A:

Firm no.	firm 1	firm 2	firm 3	firm 4
Price	200.00	230.00	190.00	180.00
Quantity sold	114590	99507	110812	99607
Market share(%)	23	20	22	20
Revenue	22918098	22886560	21054372	17929195

For product B:

Firm no.	firm 1	firm 2	firm 3	firm 4
Price	250.00	270.00	240.00	230.00
Quantity sold	138007	123475	133154	120517
Market share(%)	23	21	22	20
Revenue	34501832	33338296	31956878	27718892

Firm no.	firm 1	firm 2	firm 3	firm 4
No.of salespeople	15	15	6	15
No.of branches	3	3	2	3
Commission rate(%)	8	15	6	10

Firm no.	firm 1	firm 2	firm 3	firm 4
Familiarity (A)	0.85	0.91	0.98	0.99
Familiarity (B)	0.72	0.83	0.96	0.99
Non-selling time(%)	10	15	12	10
Advertising amount	200000	400000	1000000	2000000
Product quality (A)	0.97	0.97	0.98	0.98
Product quality (B)	0.96	0.97	0.97	0.97
Service quality	0.61	0.61	0.60	0.61
Net profit	40657260	-147064	47080656	37327616

Appendix XXIV

Marketing's Trading & Profit & Loss A/C for the Year Ended of period 1 (firm #1)

Sales of Product A			22918098	
Sales of Product B			34501832	
				57419932
Less Cost of Good Sold:				
Product A:	Opening Stock	200000		
	+Purchase or Transferred(from Production)	1557500		
	-Closing Product A	414881		
			1342619	
Product B:	Opening Stock	750000		
	+Purchase or Transferred(from Production)	2417000		
	-Closing Product B	981654		
			2185346	3527965
Gross Profit				53891968
Less Expenses:				
	Advertising		200000	
	Cost for set up branches		0	
	Cost for closing branches		0	
	Cost for hiring salespeople		0	
	Cost for firing salespeople		0	
	Carrying cost of Finished product		250000	
	paid for loan		25500	
	Paid for overdraft		0	
	Wages of salespeople		75000	
	Commission of salespeople		20208	
	Overhead of all branches		9000	
	Rent and Rates		180000	13234708
	Net profit			40657260

Marketing's Balance Sheet As at period 1(firm #1)

Current Assets		
Product A		414881
Product B		981654
Cash on hand		54464224
		55860760
Over Draft	0	
Loan	25000	25000
		55835760
Financed By		
Capital		15178500
Net profit		40657260
		55835760

Marketing's Trading & Profit & Loss A/C for the Year Ended of period 1 (firm #2)

Sales of Product A			22886560	
Sales of Product B			33338296	
			-----------	56224856
Less Cost of Good Sold:				
Product A:	Opening Stock	200000		
	+Purchase or Transferred(from Production)	4000000		
	-Closing Product A	2110358		
		-----------	2089642	
Product B:	Opening Stock	750000		
	+Purchase or Transferred(from Production)	6000000		
	-Closing Product B	3416170		
		-----------	3333830	5423472
			-----------	-----------
Gross Profit				50801384
Less Expenses:				
	Advertising		400000	
	Cost for set up branches		0	
	Cost for closing branches		0	
	Cost for hiring salespeople		0	
	Cost for firing salespeople		0	
	Carrying cost of Finished product		250000	
	paid for loan		51000	
	Paid for overdraft		0	
	Wages of salespeople		75000	
	Commission of salespeople		33447	
	Overhead of all branches		9000	
	Rent and Rates		180000	50948448
			-------------	-------------
	Net profit			-147064

Marketing's Balance Sheet As at period 1(firm #2)

Current Assets		
Product A		2110358
Product B		3416170
Cash on hand		47029408

		52555936
Over Draft	0	
Loan	50000	50000
	----------	----------
		52505936
Financed By		
Capital		52653000
Net profit		-147064
		52505936

Appendix XXIV

Marketing's Trading & Profit & Loss A/C for the Year Ended of period 1 (firm #3)

Sales of Product A		21054372	
Sales of Product B		31956878	
		-----------	53011248
Less Cost of Good Sold:			
Product A: Opening Stock	200000		
+Purchase or Transferred(from Production)	3600000		
-Closing Product A	1885966		
	-----------	1914034	
Product B: Opening Stock	750000		
+Purchase or Transferred(from Production)	4000000		
-Closing Product B	2220081		
	-----------	2529920	4443953
		-----------	-----------
Gross Profit			48567296
Less Expenses:			
Advertising		1000000	
Cost for set up branches		0	
Cost for closing branches		30000	
Cost for hiring salespeople		0	
Cost for firing salespeople		36000	
Carrying cost of Finished product		250000	
paid for loan		0	
Paid for overdraft		0	
Wages of salespeople		30000	
Commission of salespeople		14638	
Overhead of all branches		6000	
Rent and Rates		120000	1486638
		-------------	-------------
Net profit			47080656

Marketing's Balance Sheet As at period 1(firm #3)

Current Assets		
Product A		1885966
Product B		2220081
Cash on hand		45675612

		49781660
Over Draft	0	
Loan	0	0
	----------	----------
		49781660
Financed By		
Capital		2701004
Net profit		47080656
		49781660

Marketing's Trading & Profit & Loss A/C for the Year Ended of period 1 (firm #4)

Sales of Product A		17929195		
Sales of Product B		27718892		
		-----------	45648088	
Less Cost of Good Sold:				
Product A: Opening Stock	200000			
+Purchase or Transferred(from Production)	6000000			
-Closing Product A	3729755			
	-----------	2470245		
Product B: Opening Stock	750000			
+Purchase or Transferred(from Production)	7500000			
-Closing Product B	4935785			
	-----------	3314215	5784460	
		-----------	-----------	
Gross Profit			39863628	
Less Expenses:				
Advertising		2000000		
Cost for set up branches		0		
Cost for closing branches		0		
Cost for hiring salespeople		0		
Cost for firing salespeople		0		
Carrying cost of Finished product		250000		
paid for loan		0		
Paid for overdraft		0		
Wages of salespeople		75000		
Commission of salespeople		22012		
Overhead of all branches		9000		
Rent and Rates		180000	2536012	
		-------------	-------------	
Net profit			37327616	

Marketing's Balance Sheet As at period 1(firm #4)

Current Assets		
Product A		3729755
Product B		4935785
Cash on hand		31366076

		40031616
Over Draft	0	
Loan	0	0
	----------	----------
		40031616
Financed By		
Capital		2704000
Net profit		37327616
		40031616

Combine report - program overview

Source file : combine.c
Execution file : combine.exe

Function : This program is used to merge the reports from production and marketing into a combined report. It using the data files with 'd' in the extension after running the other program.

Input file : 'XXXX.d' - data file from production
: 'XXXX.nd' - data file from marketing (where 'n' is the firm number)
Input parameter : period number of the data

Output file : report file 'XXXX' (where XXXX is user defined)

Report file type :

1. 'Production & Marketing Combined Balance Sheet As At Period XX'

2. 'Production & Marketing Combined Trading & Profit & Loss A/C for the Year Ended of Period XX'

Combine report - User's manual

User manual of combine.exe

1. Input the production output data file name : 'XXXX.d' - data file from production

2. Input the marketing output data file name : 'XXXX.nd' - data file from marketing (where 'n' is the firm number)

3. Input the combined output file name : 'XXXX' - XXXX is user defined output file name

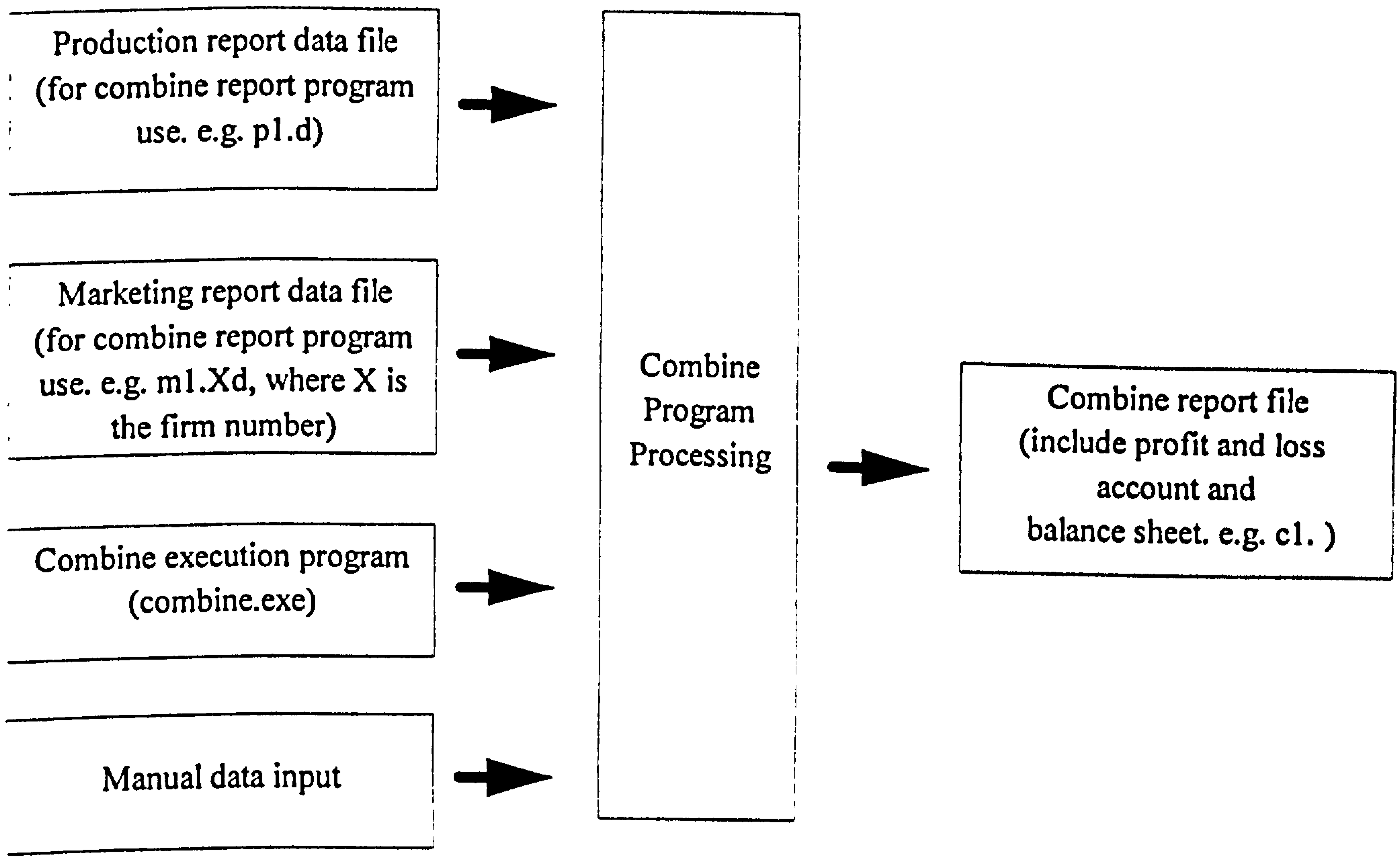
4. Input the output period : 'XX' - period number printed in the combined report

The combined report included the following:-

1. 'Production & Marketing Combined Balance Sheet As At Period XX'

2. 'Production & Marketing Combined Trading & Profit & Loss A/C for the Year Ended of Period XX'

Combine program data flow chart



Combine report - Program listing

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
main()
{
        /* program for printing the combined report */
        int i;
  char string[50];
  char out_filename[12];
  char in_file_a[12];
  char in_file_b[12];
        float a_data[32]; /* production data */
        float b_data[24]; /* marketing data */
  int period=0;
  float temp1, temp2, temp3;
  FILE *out;


  clrscr();
  printf("\nInput the production output data file name :");
  scanf("%s",in_file_a);
  printf("\nInput the marketing output data file name :");
  scanf("%s",in_file_b);
  printf("\nInput the combined output file name :");
  scanf("%s",out_filename);
  printf("\nInput the output period :");
  scanf("%d",&period);

  /*------------- read the production file ---------*/
  if((out=fopen(in_file_a,"r"))==NULL)
        printf("<<ERROR : Output file open error !>>\n");
  else
                for(i=0;i<32;i++)
                        fscanf(out,"%f",&a_data[i]);

  /*----------- read the marketing file -----------*/
  if((out=fopen(in_file_b,"r"))==NULL)
        printf("<<ERROR : Output file open error !>>\n");
  else
                for(i=0;i<24;i++)
                        fscanf(out,"%f",&b_data[i]);

  /*----------- write the output file -----------*/
  if((out=fopen(out_filename,"w"))==NULL)
        printf("<<ERROR : Output file open error !>>\n");
  else
        {
                /* for production & marketing combine Sheet*/

  fprintf(out,"\n            Production & Marketing");
  fprintf(out,"\n      Combined Balance Sheet at period %d\n",period);
  fprintf(out,"\nFixed Assets :\n\n");
  fprintf(out,"Premise                                        %10.0f\n",a_data[0]);
```

```
fprintf(out,"\nMachinery A & B                %10.0f\n",a_data[1]);
fprintf(out,"\nLess: Depreciation            %10.0f\n",a_data[2]);
fprintf(out,"                  --------            %10.0f\n",a_data[1]-a_data[2]);
fprintf(out,"                                      -----------\n");
      temp1=a_data[0]+a_data[1]-a_data[2];
fprintf(out,"                                 %10.0f\n",temp1);

      fprintf(out,"\nCurrent Assets :\n");
fprintf(out,"\nRaw Material of Product A & B      %10.0f",a_data[3]);
fprintf(out,"\nFinished Products of A & B       %10.0f",a_data[4]+b_data[0]);
fprintf(out,"\nCash On Hand                  %10.0f",a_data[5]+b_data[1]);
      temp2=a_data[3]+a_data[4]+b_data[0]+a_data[5]+b_data[1];
fprintf(out,"\n                    ----------");
fprintf(out,"\n                    %10.0f",temp2);

      fprintf(out,"\nLess: Current Liability:\n");
fprintf(out,"\nOver Draft                  %10.0f",a_data[6]+b_data[2]);
fprintf(out,"\nLoan                      %10.0f\n",a_data[7]+b_data[3]);
      temp3=a_data[6]+a_data[7]+b_data[2]+b_data[3];
fprintf(out,"                 -------              %10.0f",temp3);
fprintf(out,"\n                                     -----------");
fprintf(out,"\n                                    %10.0f\n",temp1+temp2-temp3);


      fprintf(out,"\nFinanced by\n");

fprintf(out,"\nCapital                                      %10.0f",temp1+temp2-
temp3+a_data[8]+b_data[4]);
fprintf(out,"\nNet Profit                                   %10.0f",a_data[8]+b_data[4]);
fprintf(out,"\n                                            ---------");
fprintf(out,"\n                                                  %10.0f",temp1+temp2-
temp3);

            /* for production & marketing combineed trading & profit & loss A/C */

fprintf(out,"\n          Production & Marketing");
fprintf(out,"\n     Combined Trading & Profit & loss A/C for the year Ended of ");
fprintf(out,"\n                    period %d\n",period);
fprintf(out,"\n                         Production $      Marketing $        Combined $");
fprintf(out,"\nSales                          %10.0f        %10.0f
%10.0f",a_data[9],b_data[5],a_data[9]+b_data[5]);
      fprintf(out,"\n\nLess Cost of Good Sold:\n");
fprintf(out,"\nOpening Stock                 %10.0f        %10.0f
%10.0f",a_data[10],b_data[6],a_data[10]+b_data[6]);
fprintf(out,"\n+ Purchase                  %10.0f        %10.0f
%10.0f",a_data[11],b_data[6],a_data[11]+b_data[6]);
            temp1=a_data[10]+a_data[11]-a_data[13];
            temp2=b_data[6]+b_data[7]-b_data[8];
            temp3=temp1+temp2;
fprintf(out,"\n- Closing Stock               %10.0f %10.0f %10.0f %10.0f %10.0f
%10.0f",a_data[13],temp1,b_data[8],temp2,a_data[13]+b_data[8],temp3);
fprintf(out,"\n                    -------  -------  -------  -------  -------  --------");
fprintf(out,"
%10.0f",a_data[9]+b_data[5]-temp3);
      fprintf(out,"\n\Manufacturing Overhead:\n");
fprintf(out,"\nMachinery Overhead              %10.0f
%10.0f",a_data[14],a_data[14]);
```

```
   fprintf(out,"\nWages (Indirect)              %10.0f
%10.0f",a_data[15],a_data[15]);
   fprintf(out,"\nInspection Cost               %10.0f
%10.0f",a_data[16],a_data[16]);
   fprintf(out,"\n\nFactory cost of products producted:         %10.0f
%10.0f",a_data[17],a_data[17]);
                  temp1=b_data[5]-temp2;
         temp2=a_data[9]+b_data[5]-temp3;
         temp3=a_data[18]+temp1;
   fprintf(out,"\n\n                        -------                    -------");
   fprintf(out,"\nGross profit                  %10.0f        %10.0f
%10.0f",a_data[18],temp1,temp3);
    fprintf(out,"\nLess Expense:\n");
   fprintf(out,"\nDepreciation of Machinery     %10.0f
%10.0f",a_data[19],a_data[19]);
   fprintf(out,"\nMaintainance Cost             %10.0f
%10.0f",a_data[20],a_data[20]);
   fprintf(out,"\nRepairing Cost                %10.0f
%10.0f",a_data[21],a_data[21]);
   fprintf(out,"\nGeneral Expenses              %10.0f        %10.0f
%10.0f",a_data[23],b_data[23],a_data[23]+b_data[23]);
   fprintf(out,"\nPaid for Over Draft           %10.0f        %10.0f
%10.0f",a_data[24],b_data[21],a_data[24]+b_data[21]);
   fprintf(out,"\nInterest Paid for load        %10.0f        %10.0f
%10.0f",a_data[25],b_data[22],a_data[25]+b_data[22]);
   fprintf(out,"\nCarrying Cost of Raw Material     %10.0f
%10.0f",a_data[26],a_data[26]);
   fprintf(out,"\nCarrying Cost of Finished Product  %10.0f        %10.0f
%10.0f",a_data[27],b_data[10],a_data[27]+b_data[10]);
   fprintf(out,"\nFiring Cost                   %10.0f        %10.0f
%10.0f",a_data[28],b_data[11],a_data[28]+b_data[11]);
   fprintf(out,"\nHiring Cost                   %10.0f        %10.0f
%10.0f",a_data[29],b_data[12],a_data[29]+b_data[12]);
   fprintf(out,"\nRent & Rates                  %10.0f        %10.0f
%10.0f",a_data[30],b_data[13],a_data[30]+b_data[13]);
   fprintf(out,"\nWages                                 %10.0f
%10.0f",b_data[14],b_data[14]);
   fprintf(out,"\nCommisson of Salespeople                     %10.0f
%10.0f",b_data[15],b_data[15]);
   fprintf(out,"\nAdvertising                            %10.0f
%10.0f",b_data[16],b_data[16]);
   fprintf(out,"\nCost for set up Branches                     %10.0f
%10.0f",b_data[17],b_data[17]);
   fprintf(out,"\nCost for closing Branches             %10.0f%10.0f%10.0f%10.0f
%10.0f",a_data[22],b_data[18],b_data[19],b_data[18],a_data[22]+b_data[19]);
   fprintf(out,"\n                 -------  -------  -------  -------  --------  --------");
   fprintf(out,"\n\n Net Profit                    %10.0f        %10.0f
%10.0f",a_data[31],b_data[20],a_data[31]+b_data[20]);


   }
}
```

Production & Marketing
Combined Balance Sheet at period 1

Fixed Assets :			
Premise		100000	
Machinery A & B	2106804		
Less: Depreciation	902916		
	--------	1203888	

		1303888	
Current Assets :			
Raw Material of Product A & B	900000		
Finished Products of A & B	1396535		
Cash On Hand	56441960		

	58738496		
Less: Current Liability:			
Over Draft	0		
Loan	35000		
	-------	35000	

		60007384	
Financed by			
Capital		101572273	
Net Profit		41564889	

			60007384

Production & Marketing
Combined Trading & Profit & loss A/C for the year Ended of period 1

	Production $		Marketing $		Combined $	
Sales		3974565		57419932		61394497
Less Cost of Good Sold:						
Opening Stock	1200000		950000		2150000	
+ Purchase	300000		950000		1250000	
- Closing Stock	900000	600000	1396535	3527964	2296535	4127964
	-------	-------	-------	-------	-------	--------
						57266532
Manufacturing Overhead:						
Machinery Overhead	6820				6820	
Wages (Indirect)	644000				644000	
Inspection Cost	252200				252200	
Factory cost of products producted:		903020				903020
		-------				-------
Gross profit		2471545		53891968		56363512
Less Expense:						
Depreciation of Machinery	902916				902916	
Maintainance Cost	155000				155000	
Repairing Cost	330000				330000	

General Expenses	30000		9000		39000	
Paid for Over Draft	0		0		0	
Interest Paid for load	0		25500		25500	
Carrying Cost of Raw Material	37500				37500	
Carrying Cost of Finished Product	7500		250000		257500	
Firing Cost	0		0		0	
Hiring Cost	0		0		0	
Rent & Rates	100000		180000		280000	
Wages			75000		75000	
Comnisson of Salespeople			20208		20208	
Advertising			200000		200000	
Cost for set up Branches			0		0	
Cost for closing Branches		1562916	0	759708	0	2322624
	-------	-------	-------	-------	--------	--------
Net Profit		907629		40657260		41564889

Production Summary (Product A)

Raw material
    Part A used : 50000
    Average quality level : 1.00
    Stock carryover : 150000
    Average quality level : 1.67

Capacity
    Manpower
        Newly hired : 0
        Left or transferred : 4
    Total no. of labours : 46

    Machine
        Newly purchased : 0
        Sold : 0
    Total no. of machine : 31

Assembling
    Regular man-hours used : 108.70
    Overtime man-hours used : 0.00

Machining
    Regular machine-hours used : 180.00
    Overtime machine-hours used : 29.68

Maintenance
    Number of maintenance/period : 2
    Number of breakdowns : 33
    Time lost in production : 4

Inspection
    Lot size : 1000
    Sample size : 100
    No. of defectives allowed : 5
    Average no. of inspection/lot : 604

Finished product
    No. of items produced : 50000
    Average quality level : 19.82
    No. of items transferred to Marketing : 0
    Production cost/unit : 31.15
    Existing stock : 0

Production Summary (Product B)

Raw material
    Part B used : 50000
    Average quality level : 1.00
    Stock carryover : 150000
    Average quality level : 1.00

Capacity
    Manpower
        Newly hired : 0
        Left or transferred : 4
    Total no. of labours : 46

    Machine
        Newly purchased : 0
        Sold : 0
    Total no. of machine : 31

Assembling
Regular man-hours used : 108.70
Overtime man-hours used : 0.00

Machining
Regular machine-hours used : 180.00
Overtime machine-hours used : 61.94

Maintenance
Number of maintenance/period : 3
Number of breakdowns : 33
Time lost in production : 5

Inspection
Lot size : 1000
Sample size : 100
No. of defectives allowed : 3
Average no. of inspection/lot : 888

Finished product
No. of items produced : 50000
Average quality level : 5.61
No. of items transferred to Marketing : 0
Production cost/unit : 48.34
Existing stock : 0

Production's Trading & Profit & Loss A/C for the Year Ended Period 1

Product A - Direct Sales at selling price	7500000		
Transfer to Branch(Marketing) at cost price	0		
		7500000	
Product B - Direct Sales at selling price	10000000		
Transfer to Branch(Marketing) at cost price	0		
		10000000	
			17500000
Less: Cost of Good Sold :			
Part A for : Opening stock of Raw Material	600000		
Product A: + Purchase of Raw Material	100000		
- Closing Stock of Raw Material	400000		
Raw Material of Part A Consumed		300000	
Part B for : Opening stock of Raw Material	600000		
Product B: + Purchase of Raw Material	200000		
- Closing Stock of Raw Material	500000		
Raw Material of Part B Consumed		300000	
		600000	
Manufacturing Overhead			
Product A - Machinery Overhead (per unit)	3100		
Wages (indirect)	322000		
Inspection cost	30200		
		355300	
Product B - Machinery Overhead (per unit)	3720		
Wages (indirect)	322000		
Inspection cost	222000		
		547720	
			903020

Gross Profit of Product A & B

Expenses

Product A:	Depreciation of Machinery	300972		
	Maintenance cost	62000		
	Repairing cost	165000		
	Carrying cost of Raw Material	12500		
	Carrying cost of Finished Product	2500		
	Firing cost	0		
	Hiring cost	0		
		----------	542972	
Product B:	Depreciation of Machinery	601944		
	Maintenance cost	93000		
	Repairing cost	165000		
	Carrying cost of Raw Material	25000		
	Carrying cost of Finished Product	5000		
	Firing cost	0		
	Hiring cost	0		
		----------	889944	
General Expenses (Overhead)		30000		
Paid for Over Draft		0		
Paid for Loan		0		
Rent & Rates		100000	131000	1563916
		----------	----------	----------
Net Profit				14433064

Production's Balance Sheet As at Period 1

Fixed Assest :		
Premise :		100000
Machinery A (Cost) :	702268	
Less: Accumulated Depreciation :	300972	
	----------	401296
Machinery B (Cost) :	1404536	
Less: Accumulated Depreciation :	601944	
	----------	802592

		1303888
Current Assets :		
Raw Material of Product A :	400000	
Raw Material of Product B :	500000	
Finished Product A :	0	
Finished Product B :	0	
Cash on Hand :	15503171	

	16403171	
Less: Current Liability		
Over Draft :	0	
Loan :	10000	10000
	----------	----------
		17697060
Financed by		
Capital :		3263996
Net Profit :		14433064

Development of a Computer Interactive Graphical Simulation in Manufacturing Education**

S.W.Cheung*

Business Games have been widely used in teaching of business subjects during the last 30 years and the trend is increasing. Several research studies have been carried out in order to evaluate the effectiveness of games in education. Although there is no hard evidence to demonstrate the superiority of business game against the other methods of teaching, they are certainly enjoyed more by students. Unfortunately, although simulation is used extensively in manufacturing industry for training and problem solving, it tends to be overlooked when teaching manufacturing engineering courses at universities.

This paper reveals the weakness of existing simulation games in manufacturing education and introduces the design process of a generic manufacturing simulation model called META (Manufacturing Engineering Teaching Aid) and is intended to open a new approach to manufacturing education. The package is expected to be completed within the next two years.

Introduction

Gaming has been widely used in education and training for many years. Everyone plays games from kindergarten to university. There is no hard evidence of where and when games were first invented [1]. Many people believe that the first use of a game to represent real world problems began with the Chinese game of Wei-Hai (Japanese Go) in the distant past. The game of chess is the oldest form of war game [2]. Today, exercises for business decision making may be considered as an outgrowth of military war games [3][4]. However, the name, 'game' often leads people to have misconceptions. There are several reasons why META is so named instead of business game or management game and they can be summarized as follows :

* Department of Manufacturing Engineering, Hong Kong Polytechnic, Hong Kong.

** The study was made at Department of Engineering, Manufacturing System Engineering, Warwick University, England.

1. People always regard games as a kind of entertainment which is an activity to have fun. Parents and politicians do not like their children to play games in schools especially universities. For example, a successful sociology course in the University of California, Berkeley had to be discontinued because gaming was the only teaching media used in the course [5]. Sometimes, even the participants have trouble understanding the objectives of a game; they play to win instead of to learn. Without any ideas of knowing what happens, they make the same set of decisions as in the previous round only because it yielded good results which allows them to win. Competition among players should be treated only as a form of motivation but it always hides the major objectives of a game.

2. gaming has been used in teaching business and social science subjects in universities for more than 30 years since the introduction of the first widely known business game, "Top Management Decision Simulation", which was developed by the American Management Association in 1956 [3]. The name 'business game' has existed in academia and training for so long that whenever one mentions an educational game, people always relate it to a business game. This gives the impression that games can only be used in teaching of business subjects. In fact, games can cover almost any subject ranging from survival in nuclear war to cattle breeding [6][7]. Basically, most business or social science games are built around a simulation which has been used extensively in manufacturing engineering [2]. Unfortunately, gaming seems to have been overlooked in the teaching of manufacturing engineering courses.

3. I name it a supporting activity because it does not attempt to act as a stand-alone teaching media. There are courses in business or social science using games as the only means of teaching [5][8][9] although there is no research evidence that suggests games are superior to lecturing and/or case studies [10][11]. META serves only as a complementary tool to existing teaching/research methods in manufacturing engineering. Lecturing or case studies are still necessary as the first introduction of the subject. META is not designed to replace the textbooks nor the lecturers. In fact, it creates a laboratory environment allowing students to put textbook ideas into practice which no other teaching media can satisfactorily accomplish.

4. Some authors like to put in a certain amount of random chance factors in a game to provide more fun and unpredictability of the results in order to avoid participants losing interest. This is acceptable if fact-teaching is not one of the objectives of the games. On the contrary, META is an academic development which gives an overview of a company. It shows the interactions between each functional department and their operating details. The existence of any unsuitable random factors can hide the fact conveying objective. For

example, a student may have made all the right decisions but he still gets a poor result only because he has bad luck. Realism is particularly important with META. In particular, it may be used for research purpose for which random factors can be cut down to a minimum, unless they are necessary such as the mean time between failures of a machine. In order to avoid misunderstanding, the name 'game' is avoided.

5. The relationship of simulation and game is a subtle one. One suggests that a simulation is a real time activity and a game compresses time. Another defines competition as a essential element in a game but not in a simulation [12]. Unfortunately, META has competition among players in the marketing field but not in the production simulator. Most game or simulation authors used pages and pages to make distinctions between them [12][5][13]. Even though much effort is used to classify whether META is a game, a simulation, a simulation-game, a game-simulation, a simulated case studies, a game used as a case study or a simulation-game used as a case study, at the end of the day, there always remain disagreement. In my opinion, the objectives of the package, the ways to accomplish the objectives, the design process, the implementation and the interpretation of the results that are important but not the name.

6. There remains a lot of disagreement on the definition of a game which mainly depends on the orientation of the author. In fact, META composes of a production simulator, a marketing aid and a financial aid. They can be run individually or integrated together to form a macro 'game'. It covers the lowest level of production simulation, such as the movement of material to the highest level of a management game where strategic planning is emphasized. It serves as a teaching aid as well as a research tool of new production planning technology. On the other hand, it is built in the form of a generic model to allow the administrators or researchers to program or tailor-make their own scenarios. In addition, the aid is built on a personal computer which provides fast and accurate calculation along with powerful interactive graphics to help users understand the system. Thus, in order to avoid endless arguments on definitions and putting restrictions on the structure and design process, I have named it "Manufacturing Engineering Teaching Aid".

Objectives of META

META possesses four primary objectives and one secondary objective which are summarized as follows :

Primary objectives :

1) To act as an aid to the existing teaching methods in manufacturing system engineering courses.

2) To provide an learning environment to the participants in manufacturing engineering courses.

3) To act as a research tool for the design and test of different production control techniques under controlled scenarios.

4) To provide instructors with a generic model so that they can write their own programs with minimum effort.

Secondary objective :

5) To establish a guide-line for designing a simulation teaching model in engineering science courses.

A Teaching Aid

Business games have been widely used in teaching of business subjects in the last 30 years. The trend is still increasing. According to a survey in 1987 in the U.S.A., over 1900 universities are using simulation games in some part of their business programs, a minimum of 3287 separate business courses are using simulation exercises and there are between 6100 and 7200 business firms currently using simulation exercises as part of their training programs [14]. Disappointedly, no similar statistics on the use of games in engineering departments could be found.

Several studies have also been carried out to evaluate the effectiveness of games in teaching or training in different fields especially in the business oriented areas [15][16][17]. Without surprise, no studies on using games in teaching of manufacturing engineering subjects could be found. Although there is no hard evidence to prove the superiority of games against other methods of teaching, students definitely find them more enjoyable than other teaching media [13]. The effectiveness of games in teaching is always difficult to decide upon and it is extremely difficult to measure the knowledge acquired in a game. Considerable research has been carried out to measure the effectiveness of games such as to compare the results of a before-game examination to an after-game examination or to compare the examination results of two groups of students in which one group is taught exclusively using games and the other group is taught by lectures or case studies. However, no definite conclusions have been formed. This is because the success of an educational game depends on many other factors such as the contents of the game, the backgrounds of the participants, the role of the administrators and the time available for each decision. On the other hand, where the evidence does not reveal benefits of gaming techniques over other modes of teaching, neither does it show the reverse; that is, those taught with games do not prove to have learnt less than those thought in traditional ways. While the arguments continue, gaming has been well accepted as a comparatively effective teaching media. Unfortunately, although simulation is used extensively in manufacturing industry for training or problem solving, the using of gaming is overlooked in teaching of manufacturing engineering courses at universities.

Production often involves many complex interactions between numerous functional areas such as quality control, design for manufacture, material handling and plant layout. In order to create a laboratory environment allowing students to practice ideas in dealing with these interactions, computer-based models are irreplaceable. Studies have shown that the use of interactive graphics is extremely helpful in understanding complex production systems [17][19][20]. So, META is actually a computer interactive graphical simulation educational package.

In 1977, Couger listed the twenty most popular games and none of them were related to production [21]. It is impossible to estimate the total number of gaming-simulation that have been developed but at the time of writing, the most complete listing of gaming-simulations is in the latest edition of "The Guide to Simulation/Games for Education and Training", edited by Robert Horn and Anne Cleaves in 1980. Over 1000 games are listed and only 21 of those are classified under "Production, Logistics, Operations". They are mainly small functional games on inventory control or distribution management which only covers a small portion of manufacturing engineering.

Ten years later, the situation has not improved. One even has trouble locating one topic related to production or manufacturing engineering on the resource lists of SAGSET (Society for the Advancement of Gaming and Simulation in Education and Training) in which even topics like music or health education can be found [22].

According to Elgood who had listed more than 200 management games currently available in Britain in 1988, there are only 13 games devoted to production but 11 of those are not computer based model. They are either discussion games or progressive games using charts or cards. The objective of the twelfth one is to study the managerial behaviour in manufacturing industry. Finally, there remains only one game (The MRP game) out of 200 in which MRP and MRPII are covered [23].

In addition, there is a common drawback in most functional games. A scenario set up by a game author is always tailor-made to a particular strategy or production technique which he wants to convey to the participants. The participants have no option to try out any other techniques under the same scenario. For example, if a game author wants to demonstrate how MRP works on the shop floor, there is no way for the participants to try out other production control techniques like JIT, PBC or OPT under the same scenario which is probably set-up to best suit MRP. In fact, in order for a student to learn MRP, not only the mechanics of the system that should be explained, it is the identification of different production environments. In fact, laboratory environment is required for the participants to experiment different techniques under the same scenario.

Obviously, there is a gap here and META is designed to fill in. META covers carefully selected elements from manufacturing in order to simulate the actual environment of a production company. It gives a overall picture of the industry with options to high-light a particular area. In particular, the production simulator of META provides the participants with the flexibility to set up his own factory, buy his own machines, design his own layout, and plan his own shop floor scheduling. It covers design for manufacture to material handling. It is not intended to replace textbooks or lectures since META does not teach knowledge but acts as a generic model allowing the participants of creating his own production environment, trying out ideas from the textbook and to observe the complex interactions on the screen.

In addition, the marketing and financial aids are designed to the explore knowledge of the participants, who are mainly working or studying in manufacturing engineering in marketing and/or financial positions. It is absolutely true that engineers and management personnel in manufacturing industry today are expected to have a certain amount of marketing and financial knowledge but one cannot expect them to be as good as an accountant to deal with dimes and pennies. It is not uncommon to find games with several pages of accounting reports after each round and many participants have difficulties of understanding the sophisticated accounting formulae. A global picture is important in order to analyze the interactions within the system, accounting details should therefore not be over emphasized to confuse the participants. Marketing and financial aids in META are designed from an engineering point of view and elements are carefully selected to match their participants' backgrounds as well as the course structure of a curriculum.

These three components, namely the production simulator, the marketing aid and the financial aid form the skeleton of META.

An integrated environment

Just imagine if a child has never seen an elephant before and he is shown a picture of the nose and it is explained to him how it handles things, then he is shown a picture of the legs and it is explained to him how an elephant walks and so on. By the end of the day, the child is required to draw a picture of the whole elephant. I have strong suspicions that the child may not get it right. One may even have difficulties of recognizing that the animal is an elephant who could have a nose on his back.

This is exactly how many manufacturing engineering courses are taught nowadays. The students learns design of manufacture as one subject and production planning as another. They play a marketing game in a marketing subject and a financial game when they learn how to balance the cash flow. It becomes worse if they even have to play several games in one single course to have different emphases. It is true that the whole manufacturing system is normally too complex to show to students in one shot, functional courses as well as functional games are necessary but at the end of the day, these functional games should be able to add up and form a whole system. An overview picture is essential to show the child how an elephant looks like. Unfortunately, the games used in different courses or even in the same course are often not compatible. The scenarios are often different not to mention the incompatibility of the hardware and software. Some of the total management games try to provide an integrated environment but two common drawbacks are found.

Firstly, in order to cover all the management area which normally includes marketing, production and finance, the total management game will simplify each functional area in order to make the game manageable. For example, in "The Scotsman Management Game" which has been run in the UK as an annual event for the last 19 years [24]. It covers a wide range of functions including marketing, personnel, production, research, purchasing and finance. Since the game covers numerous functional areas and cannot be separated into smaller ones, only a few decisions have to be made in each area and a lot of details are missing out. For example, shop floor scheduling, which is a most controversial topic, is ignored.

Secondly, the interactions between each functional area in total management games are always automatic. For example, in "STRATPLAN", the production will automatically adjust by 10% to match the sales and I have strong reservations as to whether one should buy or sell production capacity for "X" dollars per unit without any consideration of other resources [25]. These are common practices that one can find in a total management game but unfortunately, this is not always realistic.

META is designed to solve such problems. As has been previously stated it is normally divided into three sections, a production simulator, a financial aid and a marketing aid. These can be run individually as a functional aid or integrated together to give a overview picture. Students play different parts of the aids in different courses so that they get familiar with each functional area. At the end, all three areas can be put together to form a whole company and then the whole system will not seem to be as complex because the participants will have experienced each functional area before and their job will be to co-ordinate the different areas.

Furthermore, each functional aid can be divided into an infinite number of smaller aids. META provides a complete view in each functional area (such as layout, planning and quality control) but many elements can be highlighted or hidden according to the administrators. For example, quality control could be today's topic and material handling would not play an essential part in this. So, the administrator can hide the existence of material handling which may otherwise cause confusion to the participants. In fact, the function still exists in the system but the decisions on the function have been made by the administrators and there are even options of turning some elements off completely.

This allows the complexity of the aid to become totally controllable by the administrators to match the backgrounds of the participants as well as the pace of the class. That is, one may start to use the aid which requires only 5 decisions at the beginning of the class and end up with 50 decisions. The corresponding increase in complexity on a gradual basis will not frighten the students and it will enable them to learn step by step.

Research Tool

Simulation games are primarily used for training, but they have important research applications as well. Examples of research uses of business games includes areas of "International Relations", "Organizational Research", "Leadership and Team Effectiveness", "Introduction of New Technology" and "Management/Union Communication" [26][27][28][29]. When gaming is used seriously to determine optimum solutions for strategies and to determine optimum structure for systems, they may be termed "operational gaming" Thomas and Deemer define it as " the serious use of playing as a primary device to formulate a game, to solve a game, or to impart something of the solution of a game" [30]. In fact, the oldest form of gaming (military gaming) has long had this purpose.

In a functional business game involving a single person, optimum play may be achieved through attempts by the player to maximize some criterion in the game. In a multi-person, interactive game this is much more difficult, because there may be no single acceptable measure of success against the play of opponents.

Thus the production simulator is designed to run on a non-interactive environment so that research may be carried on without any noise.

Production planning technology has changed rapidly in the last decade. The introduction of Just-In-Time (JIT), brings production scheduling into a new era. Then, it comes optimum production technology (OPT) which was first implemented in United States in 1979. . Along with Material Requirement Planning II (MRP II), Period Batch Control (PBC), Group Technology (GT), Flexible Manufacture Cell (FMC) and different types of layouts (line, process, cell/group). Choosing a optimum production combination for a company is an extremely difficult job. Although there are several studies devoted to this area, there always remains the argument of what combinations of layout, production planning method etc. are the best to a particular situation [31][32][33][34][35]. Since the production simulator is a single person aid so that one can alter the scenario and try out different production combination techniques in order to obtain an optimum solution.
It will help decide on what production combination is most suitable.

Although it is out of the scope of manufacturing engineering, the marketing aid can be used to explore some aspects of the firms in an oligopoly situation. In fact, other research such as "human behaviour" or "effectiveness of computer-based simulation aid in teaching of manufacturing engineering courses" can also be carried out.

Generic model inside a generic model

Each educational game has its own specific purpose and a specific message to convey to the participants. There is always a preset scenario to best suit a particular technique or strategy which the game author wants to teach. Although some games do allow the administrators to change certain parameters, there are still a great deal of restrictions by the formats of the games. That means a game can only serve a unique purpose which is not economical in terms of development time and usage of resources. On the other hand, participants may have to play several different gam n different courses or even in the course. It takes time for the par nts to get familiar with a game stru before they can actually learn from \ warm up period is required i ıch game. According to Watson Blackstone, "Significant learning do t occur in games until all game rule descriptive facts are mastered and u ying game concepts emerge" [36]. I same game can be used in differer urses with different emphases, the l ng process will be accelerated. Stude o not have to waste time und ding the mechanics or the format game and they can concentrate thei nds on its contents. Although an ur al game is not possible, a multi-purp ckage, i.e. META may provide a sol

META is a generic age which allows the administrators ilor-make his own aid. In fact, ETA only provides a framework or leton of a teaching aid, the instructor to put the flesh on the bones to make ve. He has flexibility to build a scenar cording to his needs. For example, e subject "Design of Manufacture", ients such as "plant layout" and ection of machines" are more imp it but in another subject "productior nning and control", focus can be pu different production schedule chniques. Although it is impossible cover all topics, many typical manuf ring cases may be simulated by MET Since the student also has the flexibi to try out different techniques withi scenario which is already control by the administrator, META is actu a generic model within a generic mode

Obviously, flexibility eans more work in designing as well leveloping META, but once it is develo it can be used for teaching several ibjects in manufacturing engineering such as "Design for Manufacture", Production Planning and Control", Marketing strategy" and "Simu on of manufacture". Even in the e course, the instructor can highlight d rent areas in teaching different topics. dents will have more time to acquire owledge. For the lecturers, instead of s ching for a suitable game for their cou , they can program their own teaching aids with minimum effort. Certainly, several typical scenarios will be provided to the instructors for reference which will cover most typical manufacturing cases.

Designing engineering simulation-games

Although numerous games have been developed in various fields, little attention has been paid by the game authors in designing of games [5][13]. The designing procedures of a game are described as a "black box" by Ellington, Addinall and Percival [12]. Only four books devoted to design have been published since 1970, namely, "Simulation and Gaming in Social Science" by Michael Inbar and Clarice Stoll (1972), "Handbook of Game Design" by Ellington, Addinall and Percival (1983). The third one is Ken Jones' "Designing Your Own Simulations "(1986) and lastly, "Designing Games and Simulations" by Cathy Greenblat in 1988. Unfortunately, none of these are concerned with designing games for engineering courses. They either concentrate on design of simple simulations such as card games and board games or complex social science games. The design procedures of a simulation-games in engineering science are a "black box" within a "black box" . It is quite obvious that the design procedures of a social science game is quite different from that of a engineering science game. In designing a social science game, human behaviour is always a major concern but in engineering, for example mathematics has a greater emphasis.

Although it is not a major objective of META to establish a standard procedure of designing an engineering oriented educational package, valuable experience can be built up during the development process and this could be extremely helpful as a references for the game authors in this area in the future.

Conclusion

META is a software development sponsored by Hong Kong Polytechnic and Warwick University. It is expected to be completed in 1993 and will be tested in both institutes. The results however, will not be known until two years later. In short, the development of quality educational simulation packages in manufacturing engineering is something that is urgently needed .

REFERENCES

[1] Cohen, K.J. and Rhenman, E., "The Role of Management Games in Education and Research", Management Science, 1961,7,131-166

[2] "Simulation Training, A Guide for Trainers and Managers", A report by Rediffusion Simulation Ltd. for the training technology section of MSC, published by Manpower Services Commission, Sheffield, 1986

[3] Meier, Robert c. , Newell, William T. and Pazer, Harold L., "Simulation in Business and Economics", Prentice Hall, 1969

[4] Ginter, P.M. and Rucks, A.C., "War Games and Business Strategy formulation", Managerial Planning, 1983 Sep-Oct, Vol 32:2, p. 15 - 19, 34

[5] Jones, Ken, "Designing Your Own Simulations", Methuen, 1985

[6] Somogyi, Sandor and Kisimre, Istvan, "Operational Gaming of Cattle Breeding" in "Operational Gaming, An International Approach" edited by Ingolf Stahl, Pergamon Press, 1983

[7] Hartley, D.A., Johnson, P.V., Fitzsimons, A., Lovell, J., Chippendale, B. and Clayton, J.K.S., "A Case Study on the Development of the Home Defence Training Game, HOT SEAT", Operational Research society, 1979 Oct, Vol 30:10, P. 861 - 871

[8] Aplin, J.C. and Cosier, R.A., "The use of intensive simulation in executive development and academic settings" in S.C. Certo

and D. Brenenstuhl (Eds.), Insights into experiential pedagogy, Tempe : Arizona State University Press 1979

[9] Wolfe, J.A., "A comparative evaluation of a simulation-based business curriculum integration effort", Journal of Business, 1977, 50, p. 343 - 355

[10] Wolfe, Loesph and Guth, Gary, R., "The Case Approach versus Gaming in the Teaching of Business Policy : An Experimental Evaluation", Journal of Business, 1975, 48, 349 - 364

[11] Sculli, D. and Partridge, S.E., "Cases versus Gaming, The Case Approach versus Gaming in Teaching Basic Business Policy to Industrial Engineering Undergraduates", Management Education and Development, 10, 1979

[12] Ellington, Henry, Addinall, Eric and Percival, Fred, " A Handbook of Game Design", Kogan Page, 1982

[13] Greenblat, C.S., "Designing Games and Simulations : An Illustrated Handbook", SAGE, 1988

[14] Faria, A.J. "A Survey of the Use of Business Games in Academic and Business", Simulation and games, Vol 18, No. 2 , June 1987, p. 207 - 224

[15] Robinson, J. Nicholas, "Using Games to Present to Managers : A survey of the Literature", Management Education and development Vol 16, Pt. 1, 1985, p. 17 - 30

[16] Klein, Ronald D., "Adding International Business to the Core Program via the Simulation game", Journal of International Business Studies, Spring/Summer 1984

[17] Horn, Robert E. and Cleaves, Anne, "The Guide to Simulation/Games for education and training", SAGE, 1980, 4th Ed.

[18] Hurrion, R.D., "An Investigation of Visual Interactive Simulation using the Job-shop Scheduling Problem", Operational Research Society Journal, Vol 29, No 11, Nov 1978

[19] Miller, L.A. and Thomas, J.C., "Behaviour Issues in the Use of Interactive Systems", International Journal Man-Machine Studies, Vol 9, No. 5, Sept 1977

[20] Secker R.J.R., "That Visual Interactive Simulation Offers a Viable Technique for Examining Complex Production Planning and Scheduling Problems", Msc. Thesis, University of Warwick 1977

[21] Couger, J. Daniel, " 20 Most Popular Games", Computing Newsletter , Nov. 1977, p. 1

[22] SAGSET (Society for the Advancement of Gaming and Simulation in Education and Training), Resource lists on the following topics :

(1) Introductory Reading
(2) Game and Simulation Design
(3) Game and Simulation Evaluation
(4) Chemistry
(5) Economics
(6) International Relations
(7) Business and Management Relations
(8) Mathematics
(9) Teaching English as a Foreign Language
(10) Health Education
(11) Education Management
(12) Human Relations
(13) Geography
(14) Music

[23] Elgood, Chris, "Handbook of Management Games", Gower, 1988, 4th Ed.

[24] "The Scotsman Management Game", The Training Officer, Dec. 1980, p. 333 -337

[25] Hinton, Roy W. and Smith, Daniel C., "STRATPLAN", Prentice-Hall, 1985

[26] Fripp, John "Games for Research", Journal of European Industrial Training, 1984, Vol 8:3, P. 17 - 22

[27] Hogarth, R. M. and Makridakis, S., "The Value of Decision Making in a Complex Environment : An Experimental Approach", Management Science, Vol 27, No. 1, 1981, p. 93 - 107

[28] Rowland, K.M. and Gardner, D.M., "The Uses of Business Gaming in Education and Laboratory Research", decision Sciences, Vol 4, 1973, p 268 - 283

[29] Sloman, M. "How operational Gaming can Help with the Introduction of a New technology", Operational Research Quarterly, Vol 28, No. 4, 1977, p 781 -793

[30] Thomas, Clayton J. and Deemer, Walter L., Jr., "The Role of Operational Gaming in Operations Research", Operations Research, Vol 5, No. 1 (February, 1957), p. 6

[31] Fox, Robert E., "OPT vs. MRP : Thoughtware vs Software", Inventories & Production, November/December 1983

[32] Fox, Robert e., "MRP, Kanban, or OPT : What's Best ?", Inventories & Production, July/August 1982

[33] Burbidge, John L., "The Introduction of Group Technology", Wiley/Halstead, New York, 1975

[34] Schonberger, Richard J., "Some Observations on the Advantages and Implementation Issues of Just-in-Time Production Systems", Journal of Operations Management, 1982

[35] Goddard, Walter E. and Education, Oliver Wight, "Kanban versus MRP II - which is best for you", Modern Materials Handling, November 1982

[36] Watson, Hugh J. and Blackstone, John H., "Computer Simulation", John Wiley & Sons, 1989, 2nd Edition

The Second International Conference
on Automation Technology, July 1992

DEVELOPMENT OF A HIERARCHICAL, MODULAR MANUFACTURING SIMULATION MODEL

S. W. Cheung†, R. Roy††, and S. K. Bhattacharyya†††
Warwick Manufacturing Group
Warwick University
Coventry, CV4 7AL
U.K.

Abstract

This paper suggests a new approach to building a manufacturing simulation model. Writing a simulation model is always a complex and time consuming task even with the help of sophisticated simulation software like Witness or Siman. This is even more obvious when one tries to simulate a complex manufacturing environment. An approach suggested in this paper makes use of the concept of modularity endorsed by Zeigler [1][2]. Different production environments have a lot of common areas. The suggested approach is to break-up a complex manufacturing model into several small models which are called 'cells' in this paper. A cell may either be a 'machine cell', a 'resource cell' or a 'control cell'. The same cell can be used recursively and hierarchically in building up a model. Several generic cells, either the same or different, can be run simultaneously in a Windows multi-tasking environment. With the help of Dynamic Data Exchange (DDE) facilities, they can exchange information on a real time basis and share the same simulation clock as well as other resources. A user may create his own unique simulation model by integrating cells from the library. The user defined simulation model will be 'correct by construction'. Since the library models will be fully characterised and tested, the overall time to create a functional simulation will be drastically reduced. The cell interface will be specified in advance, and hence competent users (programmers) can also create their own unique cells to add to the library.

Keywords: discrete-event simulation, modular, hierarchy, DDE, multi-tasking, DEVS

Introduction

As the design of systems becomes even more complex and expensive, researchers have to pay more attention to the ability to model as well as to simulate these systems. There are two main research themes in discrete event simulation modelling - How to speed up the execution and how to simplify the modelling? On speeding up the execution, the introduction of distributed simulation (or parallel simulation) and Time Warp operating system provide a forum for discussion [3][4]. Special purpose languages, like Maisie, Sim++ and CPS have been developed for parallel simulation [5][6]. Also, a number of distributed simulation algorithms, broadly classified as 'conservative', 'optimistic' and 'conditional-event' have been constructed [5].

On simplifying the modelling process, most research work is built on the foundation of 'The Discrete Event System Specification (DEVS) Formalism' which provides a formal means of specifying systems in a modular and hierarchical manner [2][7][8]. To facilitate manufacturing simulation modelling, object-oriented simulation program generators, like Modsim, Robots and Robot-Sim have been developed. These, and the development of GIBSS (Generalized Interaction Based Simulation Specification) and Heterogeneous simulation models enable engineers to design manufacturing subsystems into various levels of abstraction [7].

The suggested approach modifies and combines DEVS with distributed simulation. Since the Windows multi-

† Senior Teaching Fellow
†† Lecturer
††† Professor and Head of Warwick Manufacturing Group

tasking environment simulates parallel processing, a complex manufacturing model can be broken up into several clusters which run simultaneously. Although the actual processing time would not be shortened since multi-tasking is only a time-sharing system, the authors believe that it is more important to provide a friendly, flexible, code-reusable and modular programming environment to the developers. Also, the problem of long processing time will eventually be solved by improvements in computer hardware which has increased the speed of computers exponentially over the last decade. In this paper, terminology used in DEVS will be bracketed with < >.

Correctness by construction

Building a simulation model is intricate (and hence tedious) work. Sophisticated simulation package like Witness and Siman do make the job easier but code reuse is difficult. One of the traditional approaches to modelling complex system is that of incremental development, that is, to start with a simple model and add in detail gradually. Unfortunately, difficulties in debugging grow with the complexity of the program. Different models usually reflect different combinations of basic resources such as machines, labours and materials, etc. In practice, they share some fundamental constructs and the code for these should be reusable; an analogy is like playing with Lego. Several bricks (similar or different) are arranged together to form a car. By rearrangement of the same fundamental bricks, one may get a van instead. Similarly, one may put different cells together to form a simulation model of plant A and by rearrangement of the cells, a simulation model of plant B can be built. Of course, new cells may be added, while some existing cells may be removed. Since each cell is a separate program which can be run and debugged individually, one only has to consider the interfaces between cells during the process of integration of different cells into a system. These bug-free cells constitute a library for future usage. This 'correctness by construction' has been applied in the development of systems for the design of Integrated Electronic circuits by many vendors.

Generic Cells

A generic single machine cell:

A cell <atomic model> is a basic unit of a simulation model. A cell can either be a machine cell, a resource cell or a control cell. A basic machine cell consists of only one machine. It is called generic because most of the characteristics of the machine (type of machine, machining time, set-up time, mean time between failure, type of labour, etc.) are inputted by users.

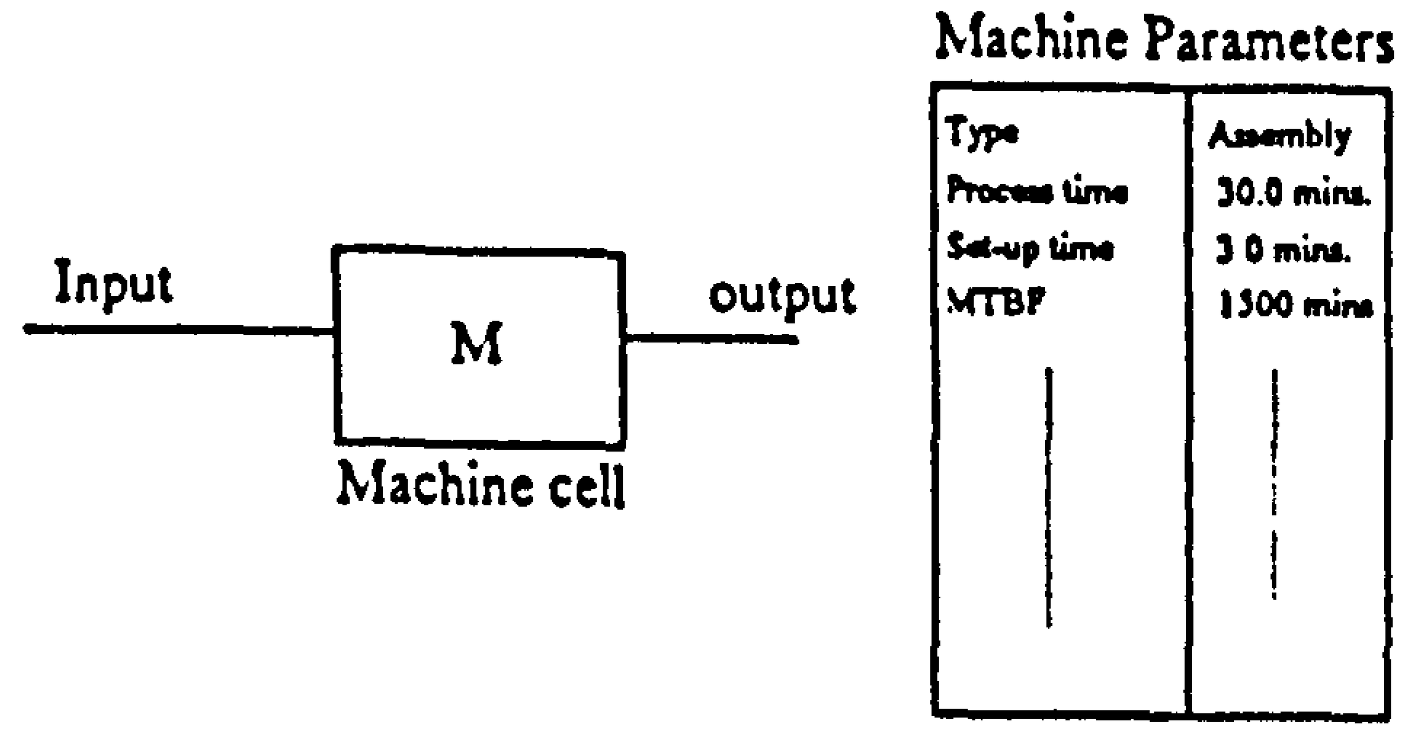


Fig. 1 A Single Machine Cell

A Generic Resource Cell:

A resource cell provides resources to machine cells. Resources may include labours, tools and materials, etc. A resource cell keeps track of the amount of available resource. A resource may also be a storeroom or a form of material handlers like robot. The basic format of a resource cell is as follows: (Fig. 2)

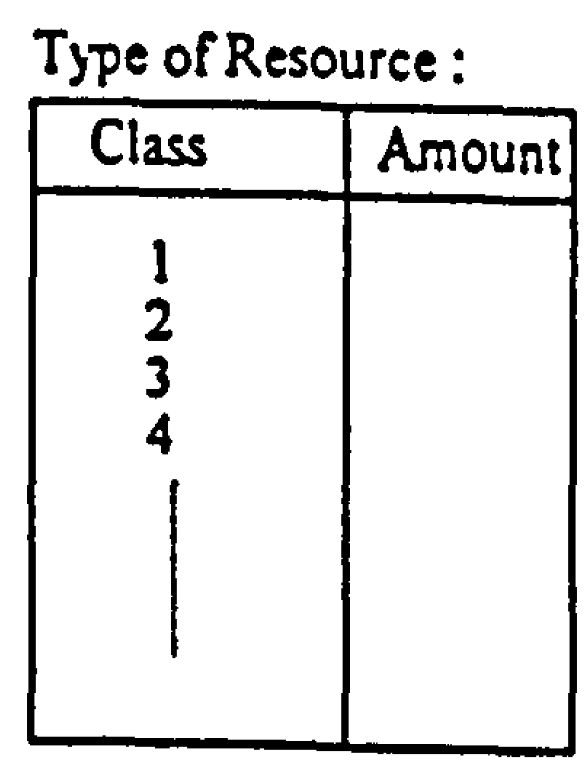


Fig.2 Basic Structure of a Resource Cell

The user may define the type of resource (say labour), number of classes (skill levels or types of skill) and the amount available in each class. When a machine needs a particular type of resource, it raises a request to the specific resource cell which then provides the resource if available. If the resource is not yet available, the machine waits and then restarts when the resource cell can supply the resource. That is, restarting of the machine is a consequent event of the arrival of resource. If the resource is reusable (like labour), the machine releases it back to the resource cell after the process. That means, two or more machines can share the same resource and the number of resources available is determined by the user.

A Generic Control Cell:

A control cell makes decision. It governs the logic of a system. For example, two parts (Part A and Part B) arrive at a machine at the same time. Which one should the machine first operate on? Then, a request for decision, the required data and a rule number are sent to a control cell which invokes the appropriate routine to perform the execution. The decision is then passed back to the control cell which in turn passes it on to the machine cell. One control cell is in charge of one type of decision and new rule routines can be developed and added to the list.

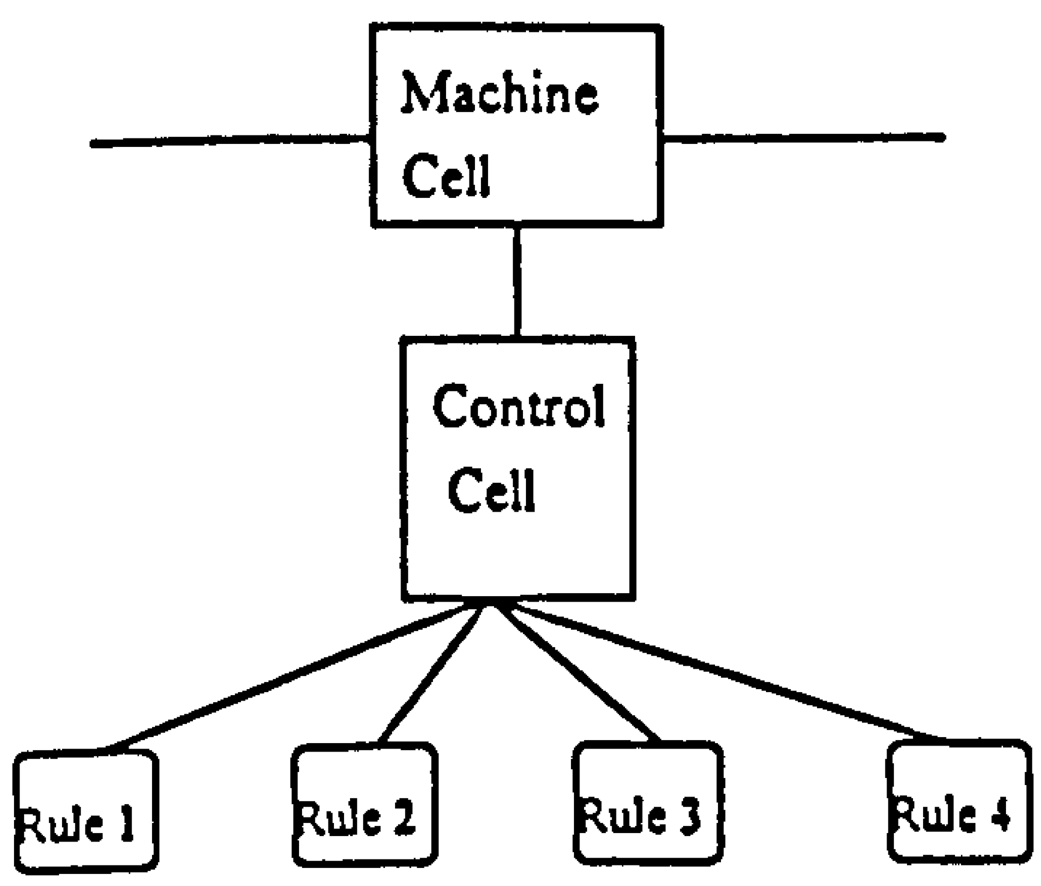


Fig. 3 Control Cell and Rule Routines

Recursion of a cell

Microsoft Windows allows two or more programs to run simultaneously. This multi-tasking feature enables two or more copies of a generic cell model to run at the same time. Each copy of the model may possess different parameter settings. The output of copy1 (cell 1) may be fed into copy2 (cell 2) as input <coupling> by using the Dynamic Data Exchange (DDE) facilities. That is, after a work-piece finishes its production process in cell 1, it will enter into cell 2 to continue the production sequence. Theoretically, infinite variants of a basic cell may be linked together to run simultaneously to build up a very large, complex simulation model. At the same time, they may share the same resources. For example, in figure 4, cell 1 to cell 5 are single machine cells in which cell 1 and 2 need labour and cell 2 and 4 require tool changes. Cell 1 also obtains raw material from resource cell and cell 5 releases the output to the storeroom (another resource cell).

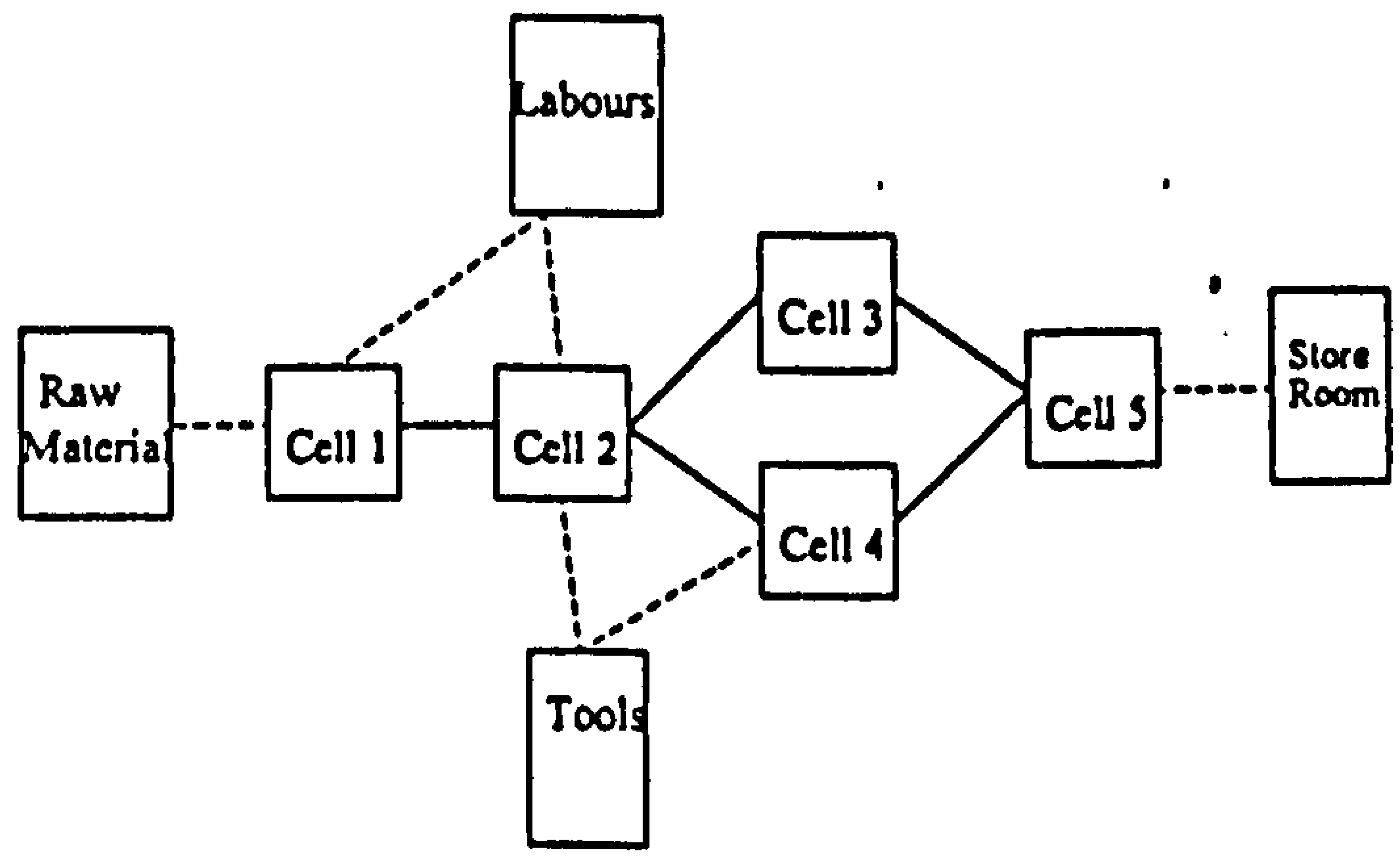


Fig. 4 A Sample Production System

If cells are designed with different configuration (with/without labour, constant/variable machining times, with/without buffer, with/without breakdowns, with/without machine set-ups, push/pull production systems, etc.), an almost infinite combination of production systems can be formed as long as the protocols (formats of data exchange) between cells are well defined and remain unchanged. Since the production model is built on a modular structure basis, updating or editing will be quite straightforward.

In order to minimize the amount of data flow between cells, characteristics of a part (e.g. processing time, required resources) are stored as machine parameters. When there is more than one product type going through a machine, a corresponding set of machine parameters is read in to change the settings of the machine if necessary. For example, when two different parts (part A, part B) are routed through a machine with different process times and set-up times, two sets of machine parameters (file A and file B) are prepared respectively. When a machine receives a part (say A), the product type is checked. If the type does not match with the current settings (say B), the corresponding file (file A) will be read in to update the settings of the machine.

Standard Machine Cells <Coupled Models>

Theoretically, a single generic machine cell can make up any production system but the time involved in linking up each cell as well as the running time of the simulation model increase greatly with the complexity of the system. Standard units of enhanced functionality are built to facilitate the process of modelling. An analogy is like construction of an electronic circuit. Basically, only three types of logic gates (AND, OR, NOT) are required for any circuit but one can imagine the size of a personal to start from the beginning. In the modular approach, the user can execute all three models at the same time; delete the unwanted portions and do the re-routing on the screen to obtain the fourth one. Furthermore, the output of modular modelling is itself a generic model on a cell level.

Simulation Clock

The Simulation clock is a key component of a simulation model. It does not matter how many cells are put together to form a system, there is only one simulation clock inside the system. As usual, it keeps track of the completion time of events. Dynamic Data Exchange facilities are used to communicate between the simulation clock and the machine cells. Even if the production system is composed of only one cell, two programs (the simulation clock and the cell model) have to be called up simultaneously. If one tries to develop a new cell model, the simulation clock is treated as a built-in function as in other simulation languages like SeeWhy.(Fig. 8)

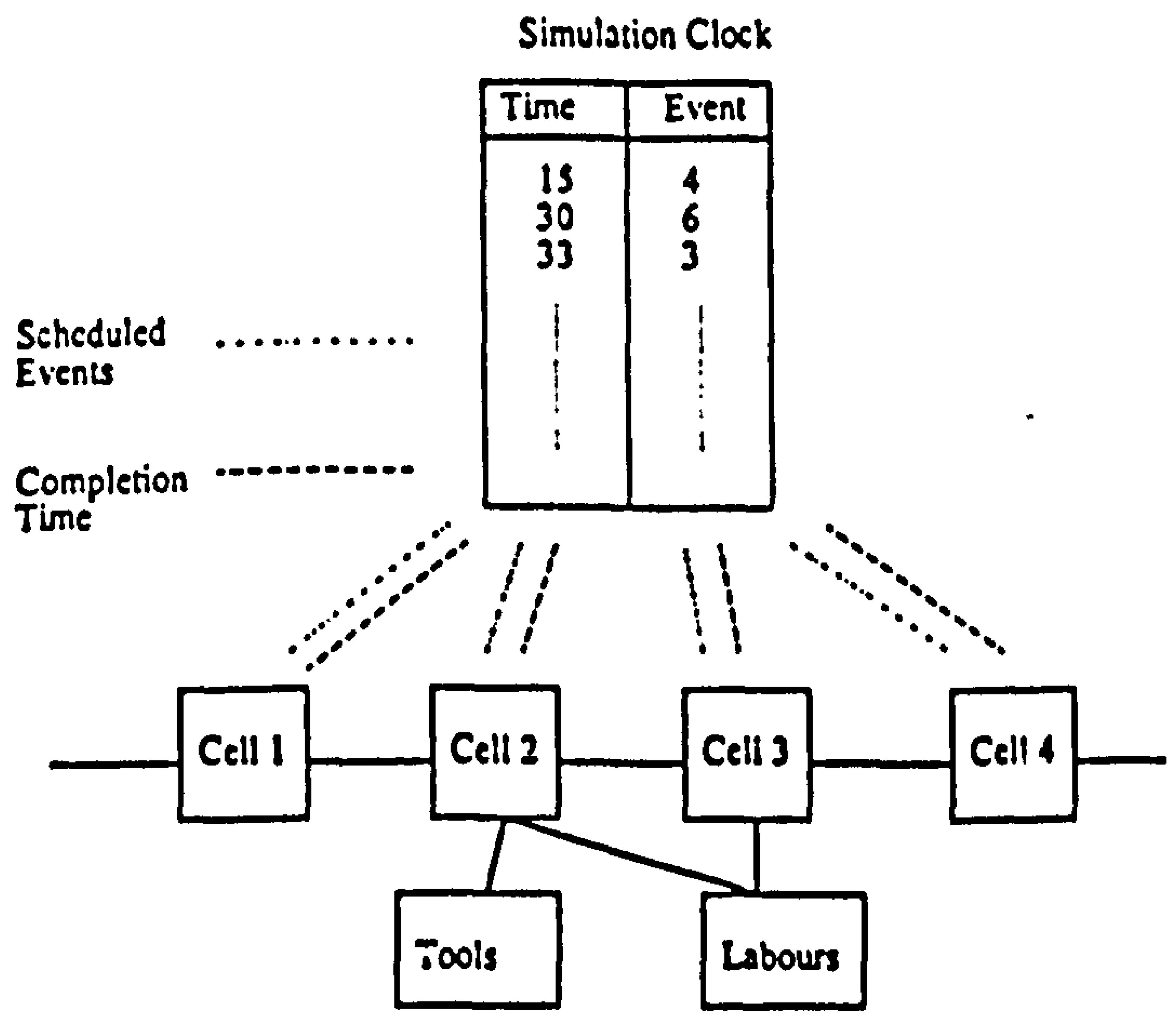


Fig.8 A Simulation Clock

Modular Model Vs. Generic Model

To a certain extent, a Modular Model and a Generic Model look similar. For example, in both cases, a user can input different sets of parameters to make up different simulation models. But in fact they are quite different. Physically, a generic simulation model is a single program but a modular model may involve several programs running at the same time. In a generic model the basic structure of a program is fixed. The variation of a model is limited by the number of available parameters and the variation allowed in each parameter. Whether the users find the model 'generic' enough depends on the original design by the developer. If a new machine is purchased and the characteristics of this machine are not covered by the existing parameters, a user may have to make appropriate revisions to the existing code. If the same situation occurred in modular modelling, one only has to program a new machine cell and add it to the system. In fact, in a Modular Model, 'generic' only occurs at a cell level. The actual structure of a model is determined by the user. However, a user may find a generic simulation system more user friendly but in terms of flexibility, control and expandability, modular model is preferred.

Selection of software

The concept is now being developed in QuickC for Windows ver. 1.0 with the help of Software Development Kit 3.1 under Windows 3.1. C, a procedural language is chosen to implement the object-oriented paradigm because C is more popular and portable than any object-oriented language. Hence, many traditional programmers can participate in development of new cells without changing their programming styles. Object-oriented concept are utilized in this work on a structural level, but within the cell itself, top-down design is still applicable. Thus, the formation of the model or the structure follows the object-oriented concept, but programming itself remains on a cell level which can be easily implemented with traditional computer languages (like Pascal or Basic).

QuickC is chosen as the development software because Windows programming in Microsoft C with Software Development Kit (SDK) is fairly complicated and QuickC can resolve some programming overhead. In fact, Microsoft claims that QuickC for Windows is a total solution for Windows programming and it provides reasonable control as well as flexibility. Furthermore, it is upward compatible with C and SDK. That is, if something cannot be programmed with QuickC, SDK can always step in to fill in the gap. At the time of writing this paper, DDEML (Dynamic Data Exchange Management Library) is the only function that is missing in QuickC which may be required. Fortunately, without too much difficulties, the DDEML function from SDK 3.1 is incorporated into QuickC environment [9][10][11].

New cell development is best conducted under the same programming environment to ensure compatibility but any package (like Excel 4.0) or language (like Visual Basic) which supports Dynamic Data Exchange can be used to develop the atomic model.

computer if only three types of logic gates are used. The followings are examples of standard units and the programmers or developers are always welcome to design their own favourite standard units. (Fig. 5)

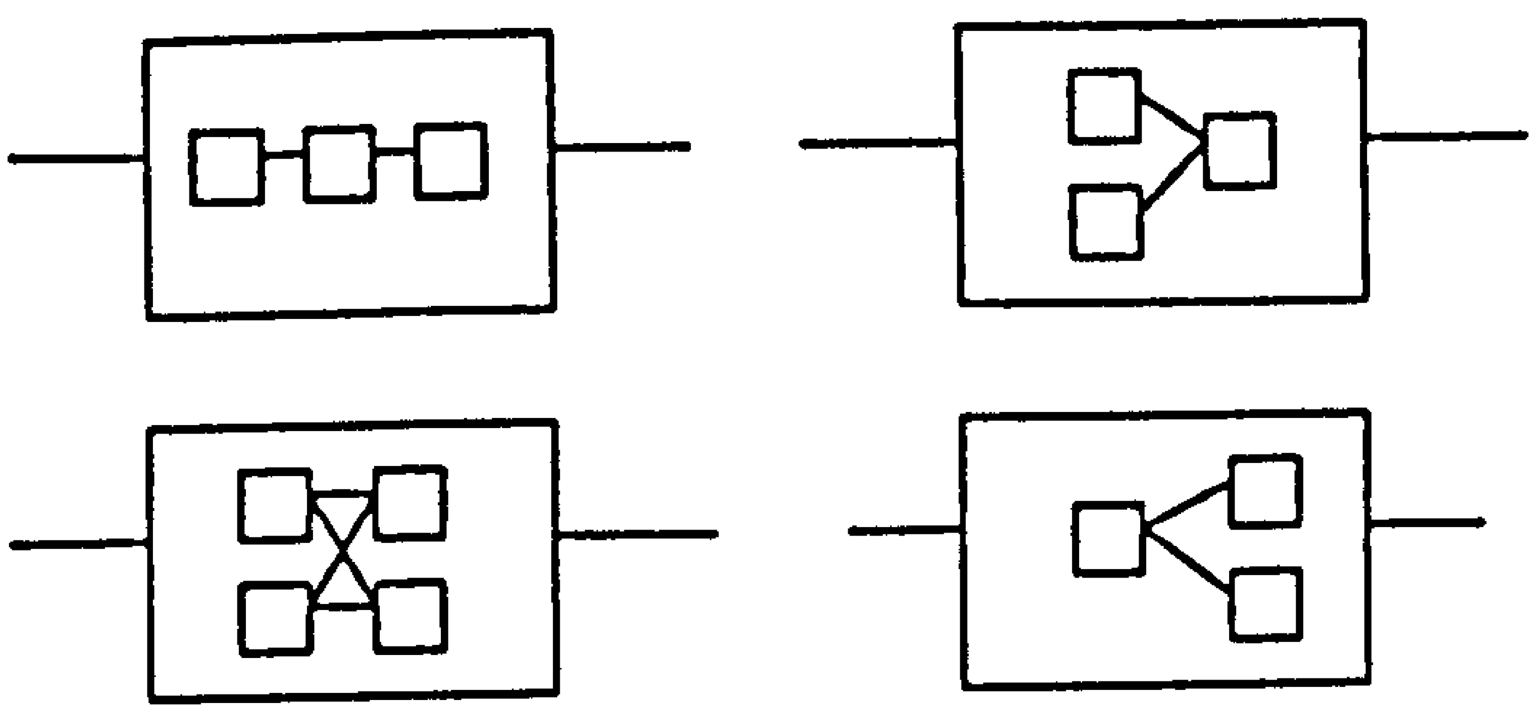


Fig. 5 Sample Standard Machine Cells

After testing, these standard units will be available from the library <model base> to build a more complex system which can in turn be added to the library. This hierarchical approach of building models enriches the library endlessly so that even more complex models may be built quickly and easily.

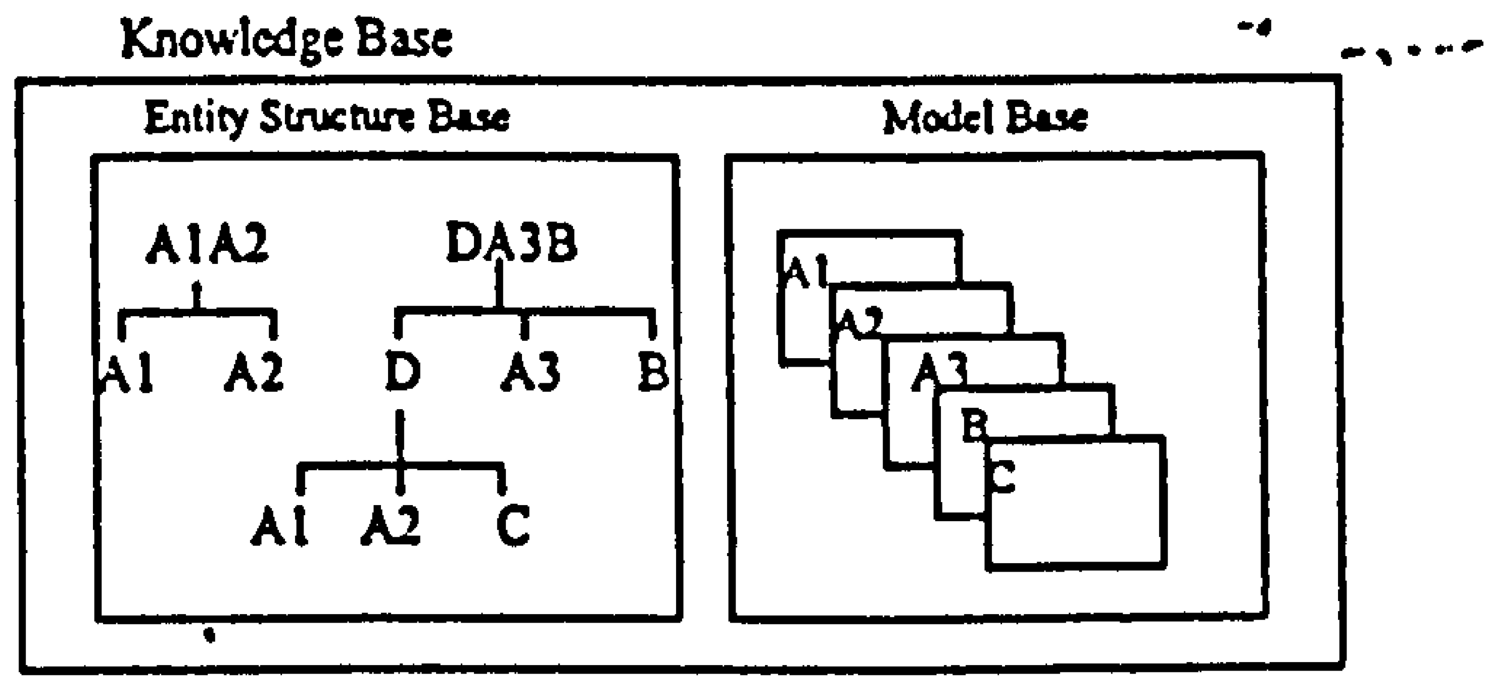
System Entity Structure

Zeigler, in his book, 'Object-Oriented Simulation with Hierarchical, Modular Models' defines System Entity Structure as:

"The System entity structure (SES) directs the synthesis of models from components in the model base. The SES is a knowledge representation scheme that combines the decomposition, taxonomic, and coupling relationships. Associated with an SES is a model base which contains models which may be expresses in any of dynamic formalisms mentioned earlier. The entities of the SES refer to conceptual components of reality for which models may reside in the model base. Also associated with entities are slots for attribute knowledge representation" [2].

In the suggested approach, each type of a generic cell (machine cell, resource cell and control cell) is a 'object'. Copies of the same cell are 'instances'. The operations performed in the cell are 'methods'. The generic cells and the standard units <coupled models> form a model base. Different possible ways of linking cells together describe the system entity structure. The entities of the SES refer to any simulation model which may be reused, that is, they can reside in the model base.

A System Entity Structure/Model Base (SES/MB) framework is drawn below (Fig. 6):



A1 : Machine Cell 1
A2 : Machine Cell 2
A3 : Machine Cell 3
B : Control Cell
C : Resource Cell 1

Fig. 6 The System entity Structure/Model Base Framework

The DA3B model:

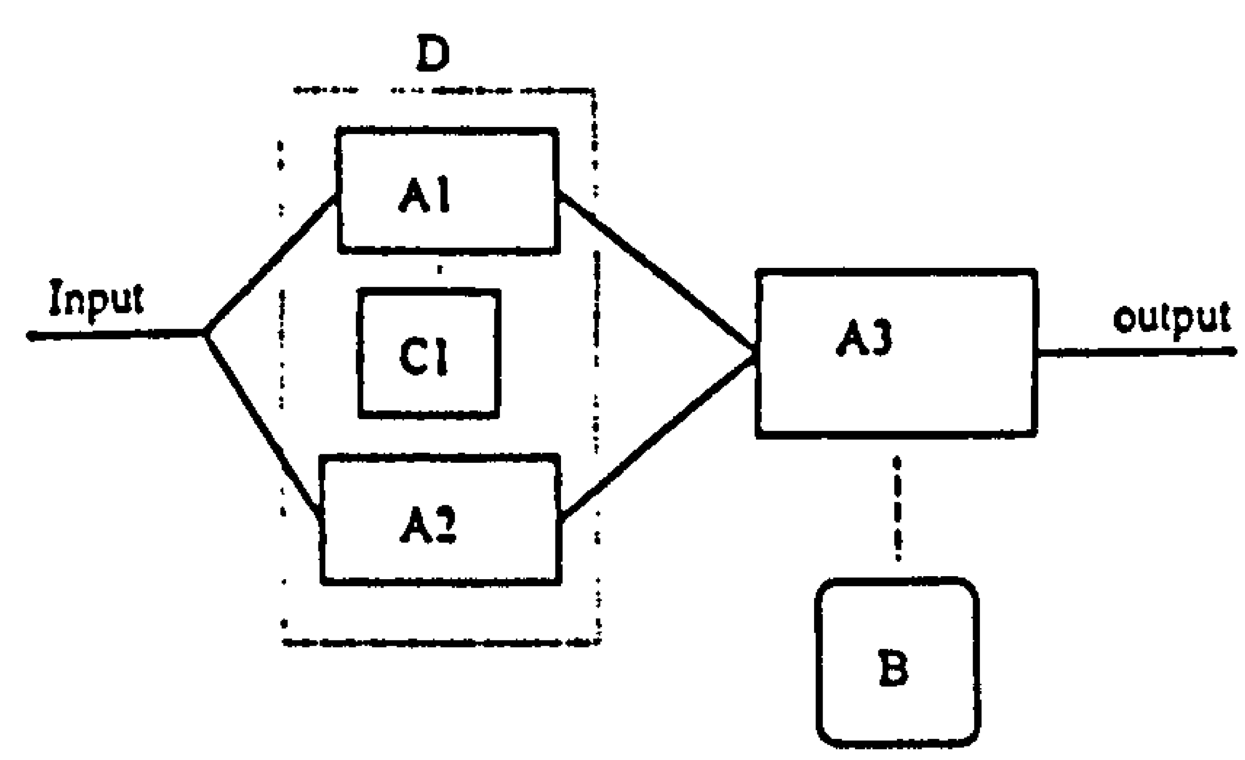


Fig. 7 DA3B Simulation Model

The DA3B tree is referred to as The Composition Tree under DEVS formalism. Instead of a direct implementation of Zeigler's DEVS formalism into manufacturing simulation, several modifications have been made to increase the flexibility as well as the user-friendliness of the programming environment. The modified approach provides high-level code-reusability without recompiling. All the cells <atomic models> in the library <model base> are in executable format. They can be run independently and simultaneously. Firstly, the user executes all the chosen cells <atomic or coupled> under Windows. Then, dynamic links between cells are set up during run time by the user without having to write a single line of code. The construction process is quite similar to building model with a generic package (like Witness). Unfortunately, generic modelling does not provide reusability of code. For example, if a user wants to take out three subsystems from three different complete models and merge them together to form a new one, he may have

Conclusion

The suggested approach is not intended to supercede the traditional approaches to building simulation models but to augment them. Other production facilities (object types) like conveyor or automatic guided vehicle (AGV) have to be implemented to cell structures before a complete modular simulation environment is formed. In this application the modular concept is being implemented using C in a Windows environment, but there is no reason why other techniques such as rule-structured knowledge based systems may not be applied.

References

[1] Zeigler, Bernard P., 'Hierarchical, modular discrete modelling in an object-oriented environment', *Simulation*, Nov. 1987 pp. 219 - 230

[2] Zeigler, Bernard P., *Object-Oriented Simulation with Hierarchical, Modular Models*, Academic Press, Inc., 1990

[3] Abrams, M., Lomow, G., 'Issues in Languages for Parallel Simulation', Distributed Simulation, *Proceedings of the SCS Multiconference on Distributed Simulation*, 1990 San Diego, edited by David Nicol, Society for Computer Simulation.

[4] Zeigler, B.P., Christensen, E.R., 'Distributed Discrete Event Simulation: Combining DEVS and Time Warp', Distributed Simulation, *Proceedings of the SCS Multiconference on Distributed Simulation*, 1990 San Diego, edited by David Nicol, Society for Computer Simulation.

[5] Bagrodia, R.L., Liao, W.T., 'Maisie: A Language and Optimizing Environment for Distributed Simulation', Distributed Simulation, *Proceedings of the SCS Multiconference on Distributed Simulation*, 1990 San Diego, edited by David Nicol, Society for Computer Simulation.

[6] Baezner, D., Lomow, G., Unger, B.W., 'Sim++: The Transition to Distributed Simulation', Distributed Simulation, *Proceedings of the SCS Multiconference on Distributed Simulation*, 1990 San Diego, edited by David Nicol, Society for Computer Simulation.

[7] Meter, E.C.D., Deisenorth, M.P., 'GIBSS: A model specification framework for multi-stage, manufacturing system design', *Simulation*, June 1991, pp. 413-421.

[8] Zeigler, B.P., Lee, C., Kim, T.G., 'A Hierarchical Methodology for Multilevel AI Computer Architectures', Distributed Simulation, *Proceedings of the SCS Multiconference on Distributed Simulation*, 1990 San Diego, edited by David Nicol, Society for Computer Simulation.

[9] Microsoft, *Manuals of QuickC for Windows ver 1.0*, 1991.

[10] Tom Vager, 'Swift Programming for Windows, in Windows', *Byte*, pp. 251-252, March 1992.

[11] Charles Petzold, *Programming Windows*, 2nd Edition, 1990 Microsoft Press.

Object-oriented, Hierarchical, Modular Manufacturing Simulation Models with DEVS and Distributed Simulation

S.W. Cheung, C.H. Chan and H.M. Wong
Warwick Manufacturing Group
University of Warwick
Coventry, CV4 7AL, U.K.

ABSTRACT

A suggested approach building Hierarchical, Modular manufacturing simulation model was suggested by Mr. Cheung. A prototype consists of 9 machine cells and a simulation clock has been developed under Windows multi-tasking environment using QuickC for Windows. In the model, each cell is an individual program. Dynamic Data Exchange (DDE) facilities are used for communication between modules. Users can build models without programming and vary different parameter values to the models for experiments. In this paper, design of machine Cells, simulation clock and their protocols are explained. Besides, the logic design of other cells such as Vehicle Cell, Resources Cell and Control Cell are discussed. Further research work is suggested to enchance the deisgn from facility cell level to shop floor level.

INTRODUCTION

With respect to hierarchical, modular manufacturing simulation, a prototype was developed using QuickC for Windows. This approach was suggested by Cheung (refer to [Cheung 1992]) which modified and combined Discrete Event System Specification (DEVS) of Zeigler with distributed simulation. The approach is to break up a complex manufacturing model into several small models which are called 'cells' in [Cheung 1992]. The system components like machines and resources are modelled in separate executable programs (cells) running simultaneously under Windows multitasking environment. Communication between these cells is implemented by Windows Dynamic Data Exchange (DDE) facilities. Besides the modular modelling and distributed simulation concepts, object-oriented paradigm is used in the suggested approach on a structural level. This concept enables the hierarchical decomposition of systems and the development of modular, reusable models. The approach explores the possibility of employing DDE protocol to implement the Object-oriented paradigm without changing the programming style of traditional programmers.

The concepts of hierarchical, modular modelling applied in this model were first suggested by Zeigler. In [Zeigler 1987]. He defined modularity as:

"the description of a model in such a way that it has recognised input and output ports through which all interaction with the external world is mediated."

Zeigler also explained the concepts of Modularity and Model Base in [Zeigler 1990]. In this paper, these concepts are implmented with the application of DDE in multiple programs (modules).

DYNAMIC DATA EXCHANGE (DDE)

Microsoft Windows provides multitasking environment which allows several programs to be run at the same time. DDE is a protocol for exchanging data between Windows applications. Messages are passed between the programs to transfer data. Windows is message based and a standard message system has been provided by Microsoft. By DDE, different programs can exchange data without ongoing user interaction. Once the link is established, data can be updated automatically. In the suggested approach, different modules are independent Windows executable programs in which DDE forms the protocol.

The Object-oriented concept in software development enhances software maintainability, extendibility and reusability

[Kim 1988] and [Doyle 1990]. However, the drawback to OOPS is a long learning curve for programmers who are ingrained in traditional programming techniques [Ladd 1990]. Kornfeld and Gilhooly stated that DDE can meet the requirements for an Object-oriented language. According to [Kornfeld & Gilhooly 1992], the benefit of DDE implementation is that:

> *"it allows a staged migration between development cultures and languages. For example, in a C shop, programmers could write application processes in the familiar language but use DDE constructs to communicate and synchronize with other processes, creating a framework for an object-oriented system."*

Two programs are in 'conversation' when they are communicating with DDE. The one that provides the required data is called 'server' while the other that receives the data is called 'client'. An application can be a server and a client at the same time and involve in more than one conversation. In order to distinguish between different conversations, the identity of a conversation is determined by 'application', 'topic', and 'item'. These names are supplied to initialize a DDE linkage.

Different operations of DDE are based on 9 messages (details please refer to [Microsoft Windows Guide to Programming]). Three kinds of linkage can be setup - Hot link, Cold link and Warm link. The difference determines the actions that are needed when an item has changed its value. Hot link is used when the client wants to get the updated data whenever it changes. In cold link, the client will post 'request' message each time to the server when it needs the data instead of posting 'advise' message. For warm link, the server will post a null 'data' message to the client whenever the data has changed. The client can then post 'request' message to server to get the update value depends on its own conditions.

SIMULATION CLOCK

The simulation clock is a time-advance mechanism. It keeps track the current value of the simulation time and advances the time as the simulation proceeds. In the model built, the simulation clock is initialized to zero and time of occurrence of future events is determined. The simulation clock is then advanced to the time of occurrence of the next event and synchronizes all production facilities.

The ClockCell communicates with the Machine Cell (MCell) by hot link. The following table (Table 1) shows the protocol between a ClockCell (application name: sclock) and a machine cell (application name: MCell).

Table 1: Descriptive protocol between a ClockCell and a Machine Cell

From	To	Message	Descriptive Protocol
MCell# (Client)	ClockCell (Server)	Sends WM_DDE_INITIATE	Hello, is this sclock (application name)? I would like to initiate a conversation about the sclock (topic name).
ClockCell	MCell#	Sends WM_DDE_ACK	Yes, this is ClockCell. I am sclock (topic name). I will be glad to converse with you.
MCell#	ClockCell	Posts WM_DDE_ADVISE	I want you to send me the current time of simulation clock.
ClockCell	MCell#	Posts WM_DDE_ACK	Yes, I do.
MCell#	ClockCell	Posts WM_DDE_REQUEST	Can I get the current time of simulaton clock?
ClockCell	MCell#	Posts WM_DDE_DATA	Here is the current time.
MCell#	ClockCell	Posts WM_DDE_POKE	I give you the future completion time of the part(s).

After a production operation (machining or transportation), the next scheduled event time and the corresponding cell number are posted to the ClockCell by WM_DDE_POKE message and the time is stored in an array. The time from the MCell1 is assigned to element array[1] and those from MCell2 is assigned to array[2] and so on. In every cycle, the ClockCell has to check if all the MCells have posted the scheduled event time before the next simulation time can be determined. Then, the ClockCell sorts for the smallest value in the array and updates the simulation time of the machine cells by posting 'data' message through the hot link. Figure 1 shows a flow chart of the above functions of the ClockCell.

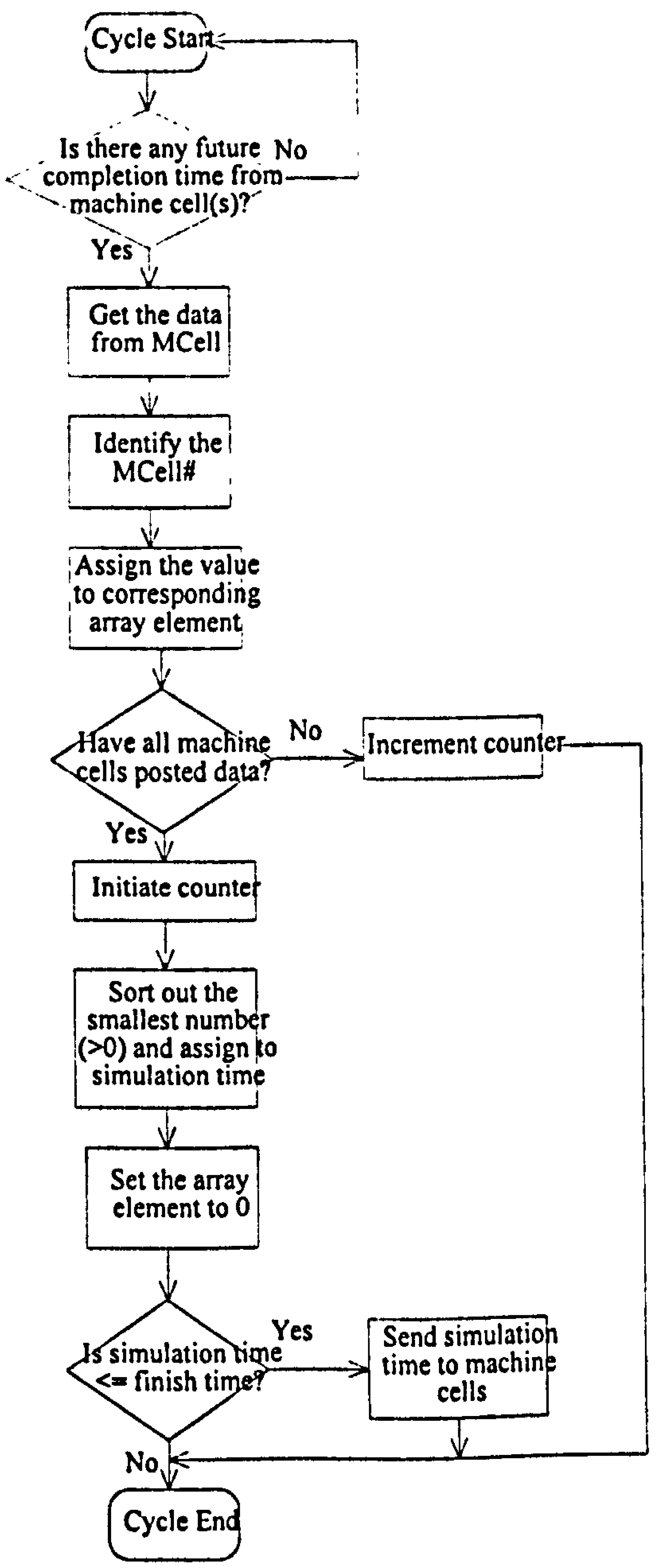


Figure 1: The major functions of ClockCell

Array limits the maximum number of cells which is fixed by the array size defined in the program. The number of cells can be varied by using pointers data structure instead of array to increase flexibility. However, using pointers will increase the sorting time of the ClockCell which is the potential bottleneck of the model.

FACILITY CELLS

Facility cells consist of machine cell, vehicle cell and buffer cell etc. The prototype of a machine cell has been developed. A machine cell schedules the delay of a part (processing time) and changes the attributes of the processed parts within the system. The machine cell changes from busy to idle or from idle to busy as time advances.

Cells can communicate between each other once the data link is established. The Transfer Quantity Data (TQD) contains the batch size and other part attributes. The part attributes consist of production sequence, cycle time, setup time, product types etc. Machine parameters like mean time between failure (MTBF) are inputted by users initially to each machine. A machine cell transfers a part (TQD) to the next machine and it can also exchange the current and future job completion time with the simulation clock.

Generally, a manufacturing simulation model is composed of a number of instances of different types of cell and a simulation clock. Each instance has a unique topic name. Topic names of different machine instances are called MCell1, MCell2, MCell3 and so on.

After the conversation is established, specific data values can be referenced by names and transferred between client and server. The following example describes how the protocol works. In Figure 2, three machines form a single machine line.

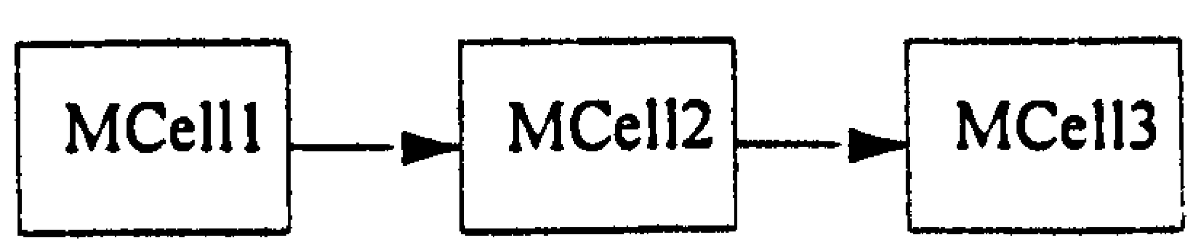


Figure 2: A machine line

Machine cell 2 (client) initiates a conservation with machine cell 1 (server) by specifying "MCell" (application name) and "MCell#" (topic name). After the link is established, machine cell 1 can send the part's information to machine cell 2. Since machine cell 3 (client) also requires to initiate a conversation with the machine cell 2 (server), machine cell 2 becomes server. That is, all machine cells are both client and server at the same time. The following table (Table 2) shows the protocol between machine cell 1 (MCell1) and machine cell 2 (MCell2).

Table 2: Descriptive protocol between Machine Cells

From	To	Message	Descriptive Protocol
MCell2 (Client)	MCell1 (Server)	Sends WM_DDE_INITIATE	Hello, is this MCell (application name)? I would like to initiate a conversation about the MCell1 (topic name).
MCell1	MCell2	Sends WM_DDE_ACK	Yes, this is MCell. I am MCell1 (topic name). I will be glad to converse with you.
MCell2	MCell1	Posts WM_DDE_ADVISE	I want you to send me a part (item name: TransferQty) when you finish the process.
MCell1	MCell2	Posts WM_DDE_ACK	Yes, I do.
MCell2	MCell1	Posts WM_DDE_REQUEST	Can I get the part from you (MCell1)?
MCell1	MCell2	Posts WM_DDE_DATA	Here is the part.

PUSH SYSTEM

Push system is implemented with hot link between machine cells in the prototype. Whenever a upstream machine cell finishes a process, it passes the job to the next machine cell. The downstream machine cell stores the transferred job in the input buffer. In fact, the upstream machine cell passes a Transfer Quantity Data (TQD) message to the downstream machine cell each cycle to update the buffer value. The downstream machine cell has to wait for the TQD message before it can determine the next schedule event time and to start the operation. Eventhough the upstream machine cell does not have a part, a TQD with zero value will still be transferred (Program Listing is attached in Appendix).

PULL SYSTEM

For pull system, cold link is used. Whenever a facility cell needs the supply of job (conditions depend on different cell design), it posts a Request message to the server facility cell to get a new job. If the server facility cell can supply the job, it posts a Data message to the client cell. If not, it posts a Negative Acknowledgement message. The client cell then becomes starved and waits for the next cycle. For the server facility cell, when the maximum output buffer size is reached, the machine stops.

FUTURE EVENT TIME

After receiving the current simulation time from the simulation clock, the machine cell compares the scheduled event time with the present simulation time to determine whether the current process has completed or not. Then, the statistical counters such as the percentage utilisation, the Input and Output buffer value, etc. are updated and the future event time is determined. (Figure 3).

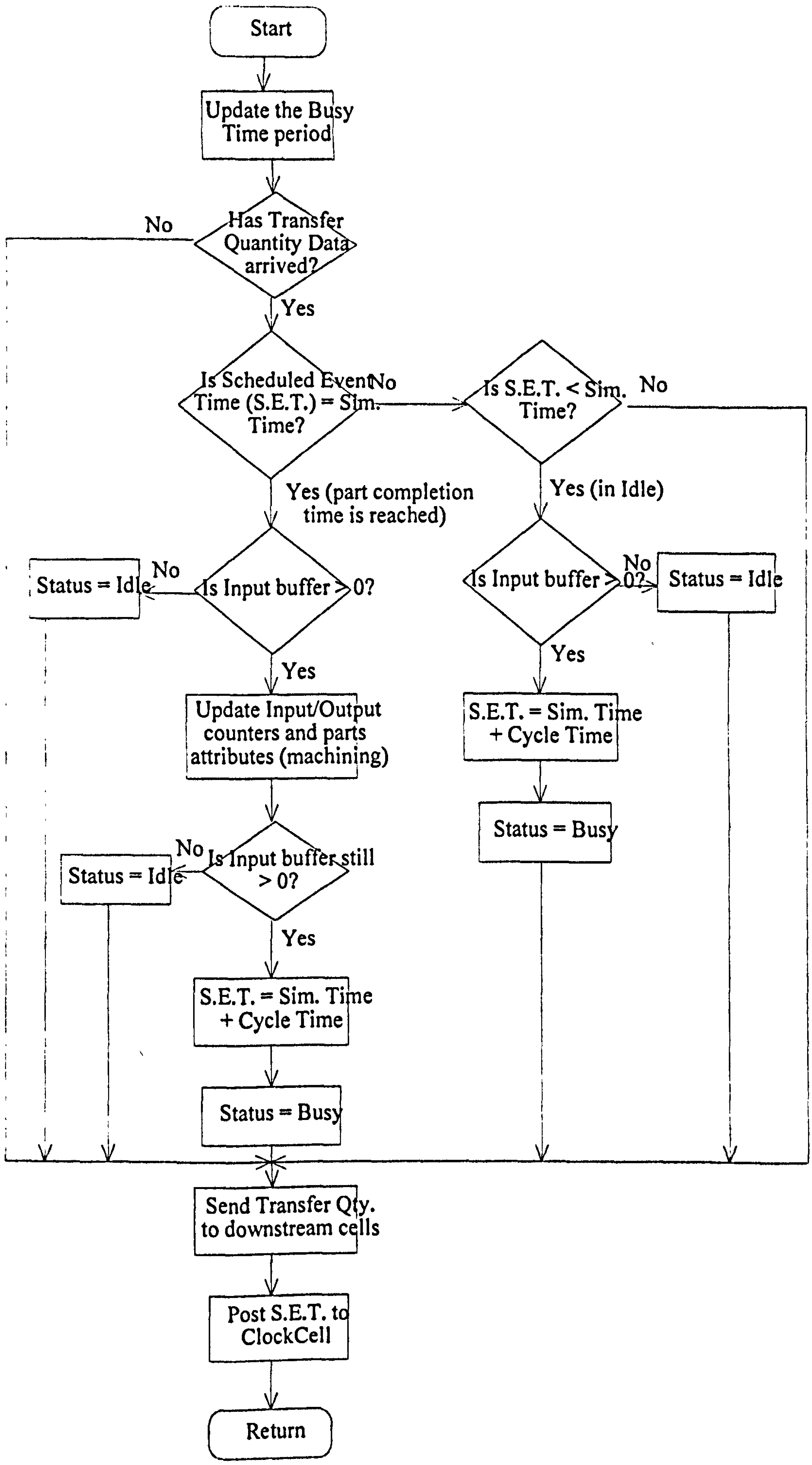


Figure 3: To determine the Scheduled Event Time of MCell

LOGICAL DESIGN OF VEHICLE CELL

Vehicles (such as Forklift Trucks, Automatic Guided Vehicles (AGV), and Cranes) transport parts around the production system. Number of vehicles, speed of vehicles and distance travelled between machines are inputted to a vehicle provider cell. Then, the vehicle provider cell supplies the vehicles to the required machine cell and calculate the time required for a vehicle to travel from one station to another (Figure 4).

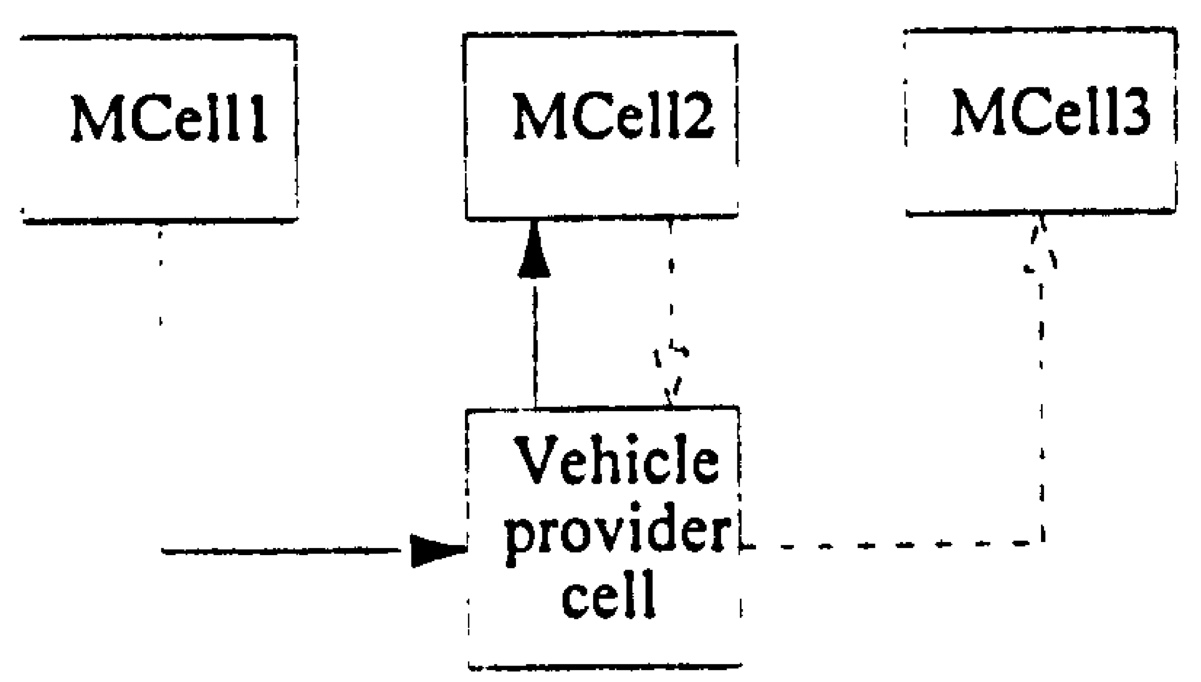


Figure 4: Vehicle provider cell

When a machine cell needs a vehicle for transporting a part to a downstream machine, the machine cell invokes a request to vehicle provider cell which checks the availability of the vehicle. The number of vehicle is kept track with an internal counter. If vehicle is available, the provider cell sends the sum of the completion time of the machine cell and the required travelling time to the destination machine cell. This machine cell then sends the data to the simulation clock. If not available, the vehicle provider cell sends a pre-defined dummy data to the machine cell indicating that the vehicle is not available.

RESOURCE CELL

Resource cell is classified as nonconsumable or consumable. Nonconsumable resources include labours, robots, tools etc. Consumable resources are raw material, finished goods stock etc. Generally, there is no need to establish a link between resource cell and simulation clock. However, raw material cell may be linked with the simulation clock in a push type production system to schedule the next arrival time of the part.

Machines either obtain the requested resource immediately or are delayed until the required resources become available. For example, when a machine cell requires labour(s) (non-consumable resource) to complete a job, a message is sent to the resource cell. If the labour is available, a pre-defined dummy data (lets say 0) is sent from the resource cell to the machine which then begins to process the part. The future completion time for this job is calculated and sent to the clock. After it finishes processing, the labour is released back to the resource cell which then increments the resource counter and the part is transferred to the downstream machine. If the resource cell cannot provide the resources, a pre-defined data (lets say -1) is sent to the machine which then keeps on request in the next cycle. Consumable resource acts like a non-consumable one except the used resource will not be released back to the cell.

CONTROL CELL

Control Cell makes decision and controls the logic of a system. When there is more than one production route available, the upstream machine cell consults the control cell for decision. In Figure 5, MCell1 asks the control cell whether the parts should be transferred to MCell2, Mcell3 or Mcell4. After getting the decision, MCell1 will pass the part information (TQD) to the destination machine cell.

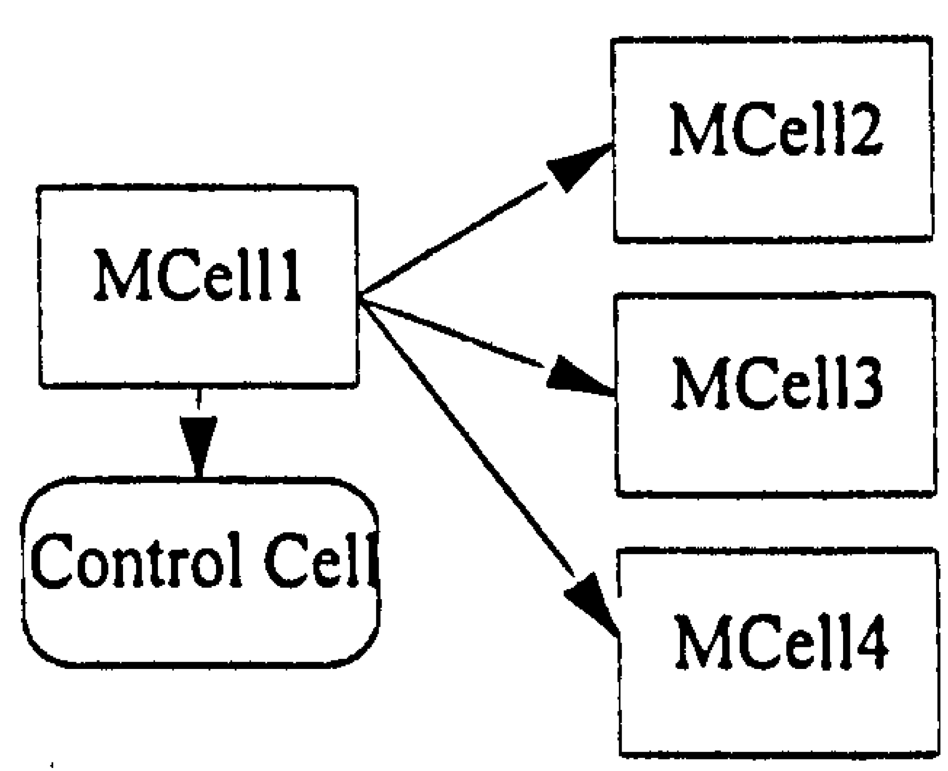


Figure 5: Control Cell for choosing machine

When there are more than one product type waiting to be processed, the machine consults control cell for decision. (Figure 6).

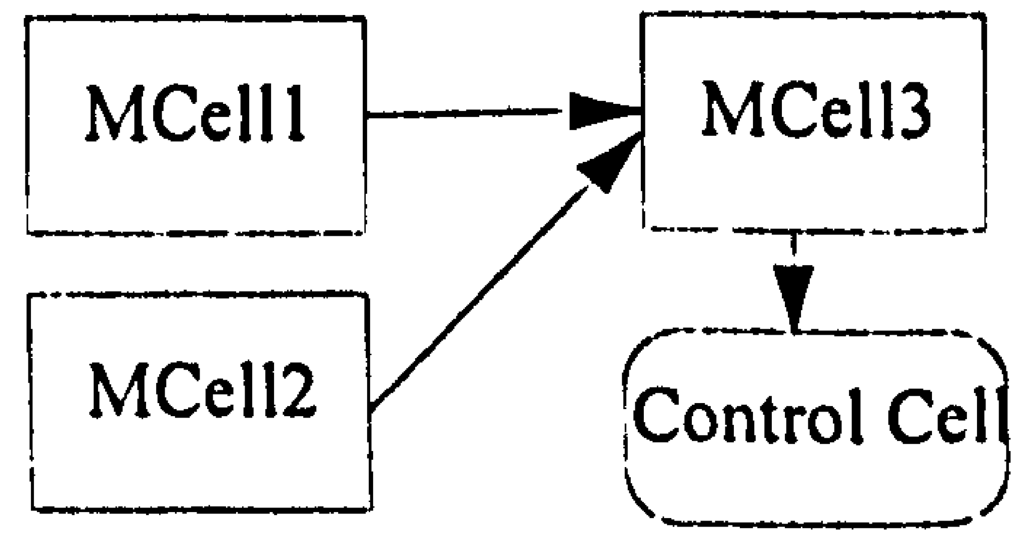


Figure 6: Control Cell for choosing product type

CONCLUSIONS

The development of Hierarchical, Modular manufacturing models within Windows multi-tasking environment on PC has been presented. A sample production model of eleven programs with nine machine cells, a simulation clock and a initialization cell has been built and tested. No significant sign of slowing down by the simulation clock (which is the potential bottleneck) has been detected.

Since Windows only allows a limited number of programs to be invoked at the same time, the maximize size of the model built is fixed. This can be solved by moving up the level of abstraction to shopfloor level. A shop itself may contains several machines, its own resources and transportation means. In shopfloor level modelling, each shop is a program. Several shop programs run parallelly under Windows multi-tasking environment to form a manufacturing system. The concept is basically the same as in the machine level but logic design within each shop needs not be considered in this level and only two type of cells, control cell and shop cell are required.

REFERENCES

[Cheung 1992]
S. W. Cheung, R. Roy and S. K. Bhattacharryya, "Development of a Hierarchical Modular Manufacturing Simulaiton Model," Proceedings of the Second International Conference on Automation Technology, edited by K. M. Wang, July 1992, pp. 155-160.

[Doyle 1990]
R. J. Doyle, "Object-Oriented Simulaiton Programming," Proceedings of the SCS Multiconference on Object-Oriented Simulation, edited by A. Guasch, The Society for Computer Simulation, January 1990, pp. 1-6.

[Kim 1988]
T. G. Kim, "A Knowledge Based Environment for Hierarchical Modelling & Simulation," PHD Thesis, University of Arizona, 1988.

[Knrnfeld & Gilhooly 1992]
K. Kornfeld and K. Gilhooly, "OOPS via DDE," *Byte*, vol. 17, no. 6, June 1992, pp. 145-154.

[Ladd 1990]
S. R. Ladd, "Objects of Attention," *Infoworld*, 30 April 1990, pp. 59, 60, 67.

[Zeigler 1987]
B. P. Zeigler, "Hierarchical, Modular Discrete-event Modelling in an Object-Oriented Environment," *Simulation*, vol. 49, November 1987, pp. 219-230.

[Zeigler 1990]
B. P. Zeigler, Object-Oriented Simulation with Hierarchical Modular Models : Intelligent Agents and Endomorphic Systems, Academic Press, 1990.

BIBLIOGRAPHY

C. Petzold, "Programming Windows," Microsoft Press, 2nd Edition, 1990.

"Microsoft Windows: Guide to Programming," Microsoft, 1990.

"Microsoft QuickC for Windows 1.0 Manuals," Microsoft Corporation, 1991.

APPENDIX

```
/****************************************************************

        FUNCTION: GoComp

        PURPOSE:  Determine the next schedule event time of the Cell
                  and Poke it to Simulation Clock. Also update the
                  statistical counters

****************************************************************/
void GoComp(hwndClientDDE, szItem)
    HWND    hwndClientDDE;
    char    *szItem;
{
        struct  ITEM * pItem;
        HWND    hwndServerDDE;
        HWND    *phwndTimeDDE = &hwndTimeDDE;
        BOOL    *pIniAdviseData = &bIniAdviseData;
        BOOL    *pStatus = &bStatus;
        BOOL    *pTransferClient = &bTransferClient;
        float   *pMct  = &Mct;         /* var. for Scheduled Event Time */
        float   *pCyct = &Cyct;        /* var. for process cycle time */
        float   *pClkt = &Clkt;        /* var. for simulation time */
        float   PreviousMct, PreviousClkt; /* var. for previous S.E.T. & sim. time */
        float   *pBusyTimePeriod = &BusyTimePeriod; /* var. for Busy Time */
        int     *pIn   = &nIn;         /* var. for Input buffer */
        int     *pOut  = &nOut;        /* var. for Output buffer */
        int     *pTransfer = &nTransferQty;  /* var. for Transfer Quantity Data */
        int      *pTotalIn = &nTotalIn;
        int      *pTotalOut = &nTotalOut;
        int     nIncrease;
        char    *pT = szT;
        char    szCellNumber[3];
        char    szCellName[4];

    if(*pIniAdviseData==FALSE)
    {
        *pIniAdviseData=TRUE;
        return;
    }

    pItem=FindItem(hwndClientDDE, szItem);
    strcpy(szTQ, "TransferQty");
    /* Store the previous simulation time */
    PreviousClkt=*pClkt;

    /* Update the Input buffer */
    if(!(strcmp(szItem,szTQ)))
    {
        nIncrease=atoi(pItem->szValue);
        *pIn=(*pIn)+(nIncrease);
        *pTotalIn=(*pTotalIn)+(nIncrease);
    }

    /* Get the current simulation time and update busy period */
    if(!(strcmp(szItem,pT)))
    {
        *pClkt=(float)atof(pItem->szValue);
        *pBusyTimePeriod= *pBusyTimePeriod +
                        (*pClkt - PreviousClkt)*(*pStatus);

        if(*pTransferClient==TRUE)
        {
                return;
        }
    }

    PreviousMct=*pMct;
```

```
    /* Check if Scheduled Event Time is equal to current simulation time */
    if(fabs((*pMct)-(*pClkt))<0.01)
    {
        if(*pIn>0)
        {
                if((*pClkt)>0.0f)
                {
                        /* Update Input/Output counters */
                        *pOut=*pOut+1;
                        *pIn=*pIn-1;
                        *pTransfer=*pOut;
                        *pOut=*pOut-1;
                        *pTotalOut=*pTotalOut+1;
                }
                if(*pIn>0)
                {
                        /* Calculate the next Scheduled Event Time */
                        *pMct=(*pClkt)+(*pCyct);
                        /* Change status */
                        *pStatus=BUSY;
                }
                else    /* Machine does not have next job yet */
                {
                        *pMct=0.0f;
                        *pStatus=IDLE;
                }
        }
        else
        {
                *pMct=0.0f;
                *pStatus=IDLE;
        }

    }
    else
    {
        /* Check if Scheduled Event Time < current simulation time */
        if(((*pClkt)-(*pMct))>=0.01f)
        {
                if(*pIn>0)
                {
                        *pMct=(*pClkt)+(*pCyct);
                        *pStatus=BUSY;
                }
                else
                {
                        *pMct=0.0f;
                        *pStatus=IDLE;
                }
        }
    }
    /* The Scheduled Event Time of the cell is not yet reached */
    if(!(fabs(PreviousMct-(*pClkt))<0.01))
    {
        *pTransfer=0;
    }
    /* Send Transfer Quantity message to client Machine Cell */
    MaybeAdviseData();
    strcpy(szSelectedItem, "t");
    sprintf(szSelectedValue, "%.2f", *pMct);
    /* Add cell number to the data message */
    strcpy(szCellName, "C");
    itoa(nInstCount,szCellNumber,10);
    strcat(szCellName, szCellNumber);
    strcat(szSelectedValue, szCellName);
    /* Poke schedule event time to Clock */
    hwndServerDDE=GetHwndServerDDE(*phwndTimeDDE);
    SendPoke(*phwndTimeDDE,hwndServerDDE,szSelectedItem,szSelectedValue);
    InvalidateRect(hWndMain, NULL, TRUE);

}
```