Control and Communication in Developing Countries: a cybernetic analysis and a proposed solution, exemplified by a distributed database system for the implementation of a National Plan.

A thesis submitted for the degree of
Doctor of Philosophy

Division of Cybernetics
Department of Electrical Engineering
Brunel University
England

By

SAMIR HUSSAIN AWNI
1985

Abstract

This thesis proposes some enhancements of the control and communication aspects of the implementation of a National Plan in any developing country. A cybernetic model for the viable system is used following the approach developed by Beer. Weaknesses of typical current practice are identified and a hierarchical organization, based on the model, is suggested as an alternative, in which all the systems implementing and monitoring the N.P. are considered as one single system, making its functions easier to recognize. Deficiencies, which are characteristic of the situation of developing countries, are identified and remedies are suggested to increase effectiveness. The model provides for the cybernetic principle of freedom, allowing the creation of autonomous subsystems with their own computational needs. A distributed system using micros, databases, and national communication networks, is described, which provides the requirements for realizing the suggested organization, together with packaged software to compensate for missing experience and know-how. The proposals are made in the form of a comprehensive package whose built-in complexity (sophistication) is very high. It is still effective even with inexperienced users but can take full advantage of their developing knowledge. This thesis includes listings and sample runs of some portions of the package which, for purposes of demonstration, have been implemented in dBASE II and an 8080 assembly program on an Intertec Superbrain QD microcomputer.

S.H.A.

1985

In Loving Memory

Of My Dear Mother

Acknowlegments

I would like to express my sincerest thanks and deepest gratitude to Dr. D. J. Stewart for his constant encouragement and guidance while supervising this project, and for his deep understanding and sympathy in the personal problems which I faced during my work.

This work would not have been possible if it had not been for my late mother's sacrifices. She chose to die alone in our country rather than for me to stop work and be near her in her illness. My thanks also go to the rest of my family for their constant moral support and encouragement.

My thanks are due to my sponsors, the Iraqi Government, and to my colleagues and friends at work, whose belief in me was encouraging and motivating.

Finally, I would like to express my gratitude for the help of all the secretaries in the Department and the operations staff of the Computer Centre.

S.H.A.

1985

CONTENTS

I. INTRODUCTION

The aim of this thesis is to propose a comprehensive solution for problems facing developing countries in their implementation of a National Plan. These problems are typically found in the area of control and communication. Cybernetics offers principles and tools which ensure the comprehensiveness of such a solution.

The solution is mainly proposed for countries which are politically termed 'third world' countries, but which we chose to call 'developing' instead, since we are not examining their situation from the political point of view. Choosing this term however might also prove controversial, in that 'developing' is also an aspect of the industrialized countries, in addition to the fact that it is a natural process in any viable system. It is necessary for the chosen term to reflect our description of the situation under study in that: (1) these countries are not technologically advanced, and lack the necessary experience to deal with complex systems, and (2) although development is a natural, and therefore an inevitable, process, these countries aim to enhance their own development by planned intervention. As the term 'developing countries' is internationally used, and understood to mean non-industrialized countries, then using this term excludes the industrialized countries. The reference to a National Plan necessarily completes the description by implying intervention by central government.

The encountering of problems by these countries, in the implementation of planned progress, is evident from their failure to achieve the anticipated rates of progress. It is assumed by this work that the planning itself is done on a sound basis. What remains therefore is the area of realizing this planning which involves managing its implementation. A common observation is the complete reliance of these countries on orthodox management in steering a highly complex task (i.e. the Plan), which is an understandable approach since cybernetics is a fairly unknown discipline in these countries. The realities of the implementation revealed a multi-dimentional problem which involved, in addition to the straightforward ´directing´ of activities, a diversity of components (managerial, technological, psychological) which almost paralyzed any counteraction, and at best rendered it ineffective.

Factors necessary for the successful intervention, which are built into the cybernetic approach and which influenced this work, are reviewed in Chapter 2. We anticipate that this review will help in introducing cybernetics to people in authority in developing countries, so that they will appreciate the advantages of cybernetics. The historical review shows the origin of this science and its natural revelation which accompanied the increased complexity facing Man, as technology grew faster than the rates he is accustomed to. The unified approach of cybernetics underlines its belief in the unity of principles and concepts governing world activities, which resulted in its real intellectual wealth of realizing that both natural and artificial systems can be analysed in terms of the same cybernetic language, that of control and communication (or information). The cybernetic principles which will be mainly involved in this work are explained in a manner which takes into consideration their abstract capacity, but with a view to implementing them in a situation where the human is the main factor and the centre of attention.

The importance of the role of the ´observer´ paves the way for the issue of the ´purpose´ of the system and for the realistic view of intervention, especially between a human and another system, that is characteristic of cybernetics. The concept of variety helps in explaining, and understanding, the complexity of dynamic systems. Variety also helps in inferring the meaning of messages, if it can be used in Weiner´s formula (1948) for the gain of knowledge. The problem here is that the basic formula for measuring the gain of knowledge was produced by Wiener before Ashby´s ´variety´ concept came forth, and was derived from the ´time series´ application. Ashby for his part explained his concept in terms of ´sets´. He found

Shannon's formulation completely compatible with his 'sets' approach, but did not show how it can be applied to inferring the meaning of messages. This area therefore remained wide open with the failure of Shannon's formula to go beyond the decoding of a closed, well-defined and disciplined set such as that of 'language', to an application with a highly dynamic and observer-biased set such as that of the variety. Wiener's formula on the other hand is completely compatible with Ashby's variety with respect to concept.

All the cybernetic principles reviewed in Chapter 2 will be embodied in the main analysis and proposals of the thesis, as they influence each section appropriately. This is the approach by which the situation of developing countries is analysed to identify its main characteristics as a first step towards identifying the components of the problem we are aiming to solve. Three fronts therefore have to be examined: the technological, the management, and the viability fronts. From this, what will be involved, and what is needed to be basically done, is outlined by identifying the special weaknesses and shortcomings which are characteristic of the studied situation on each front.

Technology has contributed hugely to the advancement of the industrialized countries; therefore it has to be a decisive front for any country. The new technology also contributed to speeding-up that advancement, which has led to yet further increase in the gap between the developed and the developing countries. There were reasons behind the slow arrival of technology in the developing countries, for example its expense.

Information technology is the core of new technology, and with it came a change of attitude, style of management and a whole range of intervention techniques. This fact cast a shadow on the management front, which is witnessing the helplessness of traditional methods in the face of even faster rates of change than those experienced by the developed countries when they first used them. Some of the traditional solutions sought in this area are proving more damaging.

Being developing countries means a dependence on the advanced countries and an inevitable influence exerted on them from these countries. This inevitability introduced some facts of life and human attitudes to the management area, which are acting in a rather different manner than in the advanced countries. The need therefore is for a rather clear-cut intervention which is done with clear procedures and techniques, and is not dependent on management by dictation. On the third front arises the issue of utilizing existing wealth, which is tightly connected to the two previous fronts. Wealth is essential to acquire the needed technology, and should be at least maintained if it cannot be increased. But the failure on the management front is negatively affecting this utilization.

The arrival of new technology had a big impact on the developing countries. The stages of development which this technology had gone through were accompanied by a gain of experience, in handling and dealing with it, by the developed countries. These stages meant a gradual change and a gradual build-up of experience in utilizing this technology, and in managing the new organizations using it. (Some aspects of this experience, especially those which affected the society negatively, are not necessary—in fact they are better kept away—but the purely technical aspects are essential.)

It is clear at this point that the viability front has caused the transfer of highly technical people to the management area—a move which deprived the technological area of people who are essentially qualified and able to deal with the new technology, but at the same time are not trained in managing it, which is what they are supposed to do now.

Cybernetics is available to improve managerial techniques and to provide alternatives to that experience which was never built up. Examining the issue of ´experience´ shows that the principle of intervention, which is identified and explained by cybernetics would essentially lead to an alternative to the missing experience. Another alternative would come from the use of software packages which can be regarded as experience in kit form.

The answer to providing new technology starts with the issue of 'information technology'. This then is divided into two main components: the computer as a data processing machine and Information Systems. The computer, although a machine, has many traits that will compensate for its lack of human intelligence. Interfacing the human with this machine in a designed and disciplined combination can certainly amplify the powers of the human.

The rapid development of the computer meant its inevitable involvement in industry and management, thereby resulting in further improvements to the intervention techniques in these areas. The microprocessor is the recent development in computer technology which has changed, yet again, the approach to computerization. With it, processing is becoming easier to distribute, and the limitations of, and difficulties associated with, centralized processing can now be overcome. The proliferation of the powers of the micro is offering the managers more solutions for their present difficulties, at affordable prices. With this, a whole range of application packages have become available, again with negligible prices compared to those of the mainframe. These are essentially up-to-date techniques which are the result of a long and well-tested experience in the field of application concerned, which is the very thing needed by the developing countries.

It is appreciated that solving the problems of hardware and software that are contributing strongly to the intervention issue, is only one-half of the solution. It is only when they are used successfully that their full impact is felt. Developing an effective Information System is a major task in any project aiming at improving the intervention process, which would definitely increase the knowledge of the user, and help a great deal in producing the correct reaction to changes. A major concern here is that information could also be a disturbing influence if there is no control of what enters the system. At the same time, it is beneficial for the system to accept information which is not related to its activities, because indications for future trends might be gained from analysing it. To allow this and at the same time ensure the control, the ´irrelevant´ information is not rejected immediately from the system, but channelled instead to a special unit for further processing.

Analysing the information entering any system is an essential step to building an effective information system. This will lead to ways of reducing the information volume and at the same time identifying its relevency. There are many ways and techniques to follow. This analysis deals also with other areas, among them: the monitoring of the information flow with an aim to remedy faults; the learning with an aim to gain proper experience; and the envisaged future needs for present information with an aim to build the system databank.

To bring the situation as close as possible to reality, we consider an illustrative application which is taken from real life, to be the focus of our attention while analysing and suggesting a proposal. A real-life system which is the closest to our task is the Central Plan Follow-up System in Iraq (CPFS). The illustrative application is based on this system's 'idea' rather than its finer details, in order to help in explaining and appreciating the various suggestions made here. The 'idea' of the CPFS is based on the traditional organizational approach and probably is second-to-none within this context. The main aim of this system, and what concerns us most, is to perform the fastest possible information-gathering regarding the performance of the implementation (and the executors) of the National Plan. This is to help in reducing the time lags that are affecting the efficient intervention of its users. At the same time it is essential that this system should not be a source for human conflict (as is currently the case) as we shall gather from the organizational analysis.

Satisfying both these aims leads to the issue of 'organization'. The starting point of organizational study is the identification of the ´system´. This is a controversial issue as to whether or not it is necessary to identify the ´whole system´ before starting work on it. Cybernetics takes the view that such identification is practically not achievable, while the more practical aim is to achieve a partial knowledge that is complete within itself and is sufficient for the worker´s ultimate practical purpose. This approach leads to a practical ´definition´ on which the organization model is based. Considering cybernetic principles individually, and suggesting as a result remedies for organizational problems, is not sufficient for a cybernetic proposal. All the organizational elements causing problems must be considered from the point of view of their dynamic interaction with all the other elements of the organization. The model of the ´viable system´ is the right proposal to produce the comprehensive solution, i.e. a cybernetic solution. This model is based on the identification of the ´functions´ of the viable system. These functions (i.e. policy making, intelligence, control, co-ordination, and implementation) are situated in two distinct philosophical regions of the system, namely, the INSIDEandNOW region and the OUTSIDEandTHEN region. They are the result of the interpretation of the ultimate task of a system in maintaining internal stability with an eye on the outside. The organization concerned can therefore be analysed, and reinforced as a result, according to this model. There are four ranks (recursions) envisaged for the organizational structure of our System. The breaking-up of subsystems at each rank (i.e. the implementation systems of each recursion) is based on the specialization, then on the scope of operations. The result of such modelling helped in recognizing a major problem, that is hidden by traditional organizational modelling. The subsystems of the System are necessarily superimposed on various other subsystems in different systems. As a planning-related system, together with its human elements, are all abstraction-orientated with a purpose to preserve the National Plan intact. All the projects of that National Plan, on the other hand, are realized by different systems which, together with their human elements, are real-life orientated with a purpose to conform the Plan's abstraction to the reality of the implementation. These purposes being incompatible, together with the fact that we have two differently natured systems on our hands, makes the inevitable involvement of our system in the affairs (and the operations) of the other systems—a major source of problems. To reduce the negative effects of such involvement needs the reduction, as much as possible, of direct human contacts with the other systems, especially when those contacts result in an intervention by a lower ranking human in the work of a higher ranking one.

The final modelling of the System essentially results in providing for cybernetic freedom for its subsystems. Such freedom would lessen the amount of intervention by the metasystems. This is a positive measure which is both helpful in spreading the responsibility and reducing its burden and also results in more realistic intervention as the source of this intervention becomes closer to what necessitates it. (The disturbances, on which the subsystem is given the authority to act alone, are those which are typically the property of the subsystem itself—they are originating within its boundaries and containing them is also done within these boundaries—and therefore the directorate of this subsystem must be more familiar with their components and have sufficient scope to deal with them.)

The establishment of autonomous subsystems suggests the presence of local computational needs. As was pointed out earlier, the microcomputer is becoming powerful enough and cheap enough for it to be feasible to provide every one of these subsystems with a local processing facility. This then lays the foundation, by computerizing the whole system, for linking the whole System electronically. The solution for the problem of the human involvement and conflict can then be achieved by extending our System ´artificially´ to the other systems by means of an electronic communication network, thereby eliminating the direct human contacts. This should then: (1) eliminate the human conflict, (2) speed up the information-gathering, (3) automate the internal tasks of the subsystems, (4) open the way for the use of packaged software applications (i.e. using tested experience), and (5) offer efficient control and communication inside the system.

Achieving this task means that we will be dealing with, and viewing, two different systems which share the same ultimate purpose with one of them superimposed on the other as a ´single system´.

The two main components in the realization of this suggested organization are the information handling and the communication structure. Analysing the types of information involved in the operations of the System helps in deciding on the way their storage has to be organized. This analysis shows two distinct types of information: ´local´ and ´global´. This arises from the need of the subsystems to maintain their operations (viability), and hence the existence of a distinctly local type of information; and from the need of the System as a whole to keep the inter-subsystem interaction dynamic to ensure the holistic effect, and hence the existence of a distinct global type.

The stream of this information will enter the System from two directions: either from its highest rank or from its lowest rank, and spread inside throughout the ranks. The number of terminal points (i.e. the subsystems) at the lowest rank are considerably more than those at the highest rank. The volume of information entering from this rank, which can be handled by its large number of terminals, has to be gathered at the upper ranks' terminals. This volume can be reduced via various techniques without affecting its informative power to the respective rank. Several benefits point here to the alternative of 'storing' the relevant information, to each subsystem, at the site of that subsystem, i.e. spreading the information over the system. This being the case, a major problem arises in the area of preserving the integrity of stored data due to the presence of global items that are stored at more than one site. The updating of any global item must be propagated to cover all its replicas. The global commands should be capable of ensuring this update. This propagation is also needed to channel information in general, and the control and co-ordination messages in particular, and therefore demands the establishment of communication links between the various subsystems. The model of the viable system determines these channels, and has to be followed therefore to ensure the viability of the suggested organization.

Two main types of connection result from the functions of these channels: 'vertical' and 'horizontal'. The vertical direction will serve the channelling of the command-type messages, and also handling the main information flow entering the system from both directions. The horizontal connection will serve channelling of the co-ordination-type messages, plus a minor transfer of information items.

With the establishment of this network of connection channels between the subsystems, a wider door is opened for utilizing the computer powers of automation that can improve the overall performance of the System. One aspect made possible by this is the automatic relay of messages, which when examined closely shows a proper utilization for the command, monitoring, and co-ordination channels which are provided by our terminal organization and communication network.

We have to look at (and treat) the two, differently natured, organizations (i.e. the real-life and the planning-orientated) as ´one system´, because in effect that is what they are. Organizational necessities demanded that they had to be two separately organized systems. In reality, only the first rank of the CPFS has to exist physically. All the other ranks are information-gathering points. In the ´implementation plane´ however, all the mapped establishments (i.e. those excuting the Plan) have to exist physically. The two existing organizations can go different ways but they have to meet at a point where the ´ultimate purpose´ is common to them both. This point will necessarily exist at the highest-most ranks of these different organizations. If we consider the ´abstraction plane´ and the 'real-life plane' together with the 'Planning Board' (this is the highest ranking—in authority and power—body in the country with respect to planning) and examine their situation, we observe a situation similar to that existing within the metasystem region of the viable system. Utilizing Beer's terminology, which is explained in full in Chapter 4, the Planning Board corresponds essentially to a System Five function (i.e. policy making), while Rank 1 of the Plan (CPFS) is heavily involved in a System Four function (i.e. intelligence), and the implementation establishments (which would be mapped at the same rank) are heavily involved in a System Three function (i.e. control). (The interaction between them is necessarily similar to that explained by the model.) We have used here the term 'heavily involved' since each of these organizations is not, as they are now, exactly mapped at the function concerned. Ideally they should be, but practically they are not, due to the existence of a well-established organizational structure. Each of them is heavily biased towards one function and performing part of the other. This is due to the 'forced' breaking-up of the natural organization, with each side having the necessary components to perfom what they are actually performing now. In our approach we are considering them together (in their natural form) and, with the help of the model, we will channel each component through its appropriate function. The principle which we have to consider carefully at this stage is that, while Rank 1 of the Follow-up has the authority to reach any information in the other ´plane´, it should not issue commands to the lower ranks there. These commands have to come from the higher rank of the intended subsystem. This condition correctly implies that there is the possibility that the command originating from the Plan´s rank does not have the approval of the implementation rank responsible for the subsystem receiving the command. The establishment of the electronic communication network offers a realistic link which can serve the various requirements resulting from this constraint. The final connection at the top rank can be made in duplicate. A ´four-cornered´ link can then be established with the Planning Board, the implementing Ministry, the Sector, and the Field establishment (which belongs organizationally to the Ministry) at each corner. The effectiveness of such a link, in joining the two planes, is demonstrated by two typical cases of intervention: (1) each Sector has to know how its subsystems are performing, and needs to intervene when it is necessary, and (2) a subsystem (say a Project) faces difficulties, its Field decides to intervene but needs the authority from its superordinate (i.e. the action required or suggested is outside its authority). In the second case, the Field sends the suggestion to its Ministry (outside our System), but the established link forces a replica of the request to go to the corresponding Sector. In both cases however the Sector becomes aware (through its links) of the need to intervene and has its own opinion about what this should be. The Sector and the Ministry then exchange opinions (interact) where they may: agree, in which case the final decision will reach the Field from its links with the Mininstry and the Sector; or disagree, in which case an exchange on the triangular link between the Ministry, Sector, and Board, is activated under the control of the Board. This would result in the final decision which is to be sent to the Field. From there on, the flow will be effectively in a straight line to the other ranks.

The whole suggested proposal here has the necessary ingredient as whether to implement it manually or with the aid of todays technology. It is obvious though that the manual implementation is not as efficient as, and not as comprehensive as, when todays technology is employed to the full. (A partial use of technology, i.e. the use of microcomputers only, without the communication network, is possible but it will be understandably less efficient than the full use. The 'floppy disk' of each branch site can be sent daily, by post or by a courier, to their node site.)

The latest developments in the technology of communication networks and databases evolve naturally towards the concept of 'distributed' processing and storage of information. These developments will support efficiently our proposals´ requirements of local processing capability, local data storage capability, and proper communication links between the subsystems. Two main benefits are envisaged from this concept: (1) it allows for the freedom of the subsystems, and (2) a total failure for the whole system is rare.

As the target System is a very large one, with all its complications, it might be difficult to show the user the suitability of the proposals presented to him. At the same time, the system concerned is such that any simulation of it cannot be but a realistic package. Such a task would be understandably huge. Nevertheless, our own belief in the validity of the principles, which are incorporated in this proposal, and the comprehensiveness of the proposed solution may not be enough to convince the user about its validity. Demonstrations of the capabilities of the microcomputer to perform typical applications, which are currently run on mainframes, are therefore required to show the feasibility of replacing the mainframe with micros, as is here being suggested. (This demonstration would understandably stop short of covering all the envisaged activities—the limitation of the resources available for this research contributes to this as well.)

It should be emphasised here that the proposed solution is capable of being implemented in ´complete portions' as upgrading stages, in proportion to the required complexity for the System. (An exception which has to be implemented immediately is the interface between the Sectors and Ministries.) The complete System should then evolve over several years, and adapt naturally to the user environment.

II. THE RESURGENCE OF CYBERNETICS

2.1. Cybernetics

The origin of the word 'cybernetics', in the sense we acknowledge today, goes back to 1948 when Norbert Wiener published his book of the same name, in which he described how four years earlier, a group of scientists, among them Wiener himself and Rosenbluth, had became aware of the unity of the set of problems centering around communication, control, and statistical mechanics, whether in the machine or in living tissue. They decided to coin an artificial neo-Greek expression to fill the gap in existing terminology which has been heavily biased to one of these fields or the other. 'We decided to call the entire field of control and communication theory, whether in the machine or in the animal, by the name Cybernetics, which we form from the Greek χυβερνήτης or steersman.' Wiener stated in his book. He also goes on to acknowledge the work of the British scientist Clerk Maxwell and to recognize his paper ' The Theory of Governors' published in 1868, to be the 'first significant paper on feedback mechanisms'. He also acknowledges that the term 'governor' is derived from the same Latin version of the same Greek word, but was unaware that the word 'cybernetics' had been used, in a limited sense, a century earlier by André Ampère (Pask, 1961). Wiener also gives a second reason for choosing this term as the fact that 'the steering engines of a ship are indeed one of the earliest and best developed forms of feedback mechanisms'.

Cybernetics looks at the world with a unified approach underlining its belief in the unity of concepts and principles governing its activities.

The fact that feedback plays a crucial role in the proper functioning of a control system led Wiener and his group of scientists to the belief that, in order for a person to perform motor activities, his cerebellum must embody types of feedback and associated information processes comparable to those used in an artificial control system.

This notion led to the conclusion that the brain could be viewed as a complex communication, computer, and control system; and the concept of feedback and control could account for internal homeostatic control of the vital operations of the human body. This being so, one can learn further from the human body (including the brain) by interpreting its activities and problem solving behaviours to improve the performance of the artefacts.

Although these analogies between the human brain and the artefacts were fruitful, the real intellectual wealth of cybernetics lies in the realization that both systems, natural and artificial, can be analysed in terms of the same cybernetic language, the language of information and control (MacKay, 1957).

A major force behind the development of cybernetics, apart from Wiener's work, was Shannon's theory put forward in his paper 'The Mathematical Theory of Communication' published in 1948, which proved, mathematically and practically, the notion of 'entropy' in information which was already established among cyberneticians. Another major force was the digital computer which was followed by swift advances in computer theory, technology, and applications, from its introduction in 1946 until now. The computer, apart from being a descendant of the cybernetics philosophy, was to play a crucial part in the further development of cybernetics itself by providing the processing power to be used in its applications on one hand, and on the other in improving the technology itself in an ever growing cycle that proves by itself once again that it is on the right tracks.

The work of cyberneticians after Wiener varied from giving emphasis to abstracting a controllable system from the flux of a real world, like Ross Ashby in 'Introduction to Cybernetics', to looking upon cybernetics as the science of proper control within any assembly that is treated as an organic whole, like Stafford Beer (Pask, 1961).

Beer's work also widened the field of application for cybernetics by tackling the awkward problem of management, thereby opening the way to the involvement of human behaviour and psychology.

A very important issue discussed earlier by Ashby, that of the 'observer' of a system, was later expanded by Beer to cover the 'purpose' of the system. Beer points out that the purpose of any system is defined by the observer of that system, and that it is quite probable that no two observers may agree on that purpose, a situation which accounts for 'unresolvable' disagreements about systems. A disagreement about the system's purpose means no agreement on its boundaries, a fact which leads to doubts whether or not we have actually recognized the system at all. Beer's answer to this conflict (The Heart of Enterprise, 1979) is that we have to agree on the 'convention' about the nature, the boundaries, and the purpose of any system before we can agree on what is to count as a fact. But Beer's answer fell short of providing a cybernetic model on which we can identify the various conflicts in an attempt to resolve them whenever that convention fails, or at least to reconcile them to reach the convention. The convention can easily fail in the present realites of the world and many examples can be referred to here of irreconcilable differences of purpose that can exist for a system. But the point is that cybernetics is about reaching a stable state—it is not about removing conflicts. Thus, recognizing these conflicts is a requisite to finding a common stable state that the system can assume if it is to survive—which must be a common purpose for all the observers (and it must have been the origin of the conflicts in the first place). The need for a cybernetic model is therefore essential for recognizing them and finding the required common stable state. The observer´s ´purpose´ is a result of his own personal preferences and interests and these are essentially human powers of the highest order. Dr. David Stewart of Brunel University (Stewart, 1982), puts forward the principle of ´ternality´, and a theory for a cybernetic model that can be used here. This theory classifies (separates) the human powers into three domains. The first is the ´primary domain´ where the physical human powers and their artefact extensions are situated. The second is the ´secondary domain´ where the information processing powers are situated. The third, and the required one in our case, is the ´tertiary domain´ where the powers of preference and choice are situated.

2.2. Feedback and Control

> ´We thus see that for effective action on the outer world it is not only essential that we possess good effectors, but that the performance of these effectors be properly monitored back to the central nervous system, and that the readings of these monitors be properly combined with the other information coming in from the sense organs to produce a properly proportional output to the effectors. Something quite similar is the case in mechanical systems.´
>
> N. WIENER, 1948.

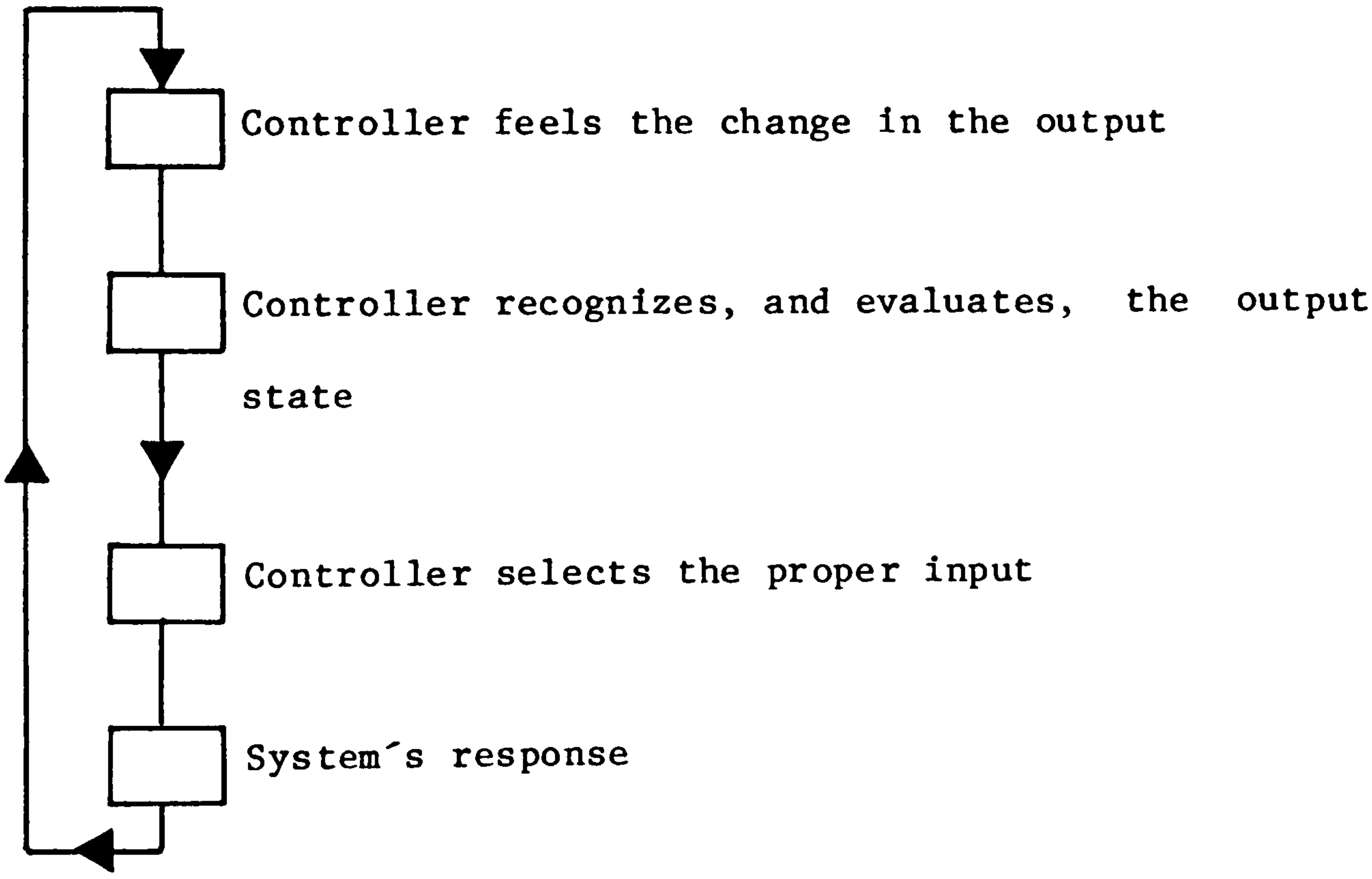
Control lies at the heart of cybernetics, and feedback, as promoted by cybernetics, is the principle of proper control. Feedback itself is said to be present when a circularity of action exists between the parts of a dynamic system consisting of two parts (subsystems), or more, with their own mechanisms which are coupled together so that each affecting the other (Ashby, 1956). The basic principle in the feedback concept is therefore that of ´circular causality´.

If the interacting parts, between which a feedback is present, are affecting each other negatively, i.e. each part is opposing the behaviour of the other, then this is said to be a negative feedback. To picture this situation in terms of actions with reference to a point representing the favourable state, Milsum (1968) sums up a non-mathematical definition from Maruyama's mutual causality approach (Ch. 5 of the same reference) that Negative Feedback equals: 'Deviation-counteracting (or Attenuating) Mutual Causal Loop'. Such feedback is evident, and favourable, in any system trying to be stable around a state of equilibrium. But when the parts are encouraging each other's behaviour, then this is termed as a positive feedback, which is according to Milsum (1968) equals: 'Deviation-amplifing Mutual Causal Loop', where the point of reference is always the immediately previous level of performance. We can safely label these loops, once identified, as either positive or negative knowing that they will continue to function that way, by nature, until they cease to exist.

From the nature of these circuits also, a perfectly representative theme can be associated with each of them for the sake of simplicity, like 'learning' with the positive feedback circuit, since it is encouraging the past performance, and 'stability' with the negative feedback circuit where the system is continually brought back towards an equilibrium state.

It must be noted here that the deviation sign in these circuits has no bearing on their types, because in both types the deviation could be positive or negative so that, say, in the positive circuit the deviation may be either positive or negative, resulting in an 'upgrading' or a ´rundown´ situation respectively, while the circuit is still positive.

The main recognized reason for the deviation in the feedback circuit is the time delay between the actual action of the controlled part and the reaction of the controlling part to it. In between these two landmarks there is the point of feeling the change in the output to mark the start of the following continuous loop of actions:



This would also explain the ´oscillations´ in the human institutions´ performance referred to in several writings (especially where there is a supply and demand situation). Several cybernetic machines can be referred to here (e.g. steam engine governor, thermostat, etc.) which show a minimum, if not negligible, deviation because of the proper application of the feedback principle. A very important issue which must be pointed out here is that of the information (to be discussed latter in length) used by the cybernetic machine. In the steam engine governor, for example, the information used was that of the 'pressure', which is direct information that contributed to cutting down the time required for the first two stages of the above loop and, more important, was an essential factor in effecting the controller reaction. It is also true that getting the direct information is not always attainable, in which case another, but correctly correlated, information must be found, otherwise the response of the system would not be what it should be. The further away the chosen information is from the source to be controlled, the less efficient will be the feedback circuit.

The feedback circuits may not be found in the systems standing in isolation from each other but, rather, they tend to form a complex web of interconnections that would result in very complicated connections, in which the behaviour of the system would not be the straightforward sum of these effects. As a result of this, the concept of feedback 'becomes artificial and of little use when the interconnexions between the parts become more complex' (Ashby, 1956), at which point we should abandon the attempt to control such systems via its individual circuits, 'Such systems cannot be treated as an interlaced set of more or less independent feedback circuits, but only as a whole.'

(Ashby, 1956). The controller now has to move to the 'black box' technique in which all these single circuits are rounded up (artificially) in a single box, and the controller couples himself to it to form a new feedback circuit with a simple connection. By doing so we are simply abandoning one level of details to one above it, but the whole effect (behaviour) of the system is still observed with the same clarity. This approach should be enough since the controller's objectives are always to control that effect and not that of the individual circuits. As we have seen with the feedback circuits before, the basic operating principle was in effect an analogous one, while with the black box it is not exactly so, for we are aiming here at an exact output initially, to be maintained, up and down, in the same manner as a single feedback circuit. The box's behaviour is observed for a substantial period of time to deduce a relation between its inputs and its outputs, i.e. what input (normally) gives that output, and proceed with control on the basis of 'to get that output we have to use this input'.

Even though the techniques are not exactly the same, the black box may still behave in the same manner in which the basic feedback circuit behaves (since the basic principle is the same), so that we may have a negative or a positive loop effect from the black box as well. At this point we can explain some seemingly strange behaviours for the black box, in the case of the economy for example and in particular the trends of ´inflation´ and ´hidden unemployment´. A positive loop for example may latch itself, despite human intervention, to produce an unfavourable overall result (like inflation) simply because a sufficient number (to force this outcome) of its inner loops are acting favourably within their own local connections.

The black box technique is intended to be used with those systems where we cannot reach their internal patterns of connections for one reason or another, but this will not be exactly true after a while. Indeed, after some experience in manipulating the system as a black box, we might be able to infer a pattern of internal connections inside that black box. When this happens, it would be wrong to assume that we have finally found THE pattern of that system, since ´which pattern of connexions will be found depends on which set of inputs and outputs used´ as explained by Ashby (1956). This explanation is supported by Wiener, who cited quantum mechanics in which the whole past of an individual system does not determine its future in any absolute way but merely the distribution of ´possible futures´ of the system, and giving the reason as ´in general there is no set of observations conceivable which can give us enough information about the past of a system to give us complete information as to its future.´ It follows, therefore, that in reality we have managed to manipulate one, or few, possible futures of that system. If we know this and are still content with it, then we may proceed with Beer's suggestion by treating the black box as a 'muddy box'. This new technique is especially important in managerial systems where, if one 'possible future' is favourable and its connection pattern is found, then it can be used safely as a template in investigating similar systems, or even in designing new systems with the same pattern and letting the unknown patterns develop themselves through time.

2.3. Information and Interference

If, as we have seen before the feedback circuit is the principle tool of control, then information is the blood of that circuit: if it is not flowing the circuit is not functioning.

Wiener and Shannon agreed on the definition of information as 'That which remove uncertainty' but differed conceptually while deriving the same formula for its measure. All writers later used a variation of the word 'remove', like 'reduce' or 'change', in an attempt to bring about a version of that definition which might be more immediately representative, conceptually and practically, in their field of application.

Because of the importance of information per se to cybernetics, the communication theory developed by Shannon and Weaver in 1948 was a signifcant contribution to cybernetics. It put forward an algebraic formula for measuring the 'amount' of information in a message (i.e. the efficiency of decoding with respect to encoding the message), which is [- p log p], whose units are the 'bits' (short for BInary digiTs), where a 'bit' is the information involved in giving the answer to a single question whose answer can only be either yes or no. Shannon found that his formula is identical to one of the most important formulas in the physical science, that which embodies the Second Law of Thermodynamics, namely 'entropy', except for the minus sign. The reason why Shannon added the negative sign to his formula lies in the way he defines that amount. This discovery came at a time when ´the notion of the amount of information attaches itself very naturally to that of entropy´ (Wiener, 1948) had already been established among cyberneticians. Entropy, in thermodynamics, is measured as [p log p] and it is necessarily negative. The Second Law of Thermodynamics states that, except in special circumstances, any orderliness there may be in a situation tends to decrease, so entropy tends to increase—thus entropy is a measure of disorder. Information on the other hand,is a measure of order, quite the opposite, since the amount of information in a message can only be dependent on the degree to which the sequence of letters spelling it out is orderly (not random). This fact made Shannon determine that the more information, the less entropy, hence the negative sign and a new term ´negentropy´. Wiener, on the other hand, did not add the negative sign to his formula, nor did Ashby who pointed out that the basic concern is the ´gain´ in information from a message after it arrives at its destination and not the absolute quantities present before or after.

Wiener and Shannon had worked concurrently on the problem of information measurements, and each of them did acknowledge the contributions of the other to this field. Shannon´s early work on switching and mechanical mathematical logic had preceded Wiener's work in addition to his independent development of the entropic ideas, while Wiener takes the credit for the basic philosophy involved in this field. Shannon had naturally been especially concerned to push the applications to engineering communication, while Wiener had been more concerned with the biological applications (central nervous system phenomena, etc.).

The enthusiasm generated by Shannon's theory led some of it's followers to regard it as THE information theory, thereby attracting serious criticism which can only be suitable if this was indeed a theory of information and not, as was the case, a theory for engineering communication. This also led to the wrong interpretation of the term 'amount' as incorporating the 'meaning' of the information to be measured, and led the critics in turn to treat it as such, and attracted the kind of criticism made by the British Professor of Animal Genetics, C.H.Waddington in his book 'Tools for Thought' (1977), where he argued successfully that a change of two letters in a certain message would alter its meaning entirely while 'for Information Theory the difference rests only in the letters third from the beginning and second from the end'. But Shannon's was an engineering communication theory about the transfer of a message from one point to another and how much that message will remain representative of its original copy (i.e. the 'meaning' is not a property of this concept), with a specific object to find the amount of information involved in that process, in which Shannon and his co-contributor Weaver disclaimed explicitly any concern with its meaning. In fact Waren Weaver wrote later, in 1962, that ´An engineering communication theory is just like a very proper and discreet girl accepting your telegram. She pays no attention to the meaning, whether it is sad, or joyous, or embarrassing. But she must be prepared to deal with all that comes to her desk.´.

In its travels from one point to another, the information is liable to encounter some losses due to media disturbances, which might be labelled in general terms as ´interference´ whether natural or man made. This loss of information, some of which is vital to the operations of the system, may be either intentional or unintentional, and can be classified accordingly. For the purpose of this work we are interested mainly in the intentional interference that would account for ´corruption´ in human institutions. Noise is an example of an unintentional interference since its effect has a legitimate origin. Noise, according to Ashby, in one system will be due to some other macroscopic system from which that system cannot be isolated. This fact accounts for the meaningless messages systems might receive and one of the basic factors in deciding the relevance of the information to the system. The intentional interference, on the other hand, is blocking legitimate, and relevant, information from reaching its destination with the resulted effect of:

(i)the receiver receives nothing, or

(ii)the receiver receives a falsified message.

In practice this would happen when someone in the system reshapes the information going through him to the top and strips from it (which is practically stopping the real information from passing) any indication that reality is different from the predictions of the top. It also happens when someone changes that information as it benefits him personally. Such interference is due, this time, to a microscopic system, which is manmade to serve his interests. It can be stopped if the microscopic system is identified so that its coupling with the system is severed. These microscopic systems can be easily created inside the system as long as there exists a gap (i.e., missing information channels) in the coupling of the system (which is bad enough in itself).

2.4. Variety

Variety, in general terms, is the number of distinct elements in a set (Ashby, 1956). Which is in turn a representative measure of the degree of complexity in a situation (be it a message, an event, an object, etc.) taken as a set of elements, and referring in particular to the number of possible states (be it meanings, outcomes, shapes, etc.) of whatever it is whose complexity is to be measured (Beer, 1979). Its straightforward measure, therefore, is determined by the number of distinct elements making the whole whose variety is to be measured and the number of values each element can assume during its lifetime. So, the straightforward measure of variety in an 8 bit computer byte is 256 (i.e. 2^8). This variety is equivalent in fact to the variety given by 8 Binary Digits, which is the second context in which variety can be equated to logarithmically, and determined by taking the logarithm to the base two of the variety of the set (log256=8 bits). But if this same byte is to work in a different manner, so that whenever any of its positions (bits) is equal to one, the others are zero, then its variety in the first context is now 8, which can be given by only 3 Binary Digits (i.e. 3 bits) which is equal to log8, as measured in the second context.

There are cases in which the variety of the subject will be different to two persons without any change to its working order or structure. Ashby's example of the two-armed semaphore is a good one, where it can place each arm, independently of the other, in any of eight positions to give a set of signals whose variety is 64 (i.e. 8^2). But for a person who cannot recognize two of the distinct positions assumed by each one of the arms, then obviously less signals will be recognized by him, thereby reducing the set's variety to 36 (i.e. 6^2). Which means that the second person is not sensing 28 semaphore positions, which he is liable to misinterpret when any of them is given by the semaphore. In other words, while the semaphore variety is 64, the person's variety regarding the semaphore is only 36.

Ashby's conclusion on this case is: 'the set's variety is not an intrinsic property of the set' (Ashby, 1956).

Another important observation about variety comes from a case of two transmitters of information reporting to the same person. Each one of these transmitters has a different set of possibles to choose from (because of the different constraints put on each of them). Ashby shows that a message from the transmitter with the larger set of possiblities (variety) gives a more satisfying message, than that of the lesser variety transmitter, a case which gave rise to his second conclusion: 'the information carried by a particular message depends on the set it comes from' (Ashby, 1956). What was obvious from Ashby's example regarding such a case is that the message which came from the set of four possible states had managed, as a tool, to destroy three of the possible states in the set leaving only one. While the second message destroyed one and left one, yet both messages managed to deal with the whole set that was associated with each of them but had different amounts of information, as we will see latter.

It is apparent from the first conclusion that the observer is defining the set, depending on his ability to discriminate between, or evaluate each of, the elements of the original set. Another qualification, which may be added to this; is that the elements of the set must ´have different meanings to him´; when it comes to processing, those elements with the same meaning will no longer be ´distinct´ except as one single element (state). If we consider a pedestrian crossing with only one set of traffic lights working, let it be the one meant for the cars, we can see that the pedestrians can use the information coming from that set for their own safety as well, from knowing when the cars are allowed to pass and when to stop. Yet, we can see that the pedestrians treat the ´amber´ light which comes after ´red´ as ´green´ although the drivers treat them as two separate messages. It might be said here that the ´amber with red´ still has a different meaning to the pedestrian, but the fact is that it is only a ´possibility´, which is not yet processed, that the cars would stop which forces him to regard it as ´green´ and not cross for his own safety (self-interest). Thus this set of 'four' distinct elements for the car drivers becomes a set of 'three' distinct elements for the pedestrians due to their own definitions, as we can see from the illusration:

	Red	Amber with Red	Green	Amber after Green
Driver	stop	stop if you can	go	get ready to go
Pedestrian	go	stop	stop	stop if not started

STATE against OBSERVER illusration.

This can be said to be, that the person is associating his own self-defined set to the situation. The variety measure is then defined from that set. So, although any situation must have its own inherent set, people interested in it would associate a set of their own to it, depending on their abilities. This set might be exactly the same (which is something rare in real life situations) or an approximation of the inherent set. This also explains why different people draw different conclusions from the same situation because, as we have seen from the second conclusion, they would associate different sets which would mean different varieties, i.e., each of them would recognize the situation as having a variety that is different from the others, that would result in different information capacity from which to draw the conclusions. The most successfull conclusion would be drawn by the person who has associated the nearest set to the inherent set.

This is true in real life situations since the vast majority of ´things´ observed by the human have their inherent sets known only to their ´creators´, thereby leaving the human to associate his own set and, as a result, a variety measure that can or cannot be enough to deal with that situation. This means in turn that: ´The situation´s variety is not an intrinsic property of the situation´.

It will be seen later that this is the variety which is required to be dealt with by the Law of Requisite Variety, in our artificial application to it since we, as observers, are limited by the facts that: (1) we can only comprehend this measure of the variety (and not the inherent one), and (2) we some-times don´t have the capacity to deal with some states even though we know it is there. It is possible therefore, that one or more of the unrecognized variety would go into action, a situation that would definitely result in an uncontrollable change, for the simple reason that this state was not expected and was therefore unaccounted for, and in addition it is totally misunderstood.

Ashby (1956) emphasised in his work that ´we must stop thinking, as we do as individuals, about "this message". We must become scientists, detach ourselves, and think about "people receiving messages".´ and gave an example of how two transmitters of messages with different constraints, when giving the same message, would result in different informations being received by the receiver. If we expanded this example and examined it from the point of view of the people involved then we can see, from the different interests the people have in the subject of the message, there emerge different sets of possible states that each receiver would associate (from his end) with that message a priori. As a result of this, the same message will give a different measure of information to each receiver, which is very much dependent on the set associated with the message, since the message is acting as a selector from that particular set (i.e. the associated set). As the set´s size increases or the selection narrows the message would get more informative. So, referring to Ashby´s examples on variety, the message ´he is alive´ was transmitted by a source whose set is made up of two elements: ´dead, alive´, while the message ´he is well´ was transmitted by a source whose set is made of four elements: ´well, ill, seriously ill, dead´. If we reversed this example to become one of a single transmitter (and forget about the other two) with a high variety and capable of transmitting any element of the two previous sets, with two persons (instead of one) concerned about the subject of the message coming from this transmitter, and changed the context of the situation by having the two persons with different interests in him, then we can see that each one of them would define a prior set of possible states to receiving the message, depending on his own interest. If those two persons were his wife and his lawyer, a possible case would be that the wife´s set will be the second set mentioned above, while the lawyer´s is the first set. Evidently, the information power of the second message is more precise than the first one concerning the well-being of the subject of the message; and the main factor in deciding the weight of the information in each message was its magnitude of selection with respect to the size of the set.

2.5. Law of Requisite Variety

Variety, when considered in the control context, is a destabilizing factor, simply because it introduces new states to the situation under control, and nudges it as a result from its equilibrium state if it happens to be in one. To restablize that situation, it is required that the new variety is contained in a manner that will bring the situation back into an equilibrium state. The question here must be, therefore, how can this be done? Ashby´s answer to it is by variety also, but this time it must be produced by the controller.

In any homeostasis situation, the controlling part is responding to the controlled part´s actions, so that the homeostat continues preserving its nature. If we explain this dynamic interaction in terms of variety, the whole interaction then starts by the controlled part producing a variety (i.e. it changes its state in a bid to escape control) and the controlling part responding by a variety of its own to keep the homeostat within its equilibrium limits (i.e. within the current basin). Basically, in a situation like this, there will be observed three sets of variety: one for each of the interacting parts, and one for the outcome of the situation which is obviously the one that will determine the state of the homeostat.

Let us observe a detailed example of this dynamic interaction in which we have a system S, as the controlling part, that has four previously prepared plans of action ($\alpha,\beta,\delta,\gamma$) to be implemented when the need rises. The environment of this system E, as the controlled part, would generate five different conditions (1,2,3,4,5) during the time of the observation. The interaction between S and E would result in an outcome (up to 20 different characters) depending on the actions of each of them. The whole interaction is simulated in Table 1 where S is represented by the columns, E by the rows, and the outcome appears in the cell position of the grid.

		S			
		α	β	δ	γ
	1	a	b	c	d
	2	e	f	a	h
E	3	i	a	k	l
	4	m	n	o	a
	5	p	q	r	s

Table (1)

The facts of this situation, which will be due to constraints, are:

1.E always starts the action with S responding immediately.

2.Each plan put into action by S would result in different outcome for the same condition — i.e. no same outcome would appear more than once in the same row.

3.A plan put into action by S would result in a different outcome for each condition assumed by E — i.e. no same outcome would appear more than once in the same column.

4.As each condition assumed by E is different, it follows that the outcome of the situation will be forced by E (which thereby becomes a controller), unless S changes its plan (state) to force the outcome it desires.

Due to these facts, it can be easily seen that if the number of columns in the table is less than the number of rows, then a certain outcome, say ´a´, if always desired, would certainly have to be missed on the application of each of the extra conditions. So that, if the outcome ´a´ appears in the first four rows, it will definitly have to appear in a different column each time, therefore, there is no way for it to appear in row number 5 unless we have a fifth column. Such extra conditions (like condition number 5 in Table 1) are supplying us, therefore, with unwanted variety, and the obvious course of action is to increase the number of plans (variety) available to S.

P.S. another solution could be by eliminating condition ´5´ from the environment, a case which needs extra intervention on behalf of S, in a more complicated case which would also prove the same result shown by the example.

It could be concluded from any table drawn according to the natural conditions in managerial systems (i.e. the number of rows is higher than the number of columns), taken with variant numbers of rows and columns, that for each plan we add to S´s capacity, we would have one less extra condition out of control — i.e. one less extra variety out of control. In other words, only variety in S´s actions can force down the variety in the outcome. Only variety can destroy variety — this is Ashby´s law, the Law of Requisite Variety.

Any system, therefore, cannot continue to survive unless it has the minimum required variety to match (destroy) that which is coming from the environment. It should be noted here that, in managerial systems, not every unwanted variety is in fact fatal to the system, meaning that it will denature the system. This can be also explained in terms of sets and tables. If S has the inherent set (of states) ´a,b,c,......,s,.......,z´, and if the sub-set ´a,b,c,.....,s´ represents the different states which the system can assume while it is still operational, and if the sub-set ´t,u,v,.....z´ represents the remaining states which the system can assume but will no longer be operational, THEN the system is said to be surviving, as long as it is within the first sub-set (Ashby, 1956). The minimum control required (hence the minimum variety required) to keep the system operational is that which will prevent the system from going into the second sub-set. Therefore, S´s task can be summarized in two moves if the ideal state for the system is, say ´a´:

(i)always to force state ´a´;

(ii)if not, at least any state of the sub-set ´a,b,c,.......,s´.

The point is that with (ii) the worst that can happen to the system will simply be that it moves into another basin of stability, but if the system moves into any state of the sub-set ´t,u,v,...,z´, then there will be no system to control any longer.

		S			
		α	β	δ	γ
E	1	a	b	c	d
	2	e	a	f	g
	3	h	i	a	j
	4	k	l	m	a
	5	x	y	n	z

Table (2)

Table 2 could be a simulation of this system from which we can now deduce some important notes:

1.Time is essential for S to react.

2.It is essential for S to recognize E´s actions.

3.We would have a super control (i.e. state ´a´ always) if S reacts almost simultaneously with E and has equal variety to it.

4.In human institutions, S´s variety cannot be equal to that of E, and S´s reactions cannot be simultaneous with E´s actions. Therefore the outcome is bound to change from state ´a´ to some other state of the sub-set during the time gap between E´s change and S´s plan to take effect.

5.If E changes condition before S´s plan takes effect, then a long time might pass before the outcome is ´a´ again. This would happen when unacceptable time is spent in:

(a) S feeling (sensing) the change.

(b) S recognizing E´s condition and deciding on the plan.

(c) the implementation of the plan.

(d) changing to another plan when E´s conditions change again.

6.From Table 2, it seems advisable to apply plan δ on condition ´4´ to force the state ´m´ (rather than plan γ to force the state ´a´), unless we have a super control, otherwise the system would disintegrate if the E condition changed to ´5´. This explains why some systems avoid going to the ideal state when they could. Those managers who do go to plan γ on condition ´4´ are said to be either taking chances or shrewd. Those taking chances have based their action on the probability of 1 out of 5 that condition ´5´ will be next, while those who proved to be shrewd were certain that condition ´5´ will not be next. There is also a bonus in going for plan γ, if it succeeds; for while everyone has gone for the less ambitious plan, they have left more for the ambitious manager to reap.

In reality, such systems are not found in isolation, and it must be realised that the environment contains many such systems functioning at the same time. The important fact here is that the outcome of each interaction by every one of these systems is affecting the environment in one way or another— a fact which, when simulated by such a table, would show a chain that would link the output of the system to the environment, and therefore contributes to inducing a condition from the environment. A living example of this is the present oil market, in which the outcome of the individual plans of marketing and production implemented by each producing country has forced the present condition of the oil glut while going for the highest revenues possible, thereby flooding the market with oil and forcing an unfavourable environment condition on those countries. The oil consumers show a better understanding of the dynamics of this situation by trying to stabilize the environment in its present condition which is pushing the producers further to lower their prices and sell even more to keep the level of revenues at its best for them—a vital contribution to stabilizing the environment in its present condition. This is not a hopeless situation for the producers, for the same fact that led them into this condition can also take them out of it.

III. THE SITUATION OF DEVELOPING COUNTRIES

3.1. Characteristics of the Situation.

- 3.1.1. The technological front
- 3.1.2. The management front
- 3.1.3. The viability front

3.2. Cybernetics in Management.

3.3. Data Processing Requirements.

3.4. The Development of information systems

3.5. An Illustrative Application

3.1. Characteristics of the Situation

Many reasons may have contributed to the developing countries having remained not fully developed, some are self-created, the others are imposed on them from outside. The problems facing them are diverse, ranging from merely providing daily food to entering the technological arena. It may be morally arguable as to which extreme of this scale is the more complex task, because of its consequences, but the case which is taken in this work is that of those who can afford, and embark on, large development programmes.

These large development programmes will necessarily mean the involvement of handling technology that was developed through different stages. These stages are not going to be experienced by the developing countries whose ambitions are to acquire present day technology. An important aspect of these stages is that they were intermediary stages during which the industralized countries (i.e. the inventors of the technology we are referring to) gradually came to terms with the change they brought about, and through which they gained the insight that led to the further development of the technology. This has resulted in a wealth of experience in utilizing this technology and in managing the new organizations using it. Not having to go through such stages by the new users means a great advantage in speeding up the development process. But this advantage is being offset by disadvantages which cast their negative effects on the result of developing programmes. and delayed considerably their implementation. These are the result of the bigger change, considering the shorter time span to feel the full effect, which was experienced by the new users in their bigger jump ahead with respect to using technology. This involved a large gap of technical experience coupled with a large gap of missing experience in handling this technology and organizations using them. Some aspects of this experience are typically not necessary, especially those which affected the society negatively. But the purely technical aspects are essentially necessary if an efficient utilization for the new technology is to be achieved, and a kind of autonomy in this field is to be gained.

It seems that the real problems which faced these countries, and therefore delayed their progress until now or resulted in half measure programmes, were concentrated on three fronts. The first is to recognize the real new technology, the second is in management, and the third is the viability of the commissioned project from past programmes and institutions which are already existing.

3.1.1. The technological front

Present day technology has hugely amplified human ability to deal with environmental changes and the more complex internal organizations which in turn brought about more rapid changes. While heavy industry and physics had amplified the physical powers (primary powers) of man during the industrial revolution, present day technology added the power of dealing with information (secondary powers). All the advances in present day industry and science were made possible by the new information technology that even made its own technology propagation much faster than any other technology before it. Information processing meant speed, and speed meant the ability to deal with more complex situations. This fact was promptly recognized by the developed societies and helped them a great deal to cope with the rapid development, or as we should truly say, has helped them to develop rapidly. Meanwhile, the developing societies are yet to acquire the last century's technology, while those who are reasonably wealthy wanted to join the advanced world. There are two major reasons why those societies are still lagging behind in the field of information processing. The first is that they failed to see the importance, and the involvement, of information (because of its abstractness) in the development programmes which they planned. The second is, even when they did recognize that importance, there was the reluctance of the producing countries of such technology to provide these societies with their needs because of the potential military use on one hand, and on the other hand the technological edge this technology gave them. The limitation of the equipment that was supplied, and of the training provided for the developing countries' nationals, was obvious in the supplier companies' dealings and the exporting regulations of their governments.

Other reasons that contributed to this lagging behind, were the very high price of this technology. As it was a new technology at the time, this cost seemed disproportionate to the cost of the whole project which that technology is meant to improve. What enforced this 'cost-effective' mentality was the ignorance of the decision makers and some of those advising them, who happened to be trained on the orthodox methods, as to the real value of information. Those who passed this self-created barrier were faced with a more serious one. Because of the potential military use of this technology, and the polarization of world politics, to acquire some of the useful mainframes and communication facilities, it was essential that the buying country should comply with a foreign government's political conditions, which meant to some of them a loss of sovreignty. Even for those who managed to acquire such equipment after all, it was run by foreign nationals to safeguard the technological edge, which meant that the decision maker was still not able to see the real scope, or have first hand knowledge, of the information importance and its usage in other applications.

3.1.2. The management front

Development programmes have proved to be highly complex and not an easy task, hence managing such programmes in the developing countries has proved to be a real problem for the country, where the classical management traditions are not yet developed to cover all levels of the development programme. It may be arguable in the advanced societies´ situation, whether the orthodox management is still as effective as ever, since it is the result of a long term and gradual experience that accompanied the development of these societies from the beginning, but the reality in the developing societies is completely different, a fact which presented them with yet another dilemma.

In the developing countries´ situation, and where technology has evolved, assigning the management task to the outstanding technical professionals has become the norm. This fact has a double negative effect on the country. Firstly, it occupies a trained technical professional with managerial activities and gradually separates him from his real profession, where there is a need for him, and if we assume that such a person has been given the managerial post because he is one of the best in his profession, then his loss is greater. Secondly, basically that same person is not trained for management; therefore, his performance must be poor (until he become experienced, if at all).

Such practice cannot be avoided as long as that managerial post remains the highest prestigeous post in the society and is sought after by the professionals themselves to prove their value in society. The more important fact behind this norm may be to free the highly capable and dynamic professional from the rigid controls of the orthodox management professionals, which would certainly inhibit his potential, or at best, limit it to his local area of work.

The stability factor and the kind of disturbances facing the system constitute an area where managers are faced with the real challenge, and are therefore required to show their potential. This area is somewhat different in its contents from its equivalent in technologically advanced societies. Instability is magnified here since its outside components are greater. When the world development system is viewed, the developing countries are situated in the environment of this system, and are considered as essential markets for the developed system. This results in a stronger influence exerted by the developing systems' environments on their systems (which now contain the developed systems with all their sophistication) whose variety contents are becoming greater than that of the system because of that fact. (The elements of the developing country's environment are more sophisticated than the system itself.) This would mean more constraints exerted by the environment on the developing system that would decrease its flexibility, which is an essential ingredient for survival (as the only alternative open to the system is to increase its internal control).

The internal factors of stability for these systems are also dissimilar to those of the developed societies. The ambitions and the failures of the human elements of the system are exaggerated because of the involvement of the future of the country, hence their rewards and penalties are exaggerated too, so are the psychological components of all parties involved. Espionage, whether industrial or political, is a fact of life. Another fact of life is that people and governments with different interests (whether in the failure of these programmes or in selling their products, or in directing these programmes towards their own interests. etc.) are spending huge sums of money as gifts (or bribes). These facts of life, are an added dimension which is hugely increasing the uncertainty both in the planner and the individual, whether in their internal dealings or in those with the outside systems. So management, in the orthodox context, need to be exceptionally capable and experienced.

3.1.3. The viability front

Many of the developing countries have been through several development plans (multi-year plans) but with less success than anticipated. This has meant incomplete programmes with some of the required links missing (whether due to bad planning, unfinished or even abandoned projects). Thus, a substantial number of the commissioned projects are left inefficient, either because they are internally so, or because they are incompatible by themselves. Such half-measure development has caused the country to inherit heterogeneous and unproductive establishments which have been maintained by the governments to provide work or for their prestige. So although these systems are functioning (for they still exist), they cannot be regarded as viable, since they are dependent on a continous supply of funds from the government to keep functioning because they cannot efficiently maintain their own existence. This in fact represented a whole problem in itself for new governments determined to get the development right this time. At the same time, their encounter has highlighted the danger of having a half-measure development, and became an education in itself for those governments embarking on new development programmes. It might be argued here that this, in general terms, is not a strange situation to the developed societies, with the existence of the subsidies system. The resemblance is in fact only on the face of it. The problem behind it is much different. The British system, for example, can be made productive and efficient, in theory at least, since the viable components and the infrastructure are there, but these are both missing in the developing country situation.

Another reason behind the weakness of the finished development programmes (in operational respect), lays in the ideological approach to the planning, and the staunch determination of the governments to strictly implement their ideological beliefs (whether in central economy or free economy). Those who believed in the central economy had an added reason to consolidate their grip on the country's economy. They had to stabilize the country economically by taking away any economical powers from the private hands of individuals who were seen to put their own self-interest above that of the country. This measure did work, but not without its price, in stabilizing the situation, which by now had reached a level where these countries could afford to relax these measures while applying indirect control. Other governments, on the opposite side ideologically (i.e. free economy advocates), had embarked on developing the non-profitable projects considered as essential complements to the profitable ones in the anticipated infrastructure, thinking that the private sector would find the incentive to commit itself to the rest, a situation that resulted in the same end.

Some of these countries have learnt from this experience and managed to overcome some of the difficulties and make some adjustments in their inflexibility. On one side, those with the central economy have found that certain low-level economic activities are better left to the private sector, and hence make some provisions for them in the central plan. On the other side, an initial belief (promoted by the central economy advocates) is enforced, in that large development (especially industrial) programmes will never be realized by the private sector, although in some of these countries the capital required is available to that sector. The reason is obvious in the world development system where these countries as markets are provided for by the developed countries' large industrial corporations, and therefore the incentive for the private sector, not to invest in such industries, is two-fold:

i. they cannot compete with such industries, and

ii. these industries offer them the opportunity to trade safely in their products for a high and easy profit.

So, they preferred trade on all levels, developing a mentality which says: 'there exists a larger system which can provide everything we need, so why take the risk of the unknown?' This experience has left this part of the world with somewhat similar development systems whether the country is following a socialist or a capitalist ideology. All the large development programmes are becoming the responsibility of central government, while the lower-level activities are left to the private sector. Our tool for control, therefore, has now become an added need to create harmony between the central government and the private sector, which is independent (or at least highly autonomous). The tool needed is one that is not strictly central, nor loosely free, that can cope with a very complicated situation materially and psychologically. The principles of cybernetics can help solve the problem of viability in order to reach the organization of freedom which is necessary for effective management (see Ch.4).

3.2. Cybernetics in Management

The cybernetic principles can help in avoiding the repeat of the kind of failures faced by developing countries on the management front. These same principles can also offer a wayout of these failures which is also necessary if the country wants to utilize existing wealth. A close look at Management, in general terms, shows that it is all about Intervention and Control. Communication is what makes such management possible; and Information and its processing is what effects that communication. Orthodox management is built around experience. Its generally accepted principles are all the actions that would comply with the term ´common sense´, which are in reality all those actions whose consequences are regenerated every time they are applied, and are known to be workable and acceptable with respect both to ethics and pay-off, from the long years of practising them and observing their application. In other words, it is a recognized pattern of action and reaction (event and action associated with it) that has been developed throughout the years of practice and based on experience (whether first hand or otherwise). The successful initiatives taken by some managers, i.e., those termed as ´common sense´ actions, are explained away by orthodox management as ´instincts´. Which is another way of giving the credit to their performers who had the ability to recognize the general pattern of a complicated situation and identify the right principles governing it, which enabled them to choose the correct technique in their approach to tackling the problem.

The main recognized requirement for successful management in the orthodox context is ´experience´. This kind of experience is built up in a particular society and grew up with that society. The dilemma for the developing societies stems from the fact that the components and the active forces of their societies are a great deal different from those of the technically advanced societies. A component borrowed from other societies´ experiences may not be the right one for their society and therefore becomes obselete. To build their own experience would mean to start decades behind, which is a delaying factor in the developing process.

What the orthodox management had discovered through a long and painful path can now be adapted to the developing societies, without its shortcomings. The principles behind them can now be explained by cybernetics and the details can be worked out for each particular society. Cybernetic laws, principles, tools and explanations are suitable for the very complex projects, which are what the development programmes in developing countries are becoming. It is also possible to train managers in cybernetic principles and techniques and arm them as a result with an appreciation of Information and Communication in the management arena. This is by no means going to create experienced managers, but it is the only alternative that seems possible.

Cybernetics makes the management task more receptive to the well-defined procedures to perform many intervention and control acts that do not need an on-line human brain to work them out. This should free the available, and scarce, resources of manpower to deal more effectively with the higher evaluative and strategic acts.

It is not an issue here whether the orthodox techniques or those of cybernetics are the right tools, for obviously many of the orthodox techniques are showing an implementation of cybernetic principles even though they are not recognized clearly as such. The issue is: given that such long term and gradual experience is missing, how can such complex development programmes be managed? Given that there are no established traditions (that are delivering what we need) to lose, or asked to go against, and there are proven principles that we can use successfully, then the answer must be found in cybernetics.

3.3. Data Processing Requirements

Computers allow us to process information and to carry out problem-solving activities that cannot be achieved at present by any other means so efficiently. They might lack human intelligence but they definitely have an infallible memory that is not matched by the human. Their speed of action, both in calculation and recall, unlike the human, is not affected by the tension of the event (emotions in general). These valuable traits would compensate for the computers' lack of 'human intelligence'. A combination of man-computer, with the tasks assigned to each side according to their best qualities, would certainly generate a higher power in information processing, where each side is compensating for the other's short falls. The power of this man-machine combination goes beyond the simple tasks of information retrieval and straightforward calculation to the ability to perform logical inferences, as demonstrated in the current applications of computers in the AI (artificial intelligence) field (Rudall, 1981).

The use of computers has came a long way since its introduction to the world when it was mainly used for scientific applications. With the progress of its technology, computers entered the field of commercial applications of a managerial nature, and recently the industrial arena. Many organizations now rely on computers in the management field, especially after the progress made in microprocessor technology, but this is not without its drawbacks. If the user is not certain about the computer's true role, or how to use it efficiently in his application then the most probable result will be that the machine will sit idle in that organization producing an adverse feeling. There is no question about whether or not the computer is useful, because undoubtedly it is, and the gap that it can fill in the organization's operations is there. But what almost all users fail to recognize is that such a powerful addition to their organization demands a respect equivalent to its powers and its providence to the organization that would require reshaping the organization around it, and not reshaping the computer around the organization.

The industrial use of computers is also being advanced by cybernetics to play a central role in controlling processes (Rudall, 1981). The digital computer in such situations receives the incoming information about the process in hand, whether from another analogue computer or linked directly to the process or otherwise, and calculates its relevance and decides on its significance, then issues the proper instructions to control that process. The practical application of cybernetics, in both areas, industrial and managerial, involves choosing those processes that can, because of their very nature, run automatically, or those that are better run in a semi-automatic fashion. Recognizing such processes might be seen as only suitable for the industrial area, but it can easily be seen that there are numerous processes in management which can be treated as such. The term ´automatic´ means running a process without human intervention, while ´semi-automatic´ refers to what is also known as a ´man-machine interaction´ process. With the intelligence limitations of the computer, and when it comes to the management area, it is better to think of ´automation´ mainly in terms of man-computer interaction, where each knows when to intervene to ensure the smooth running of the processes (Stewart, 1982). The man-machine approach would involve a man-machine dialogue. Dialogue, in the sense of free and unconstrained statements is an area of research in the field of artificial intelligence, and is yet to be developed. But the ´question and answer´ dialogue is now attainable with the ordinary commercial software, where the information is stored in the computer´s ´databank´, but is restricted to the use of particular patterns of dialogue. This has become one of the major applications of computers in government and in industry and commerce.

The appearance of microprocessors has changed the approach to many aspects of automation. Processing powers can now be distributed and the central processing limitations can now be overcome. Interfacing became friendlier with the microprocessors (they are even called ´personal computers´ in the commercial field) where the software writers are forced to write friendly software and depart from the old assumption that the user is a computer expert.

The power of the micro has come a long way now in terms of the diversity of its applications and the freedom of using the processing power in various applications. This trend is still going on, and one only has to look in the commercial publications to know what is new, for breakthroughs are happening in weeks rather than in years. A microcomputer can be used by a child to play with or to learn with; a single manager to run his small shop or his company; a salesman to use it on the street, at home, or at his work place; he can link his computer where ever he is, to a larger mainframe and talk to it, by telephone or even via satellite; a user can now send telexes via his microcomputer, together with, letters, memos, and reports to anyone, anywhere, anytime, all of which can be done very easily, very cheaply, and in minutes; they can be operated as a group in networking mode, where each computer can talk to the others. It seems that the limitations for their use and application are only in the human brain—whatever setup the human brain can think of, can be achieved, provided there is enough motivation in it.

The main obstacles in obtaining some useful products in this field have been eliminated today as some of the required mainframes are becoming available with no strings attached to them (see Subsection 3.1.1), while the introduction of microcomputers has made networking and spread computation much easier to attain. The major influence in fact has been micro technology. With it, prices became affordable and much more feasable, even to the old mentality, where a single microcomputer can now be obtained with memory and peripherals that are sometimes larger than those of a mainframe of the late 60´s or early 70´s. A typical mainframe of the late 60´s had a core memory of 16K expandable to 32K with two disk units each of 0.5MB storage capacity expandable to more units, plus a line printer capable of 300 lines/minute , and two tape units. Its price was in the range of 0.4 m $, and the overhead cost of preparing the site and the airconditioning plus the input units (card punch or magnetic tape) would bring the total cost of obtaining such a computer to 1.0 m$, an enormous amount then. A typical micro of today can provide the same facilities as the above mentioned mainframe with a negligible fraction of its price. Furthermore, there will be no overhead costs incurred as a result of obtaining the micro, because almost all the software necessary, which is available on the mainframe, is also available for the micro (N.B. the only commercial application that was thought to be suitable only for the mainframe software, i.e. the database, is now available on the micro, and its file format is compatible with that of the mainframe, e.g. dBASE II and dBASE III). As a new technology, the micros have their drawbacks when matched against the mainframe integrity, security, and speed, but their advantages make them more feasible for the developing countries. Cost is one of the advantages, whether of the hardware, software, or their maintenance. The software might be the chief benefit for the developing countries when seen through its real value. It is the experience of experts, organized into well defined procedures for use by other people. They benefit their users and teach them at the same time in the area of their application for a very competitive price. An application package is the result of a whole experience in the field concerned which has been based on the principles of control and communication governing that field. The built-in commands of a package are simply the standard interventions most likely to be made in the application by the user and they are the result of the whole accumulated experience in that field. In addition, almost all the new application software is making provisions for the inventive or highly experienced user, by adding an English-like, very simple-to-use and learn, programming language as a command language that can make full use of the powers of the package whenever the standard commands fail, or for custom-made procedures. (Two good examples here are the dBASEII which is used in this work, and SPELLBINDER word processing and office management system. SPELLBINDER has a very effective assembler-like command language for the sophisticated user. It helps among other things, to standarize the jobs in the office and eliminate those routine repetitions, thereby freeing the user from such irritating tasks). This in itself is an added power to the users of the developing countries where the scarce experts can now spread their knowledge beyond the boundaries of their local work (the software developed in the present work is one example). It also means the use of the up-to-date techniques in the application field concerned. So the benefits of these packages can really be summed up in one sentence: they are ´experience´ delivered in kit form.

Finally, with the latest developments in micro technology, it is now possible to link the micro to a mainframe and, with the availability of micro software which is compatible with that of the mainframe´s, there should be no reason for assuming any limitation for the growth of the processing powers of an organization that is depending mainly on the micros.

Solving the problem of hardware and software is only half the task of solving the problem of technology. The rest is to use them successfully, and to develop the correct applications, such as the Information System in our case. The next section will examine this area.

3.4. The Development of Information Systems

In general terms, designing an information system for a developing country is no different from elsewhere. There are, however, very few specific requirements to be taken into consideration, as indeed in any other case, that are demanded by the situation. But, as a pure situation, one might take the opportunity here to employ cybernetics principles from the start, since the whole approach followed here would makes it possible while using the commercial packages.

´Information Systems´ are those systems which process information (secondary power) and are able to extract from it knowledge that can bring the desired change, if used properly.

Distortion of information can be prevented in such systems by making it information-tight (closed to information), i.e. the system should be selective in its choice of what information is to be allowed into it.

Certain channels , however, could be designed open to information, which would connect a special unit (separated from the operational units, but linked to the metasystem) to the environment, with no accounting procedure imposed on it to analyse information and report directly to the metasystem about the future possibilities. Some of the general areas in dealing with information are: the size, the monitoring, the learning, and databanks.

The presentation of the information in large quantities and diversified forms can cause destruction and lose its impact. There are three major operations that can help reduce the size of information entering the system:

1) Categorization, where the huge amount of information is separated into groups according to its nature. To begin with, any information can be classified into two classes with respect to the system:

Class One .... Relevant Information

Class Two .... Irrelevant Information

As we have mentioned before, Class Two can be channelled through to a special unit without affecting the system; while Class One can be categorized further according to the nature of the information. Basically, it can be arranged into five different categories, thus:

a) Decisions: where the information used to reach them is no longer needed by the upper level. Only the decisions themselves are now good information. These decisions are of three types, namely: operational, tactical, and strategic.

b) Performance levels of the various operations.

c) Fixed: such as machine specification.

d) Environmental: originating outside the system.

e) Educational: which would be gained from the previous four.

2) Condensation (especially for categories ´b´ and ´c´): because of the recursive nature of systems (see Ch.4), the upper layer of one system looks on a sub-system as one unit and treats it as such, although that sub-system may be composed of further sub-units. Therefore, while the management of the sub-system looks at the detailed information of its units before intervening, the upper management would be interested in the overall performance of the sub-system, thereby requiring compacted information about the sub-system. Such condensation may also be either according to a time factor or an organizational dimension (i.e. either time summarization as daily, weekly, monthly, etc., or operational group condensation as plant, project, sector, etc.), or both. In any case, the requirement is decided by the type of decision the management is involved in (operational, tactical, strategic).

3) Filtration; to separate the two classes of information. The filter may take many forms with business applications (ranging from an input format to a computer program) but the basic principle is common to them all, and it is simply that only the information required is transported by them to the system, no less and no more. Any other information reaching the system informally (and it should not be neglected, for it might

be relevant) would be channelled through to the special unit mentioned before. Two examples could be explained briefly here: one is the paper format on which data is collected from the field, whose design only allows the selection of items regarded by the user as relevant. The second is a computer program which would select for its application only those items needed from the larger amount available to the system (an example is the database program in Ch.6).

There are certain operations required to be carried out by the system that are essential to the total performance. It is essential, therefore, that such processes be monitored and routines must be set to make sure that delays in these operations are discovered. One example is the dispatch of the daily report from the project site at the end of the working day. A recovery procedure should be designed here, in case the report never arrives, which could be substituted by temporary assumptions or estimates to be entered instead of that report, or by doing nothing except indicate the fact that it has not arrived.

Learning from the processed information is a very sensitive and essential task. In many cases, it would be required that some facts, which only experience can supply, should be incorporated in the used database right from the beginning, to make it workable. While reporting (about performances for example), some explanations should appear as comments alongside the notes referring to performances that are not behaving as expected. This can only be done if a set of cases, which have already been encountered and are known to us, are included in a separate table with codes assigned to each entry. These codes can then be used as appropriate and their explanations can then be retrieved from the table later for the benefit of the user. As our main assumption here is the lack of experience, special attention must be paid to this area. An initial table can be constructed, although not complete, but may be designed as an open-ended table, with a special dedicated program that would accept explanations of new cases encountered by the various working units. The necessary steps of verifying, wording, assigning a code to it, and adding it to the table, are then done by an interactive exchange between a specialized human and the System.

Another area involved here is that of the performance standards that are used to calculate the general trend of the global implementation of the system. These include time averages , capability figures, and some indices generally used either to plan the operations or to calculate the deviation of the actual performance from the planned one. In a way, this is an easier task than the one mentioned previously, in the context of developing vs. developed situations, in that the System can be programmed to update these indices and standards every time it senses a change in them.

To be able to store all the collected information by the system (i.e. building a databank), and at the same time organize the storage in a way that would allow easy retrieval, cross reference, and easy updating to be performed on any item stored, is an essential requirement for the good control and successful intervention envisaged for the system. This would also make possible the two previously mentioned tasks of monitoring and learning, in addition to helping the daily operations. Past experience of handling applications data files has shown shared principles, regardless of the application´s nature, in storing, cross referencing, retrieving, and updating techniques. This experience was then put into a general program, called a database, to help all computer users to organize their storage and simplify their information processing on a sound basis. The cost of this technique increases rapidly with its degree of sophistication, and would be incurred in the areas of running time, maintenance, and hardware requirements. To reduce this cost, though would not necessarily affect the application´s sophistication. One approach can be a compromise between the free storage approach (i.e. the common and current one), and

Beer´s approach of using indices (Beer, 1975, 1979) There are many occasions on which the past information has no more value than the present information derived from it; as we have mentioned before, only decisions taken after processing some information are of real value later, since a decision vs. its consequence is what really matters later.

3.5. An Illustrative Application

We have chosen the Central Plan Follow-up System to demonstrate how our tool can be employed by the system concerned (N.B. this unit does exist in one of the countries concerned but this is by no means a suggestion that it should be followed by other users). The task of the System is to gather information about all the projects under the umbrella of the central plan. The information is then analysed and a measure of completion, for the projects as individuals and collectively, is inferred. The criterion used primarily in this task is monetary, and the reason for this is in the terms of reference of this System as the informing party to the decision body who releases the money consignments of the project´s budget. The collective measure of completion is also supplied to the Central Planning Committee at its regular meetings. The information gathering is done manually and the information is entered on paper forms, either by the project management or by employees of this System who would visit the site of the project, or by both methods.

The plan implementation system as a whole, in this example, is superimposed on the government system, where the implementation of the plan is delegated to the sub-systems (Ministries, Organizations, and private sector), and where the future link and operating will be. The follow-up system is also divided into sub-systems of specialised applications which are in turn divided into further sub-systems according to further specialization. The main task of the CPF System can be summarized in two points:

1)Measure and control the progress.

2)Analyze and select the information for useful storage (building the databank).

There are three major problems experienced with this approach:

1)The representation of the gathered information, as to the real and actual level of completion reached by the projects, by the time decisions are to be made regarding them, since the manual technique usually takes a long time to produce results.

2)Some valuable information is practically lost by the system for two reasons:

a)Only data primarily serving the monetary criterion is sought after (to speed up and simplify the process) thereby losing what might be valuable indications and useful elements for future work.

b)The retrieval of required past information goes through a long process so that by the time it is retrieved there may be no need for it.

3)The organizational structure, which might be the most important issue, and which has to be in its present shape, allows human conflict in every step of it because it imposes on the people who are implementing the project.

The solution must, therefore, be concentrated on reaching faster information-gathering techniques as close as possible to a real-time/on-line situation, therefore, eliminating the time lags affecting the efficient decision making and assisting in a successful intervention on all levels of management. At the same time, the tools used should be capable of tireless gathering, whether with respect to volume or time, with adequate support for processing the gathered information, and the capability for storing and retrieving the information with matched efficiency. Finally, the solution must pay adequate attention to removing the human conflict as much as possible from the system.

IV. CYBERNETICS OF ORGANIZATIONS

4.1. The System Identified

Identifying the 'whole system', has become a controversial issue with writers taking the two extreme lines of: (1) putting a very precise definition (to help this identification), in an attempt to identify the general system properties, which sometimes amounts to an article by itself (e.g. Bertalanffy), (2) suggesting that we should give up any ambition to know the 'whole system' (e.g. Ashby). Nevertheless, each side's work was criticised by others. Bertalanffy's General System Theory was criticised by many as insufficiently complete to reach the ambition it was set to reach, despite the effort he put into it. This was later (1962) accepted by Bertalanffy himself. Nor has Ashby's definition for the system as a 'machine with input' escaped the criticism that it is inadequate when applied to the living organisms context (Bertalanffy, 1962).

Cybernetics may have put its finger on a very important matter of fact which is preventing the generally acceptable definition from ever being reached. Ashby wrote in his 'Introduction to Cybernetics' (1956), that the 'aim must be to achieve a partial knowledge that, though partial over the whole, is none the less complete within itself and is sufficient for his (the researcher) ultimate practical purpose', and went on to give a general statement about the point of view taken in that work regarding this matter, that 'science (as represented by the observer's discoveries) is not immediately concerned with discovering what the system "really" is, but with co-ordinating the various observer's discoveries, each of which is only a portion, or an aspect, of the whole truth.' The Russian biologist Kremyanskiy (1958) also argued that, granted the importance of system properties, it does not follow that maximum pay-off comes from seeking to identify the most general system properties.

The caution shown by cybernetics over this issue, and for that matter what raises the controversy, lies within its appreciation of the reality of the observer's role and its influence on the outcome of our dealings with the 'system' in hand. Ashby argued that '"The system" may refer to the whole system quite appart from any observer to study it—the thing as it is in itself; or it may refer to the set of variables (or states) with which some given observer is concerned', and went on to emphasise that although the second meaning is more important for the practical worker, it will remain ambiguous if the 'particular observer is not identified'.

While this is so, there is no disagreement on the systemic principles governing the general behaviour and nature of 'systems', and these represent a common ground for the various practicing disciplines in this area. For example, they agree on the holistic behaviour and the interaction between the parts of the system and their relatedness. It is only when we come to the specific practice of a certain discipline that the specificities of things start to demand what we may describe as a 'tailor fitted' definition for the system involved and its boundaries. Such definition, and hence the identification of the system, cannot escape the influence of the practitioner's view of the world, which is the original controversy between the different disciplines. One writer summed up the dilemma of the engineers who had been asked to transplant their technical experience in systems, and the tools they used, to social matters, in what became to be known as the 'California experiment' (when system analysis was the vehicle designated at the time by Governor Brown to convey space-age technology to the public arena) :

> ''The engineers soon discovered the seriousness of the first booby trap mentioned earlier, i.e., the vagueness of the word 'system'. For them a system was a thing; its working was governed by certain laws; its actions were controllable and predictable. In the realm of social affairs, the system is an idea; it is a construct. In the words of a political scientist, 'all systems are human artifacts designed to facilitate choice... They are overwhelmingly dependent on the hierarchy of values in the mind of the system's architect... To adopt such systems as 'real' implies a belief in the 'naturalness' of an imperfect heuristic device'(Haas and Ruggle,1975). Only the least important of a system's workings are subject to immutable laws; if its actions are controllable in the short run, long-run behaviour is all the more likely to defy precise calculation. Face to face in a real life situation, the engineers did not know what to do and the public agencies did not know what to do with them. For the engineer accustomed to explicit tasks performed in a highly structured situation, the amorphous situation was unsettling. For the public officials, The specification of managerial objectives was tantamount to the demand for a quantified 'apologia pro sua vita', and in many cases he was

not able to provide it. Many of the positions created under PEP (Public Employment Programme) petered out when the funds were exhausted.´´

IDA R. HOOS, 1976.

For the purpose of this work, we will use a definition which is summed up from Beer´s writings, as follows: A ´system´ consists of a group of elements dynamically related in time according to some coherent pattern, and has a purpose attributed to it by the observer. We may add here that this purpose is shared and accepted by all the elements of the system.

It follows then that the environment of such a system ´includes objects and changes which exert considerable influence on the material system without being part of it... The environment furthermore includes all objects and phenomena which feel the strong and direct (or not too remotely mediated) effects of the system´ (V.I.Kremyanskiy, 1958).

4.2. The Viable System

The ´viable´ system is in reality just a reference name for an existing system which seems to be capable of maintaining its existence by itself (i.e. still possessing the properties which qualified it to be a system). The word ´viable´ would move us from the abstract constructs of systems to the reality of the construction, analysis, gubernation, etc., of systems.

Viability, as referred to by Beer (1979), whose model of the viable system has influenced this work, is the ability to maintain a separate existence, which can be taken to mean the ability of self-government by the system (N.B. ´Separate´ is used here to mean ´autonomous´ rather than ´isolated´). Any system´s existence is continually threatened by its own environment´s influences on it and needs to respond to them. Any failure to respond would endanger the system´s existence. These responses can be built into the system by design, but there is no guarantee that the list of these responses would be complete to cover every stimulus coming to the system. Therefore, for a system to be able to produce responses that were not included in its design, and for that matter to survive, it should have a huge variety of its own. This fact would necessarily make such a system a complex system or, in other words, a highly sophisticated system which can maintain its existence from within. It must be appreciated here that there is a difference in the sense of usage between the commonly used term ´complex system´, and our usage of it, for the former usually stands for the difficulty of understanding the system, while our usage does not necessarily mean the same. A system can be highly complex, i.e., possess a huge variety, and at the same time its working can be obvious and clear to the observer.

The other properties of the viable system are its ability to learn and to adapt to new environments. The viable system is necessarily an efficient system, which cannot only produce new responses to the environment, but also learns from this experience as to which is the optimal response. This might demand a change of state, and hence reorganization, to adapt to the new environment, which cannot be done from within unless the system has the ability of self-organization.

Finally, what might be the most important property of the viable system——for everyone apart from a casual observer—— is the holistic property of such a system; it cannot function efficiently except as a whole. Each part of such a system is contributing to its functioning through interaction with the other parts.

According to Beer (1966), there are three facts that must be respected in handling the viable system. Firstly, dealing with such a system through ´concepts, models, and controls that are deliberately of low complexity (variety) is to rob these systems of their viability'. This is to say that for a successful dealing we also have to observe the Law of Requisite Variety in this dealing. Secondly, for such a system to adapt and produce the required response, it cannot rely only on the information built into it by the designer. The communication theory (Ch. 2) asserts that enough channel capacity must be provided in the feedback loops of any system under control to match the capacity of the system to make erroneous responses. Hence, Beer advises that 'to isolate the system artificially from its environment, as is often done in industrial control situations as a convention for ease of management, is also to rob the organism or system of its viability'. Special attention must be paid here to the positive feedback loops of the system as they are the basis of learning and adapting systems (i.e. self-organization). Thirdly, 'if one starts cutting pieces out of the viable system...., they cease to function—or at least begin to behave atypically. The organism itself is likely to die'.

The viable system will have a purpose but it is not agreed upon by all its observers. So, until all the observers can agree on that purpose, we will assume it is that of the controller (steersman). When the controller's purpose comes closer to that of the inherent purpose of the system (i.e. the real purpose), the efficiency of the system will come closer to its maximum.

All systems are recursive in nature; that is, all systems contain subsystems and in turn are contained in a larger system (with the exception of the highest- and lowest-ever recursions). The viability of the system is passed on to its lower recursions, and similarly, if a viable system is viewed, it has to be contained in a viable system. These are the findings of Beer's Recursive System Theorem:

> 'In a recursive organizational structure, any viable system contains, and is contained in, a viable system.'
>
> Beer, 1979.

The viable system has its own environment in the universe which is coupled to it in a homeostat. The subsystems of such a system have their own relative environments which are a property of the upper recursion's environment.

All these properties, aspects, and ways of handling the viable system must be incorporated in any model used by the researchers of such systems. A model of the viable system is discussed, and suggested, in the next section.

4.3. A Model for the Viable System

Beer's model, as a cybernetic model for the system, is centred around the criteria of viability (The first reference by Beer to this model was in 1969, then in 1972 a full account of the model was published, followed by an up-to-date impression in 1979; in 1974 and 1975, Beer published related principles to the cybernetics of the model.) He asserts that the viability of the system has to be maintained on two levels of contexts; INSIDEandNOW and OUTSIDEandTHEN. These two dimensions of the mixture of location and time are in fact the philosophical basis of the model, which can be summarised as maintaining the internal stability of the system with an eye on the environment, whose changes are going to affect the system itself in the future. The main functions performed inside such a system, therefore, must be expected to lend themselves to one of these dimensions or the other, with some of them primarily concerned with the on-line internal intervention, while the others are concerned with the outside of the system and its future. One of these functions should be expected to play a dual role as a link between the two streams.

As the system's various functions are all concerned with, and involved in, the well being of their system, they are all taking into account the INSIDEandNOW-OUTSIDEandTHEN aspects of the system, because of the circular causal nature of these aspects.

4.3.1. The viable system´s functions

The model is a construction of five separate functions which are logically seen as the ones maintaining the viability of the system. We shall adopt the convention used by Beer (1979) in which each function is called a system with a number attached to it for identification. These five systems, namely, System One, System Two, System Three, System Four, and System Five, are linked together via communications channels, which is a reference to the importance of the information role in the model. As far as these links are concerned, they are shown in the model as entering and leaving the subsystems without referring to the specific roles of individuals inside the subsystem regarding that link (this will be discussed later in more detail). The first three systems, i.e. Systems One-Two-Three are the functions involved in the INSIDEandNOW dimension, while Systems Three-Four-Five are the functions involved in the OUTSIDEandTHEN dimension. The five functions are identified individually as follows:

System One:

> The operations function of the system which is responsible for the actual implementation of the system´s corporate policy. They are the immediately recognized subsystems of the system.

System Two:

> The service function of the system which is

responsible for the liaison between Systems One to coordinate their actions according to their internal needs, on one hand, and links this coordination to the central control responsible for the corporate policy, on the other. This function is an anti-oscillation facility.

System Three:

This function is the central control of the system which is responsible for the on-line intervention on behalf of the corporate policy. It is situated on the pivot of the two dimensions with direct channels to the daily-inside operations of the system (Systems One) from one side, and the metasystem (Section 4.3.4) from the other.

System Four:

It is the intelligence function which analyses the current environmental changes and their trends in the future, and assesses their effects on the system in view of the current policy. There is a tentative link between this function and System Three (the controller) in order to reach the best action (state) for the operational elements in the face of change. This Systems Three-Four relation is overseen by the next function.

System Five:

This is the policy-making function for the system.

It is in close contact with the circular interaction between Systems Three and Four. This is vital especially when the outside changes prove to be dificult to combat by the current policy, a situation that demands a review of that policy with a commitment to change. This makes the system flexible to change.

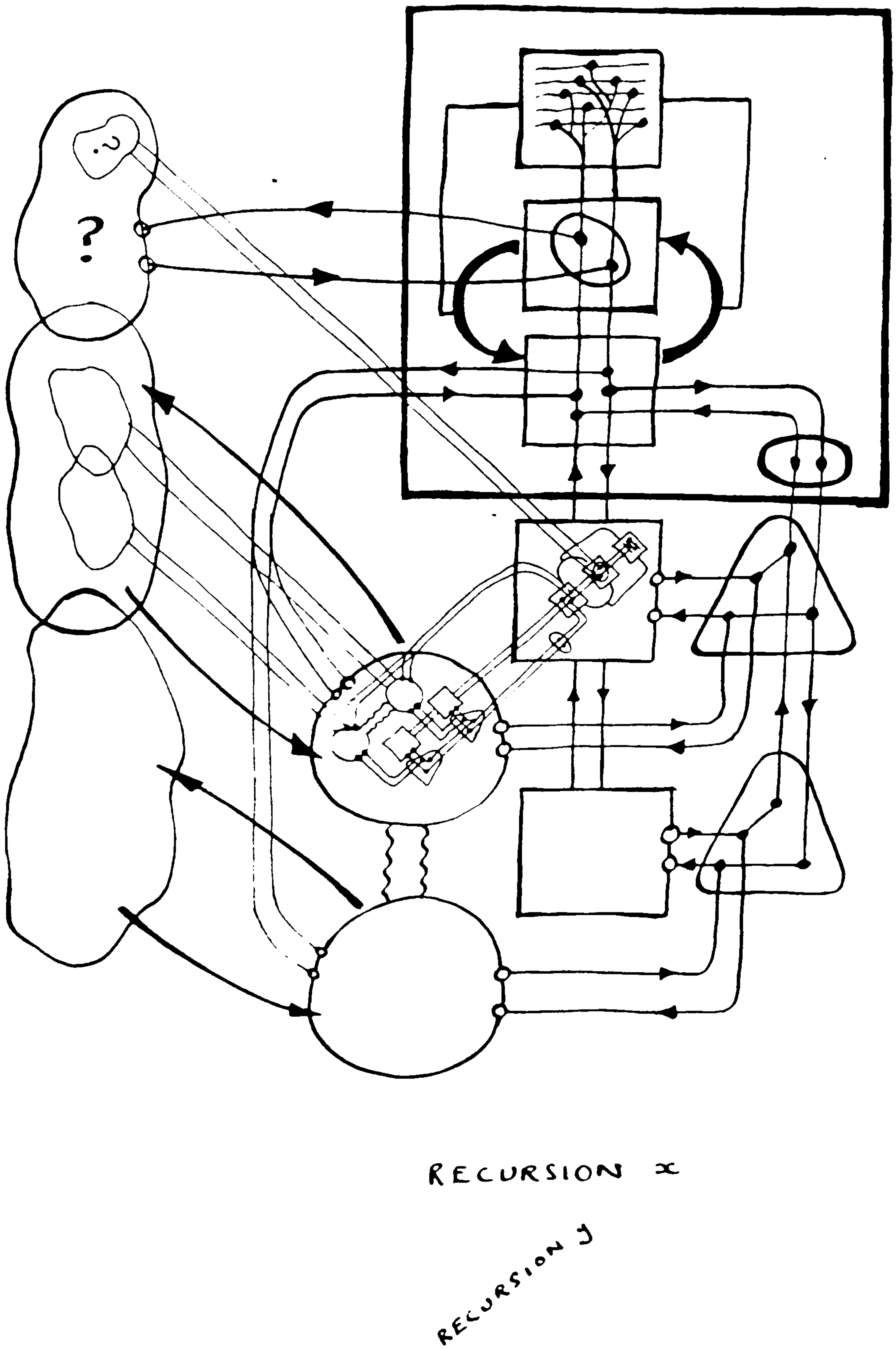
The five functions reviewed separately above are in fact complimentary to each other in a way that no one function can claim to be the decisive function in the viability of the system. In addition, if any one is missing, or even has a low capacity, this will undermine that viability. This is clear, and obvious, in the case of System One as it is the operational element of the system. The absence of System Two will leave the system subject to violent oscillations which would reduce its efficiency drastically, if not destroy it. Its alternative would be a direct control exerted from System Three on the operational elements which also has no less a price to be paid in efficiency. The need for System Three is as obvious (traditionally) as System One, but perhaps the most overlooked one of all these functions is System Four. Although it does exist in every viable system, people do not seem to recognize its presence, and hence its importance: because, in its lowest existent form, it manifests itself as a concurrent activity performed by the people at the top (Systems Three and Five) as they perform their traditional roles. As for System Five, the whole system would be going nowhere, if it did not have a policy (i.e. a purpose to fulfill).

These functions do exist in each recursion of the system with their links and interactions are depicted recursively for each layer of the system. Figure 4.1 shows a way of mapping (according to the model) two recursions of a system (recursion x and recursion y) which are linked together by communications channels.

4.3.2. The model´s distinct regions

An overall look at the model shows that the system is composed of two distinct regions (this is the result of the main philosophical approach of the two dimensions in developing the model). These regional divisions can be explained, in traditional language, as the management, or directorate, region of the system (Systems Five-Four-Three), and the implementation region (Systems Three-Two-One).

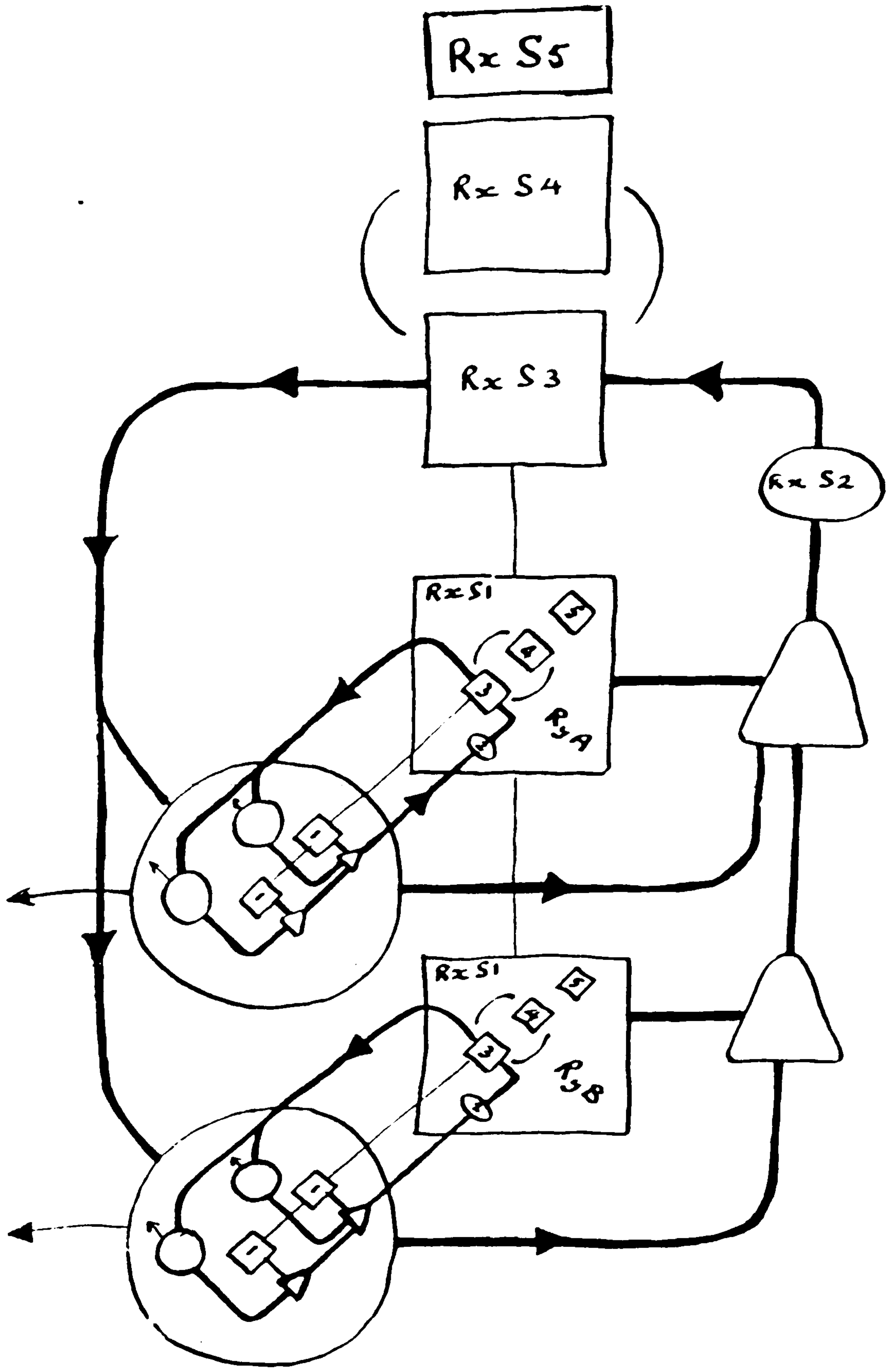
Systems Five-Four-Three make up the gubernation facility of the system and is referred to as the ´metasystem´ of the recursion in view. It is the region of the system that would shoulder the responsibility of the system´s organizational autonomy. The key to the operational



Definitive model of two recursions of the viable system (the diagonal of the page being used as the second dimension)

Figure 4.1, source: ´The Heart of Enterprise´, Beer

freedom inside the system is in the hands of the metasystem, as it lays down the regulations and directives associated with the policy it makes, and the way System Three will go about performing its task. Through the metasystem, the link of the system is secured with the upper recursion if there is one. This link can take various shapes, ranging from direct to indirect, or material to moral (ideological), depending on the nature of the system, without undermining the autonomical aspects of the system itself. In any case, the metasystem, if it happens to belong to a lower recursion, would definitely need to be in contact with the other systems implementing the same policy (i.e. other subsystems) and the upper recursion´s metasystem. So it has to secure a place on the channels passing through the control function and the coordination channels of the upper recursion (see Figure 4.2), to benefit from their inputs and let the other subsystems benefit from its outputs to these channels. This is the reason for the representation of the lower recursion on the upper recursion´s map by a square box (representing the lower metasystem) laying on the channels coming down from the control function and the connection channels with the service function. The lower recursion´s implementation region in this case is represented by a circuit facing the environment of the recursion and has, as it is the case, a relative environment that would be the environment of its lower recursion mapping (i.e. the



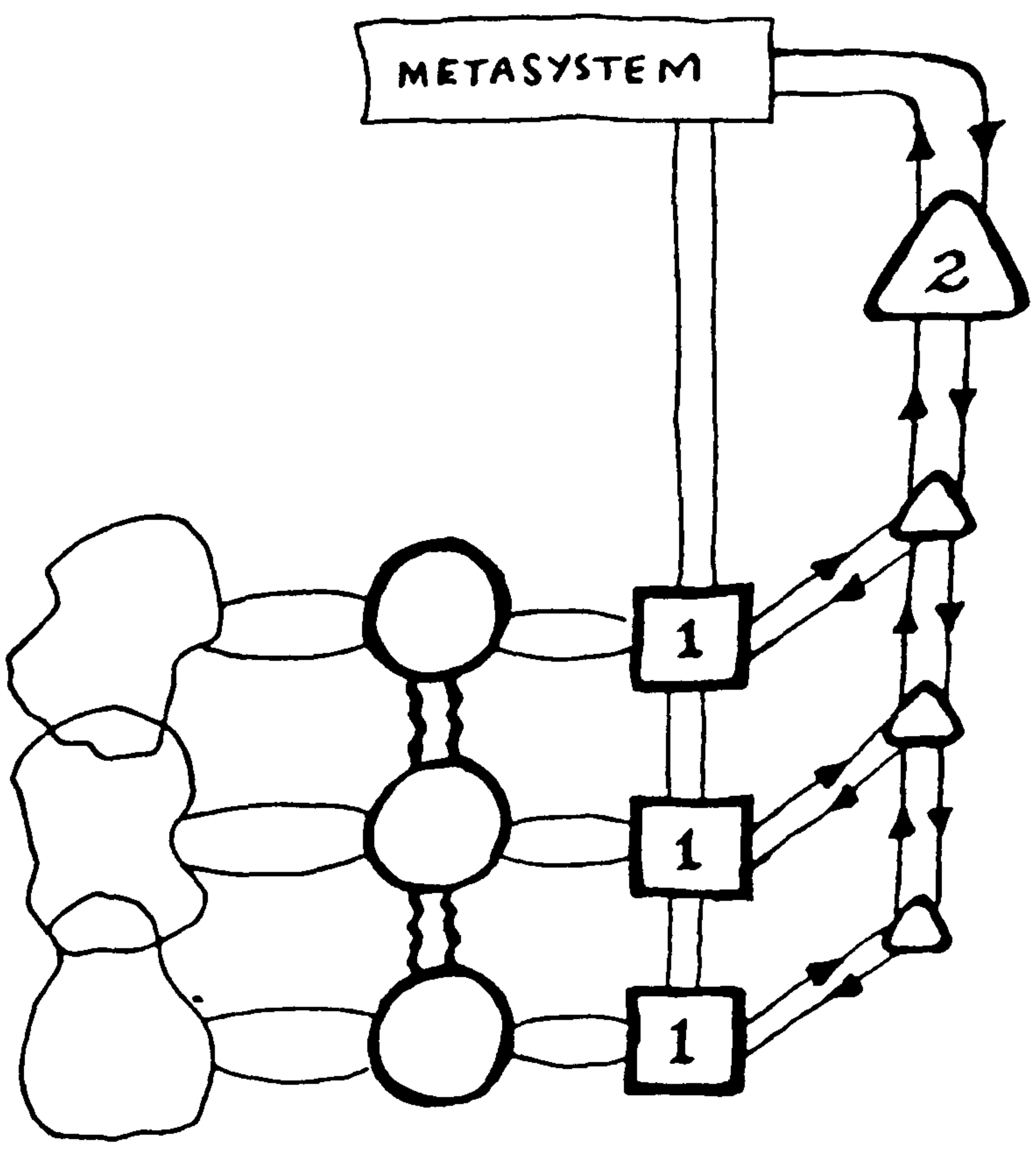
The start of a planning model for Recursion *x*, showing the routes of its interaction with Recursion *y*

Figure 4.2, source: 'The Heart of Enterprise', Beer

environment will follow the same recursive nature as the system itself).

4.3.3. The implementation region

In Figure 4.3, which represents the INSIDEandNOW region of the recursion in view, the functions are shown to be linked by a double line. These lines represent the communications channels which would facilitate the flow of the control, coordination, and services activities, while the double line represents the to-from flow. System One in the model, as we have said earlier, is the operations function of the system. It is the section which is bringing into reality the corporate policy of the system as laid down by the functions responsible for that policy. By design, it has a certain degree of autonomy, as can be seen from its links with control in the metasystem. The service provided by System Two is the liaison between all Systems One in its recursion with overseeing by the metasystem of the same recursion. This in fact is the soft coordination required with the minimum attention from the metasystem, or with no attention at all in some cases. Whereas the hard coordination, which is the control within the autonomy context, would come solely from the metasystem. The control in the autonomy context would in fact take the shape of, either suggestions to the subsystems which are formed from a consensus among them (this could be a tactical plan agreed

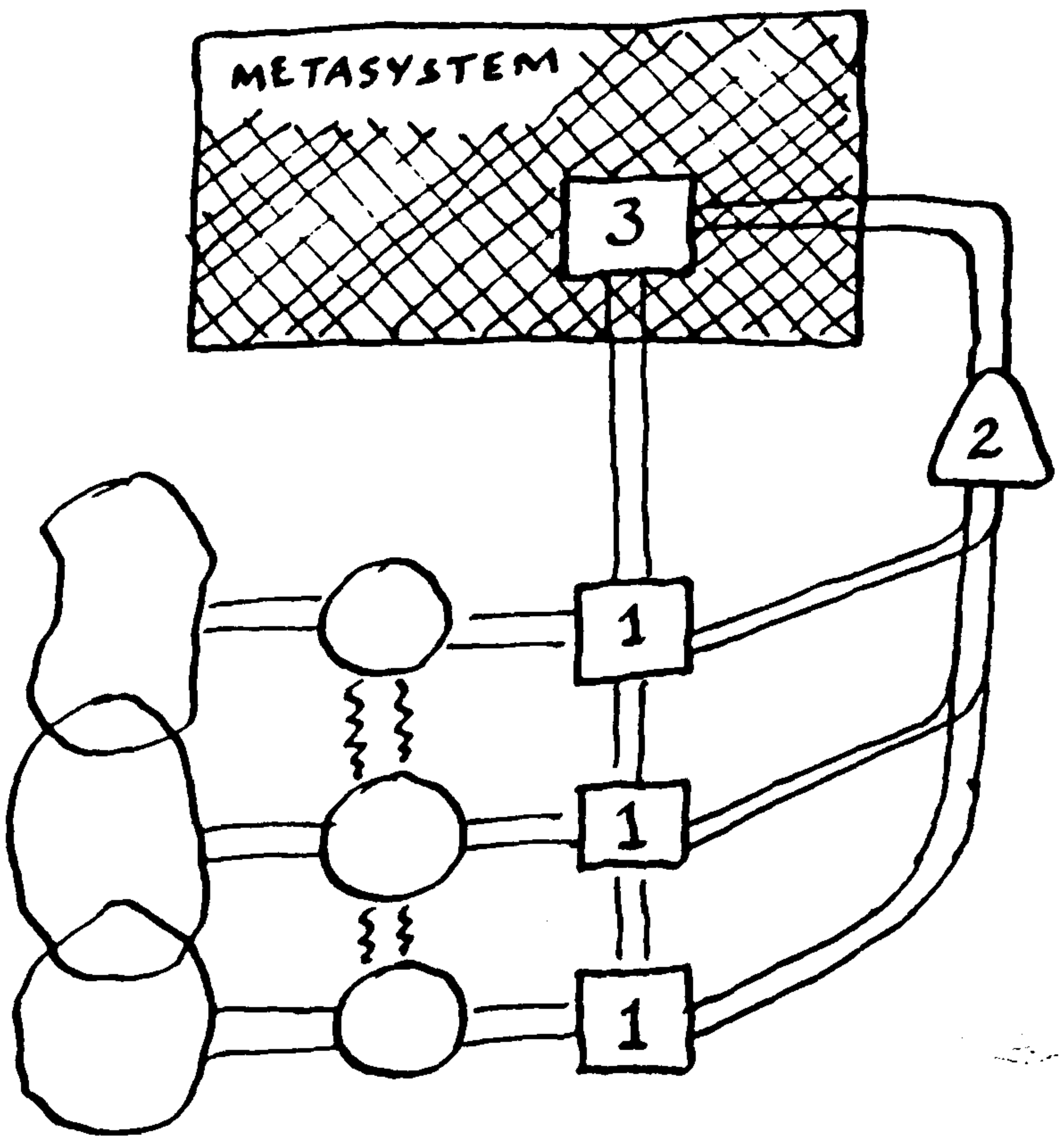


The genesis of the anti-oscillatory System Two. It is necessary to any viable system. Therefore it can always be discovered; but usually it goes unrecognized

Figure 4.3, source: 'The Heart of Enterprise', Beer

between them under the supervision of System Three), or a directive if and when a subsystem is seen to be departing from the corporate policy (i.e. the subsystem is in the direction of violating its autonomy privileges), or in the case of a change to that policy. The metasystem maintains its knowledge about the actual performance of the implementation subsystems by a direct monitoring channel.

System Three is perfectly qualified to form the link between the metasystem and the implementation part of the system because of its dual involvement in both the actual implementation and the aspirations of the system (see Figure 4.4). One might think that System Two is in fact performing a metasystemic role and therefore should be included in the metasystem region, as is the case with System Three, but its obvious role of servicing the ´inside´ needs of Systems One would put it firmly in the implementation region. On the other hand, the freedom given to Systems One by the autonomy would leave System Three with no real direct involvement in the implementation operations, while its role in the metasystem region is more active. This is why it is included in the metasytem box. (The box representation is in fact for the benefit of the upper recursion´s mapping where we need to represent the metasystem´s three functions together as the directorate of its implementation systems which are placed on the control channels.)

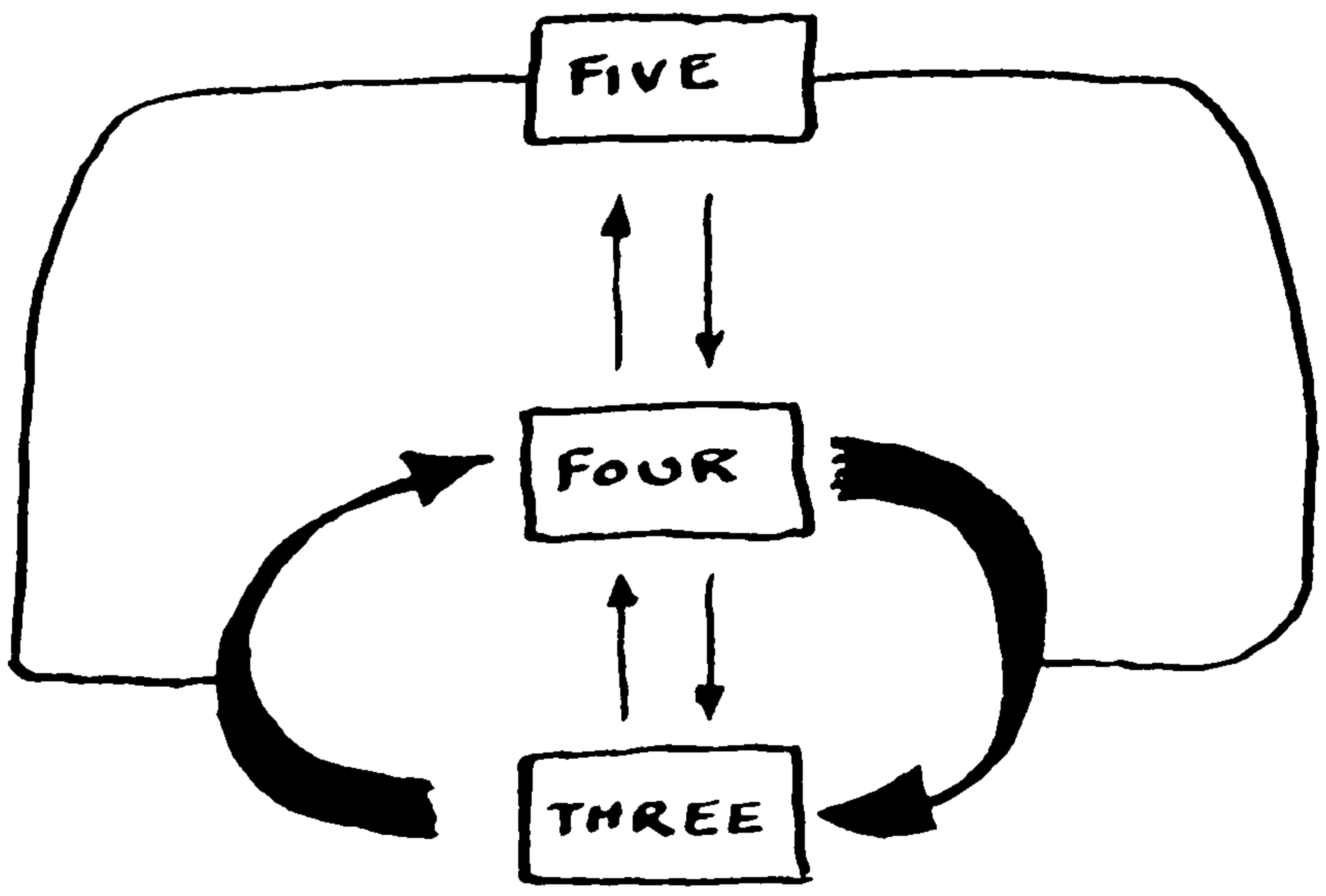


The emergence of System Three

Figure 4.4, source: ´The Heart of Enterprise´, Beer

4.3.4. The metasystem region

The metasystem represents the OUTSIDEandTHEN region of the system (see Figure 4.5). System Five function, as is traditionally performed by the highest ranking personnel in the recursion, is shown at the top of the map, but which should not confuse us as to the rank of personnel performing the other functions. The map representation, it should always be remembered, is depicting ´functions´ and not ´departments´. Thus, even though System Four is mapped in between Systems Five and Three, it does not necessarily mean that people performing this function are of lower rank than those in System Five, and of higher rank than those in System Three. (The same is applicable for all the other functions.) The main emphasis of the model is on the functions and their channelling through the system. System Four is the main contact, if not the only one, with the outside world. Its main output is communicated to Systems Five and Three and, as the specialized analyser of events, it will receive back queries or arguments from these two functions, a role which necessitates its positioning between them. The main circular causality will be between Systems Four and Three because of the urgency to react to the outside world, where System Four knows better about it and System Three can adjust the internal performance of the whole system according to the requirements. While this is



System Five conceived as the monitor of Three-Four interaction, in which their respective high varieties are deployed to absorb each other

Figure 4.5, source: 'The Heart of Enterprise', Beer

taking place, System Five is made aware of these trends, and of the attempts to update the state of the system (which is also available to System Four). If the need arises for the intervention of System Five, and that is when it becomes necessary to adjust or even to re-draw the corporate policy, it will do so by linking itself to the Four-Three circuit to bring about stability, as it is the only function qualified to do so. As the system survives the outside influences, it returns to its normal (acceptable) stability level, with System Three performing its routine function.

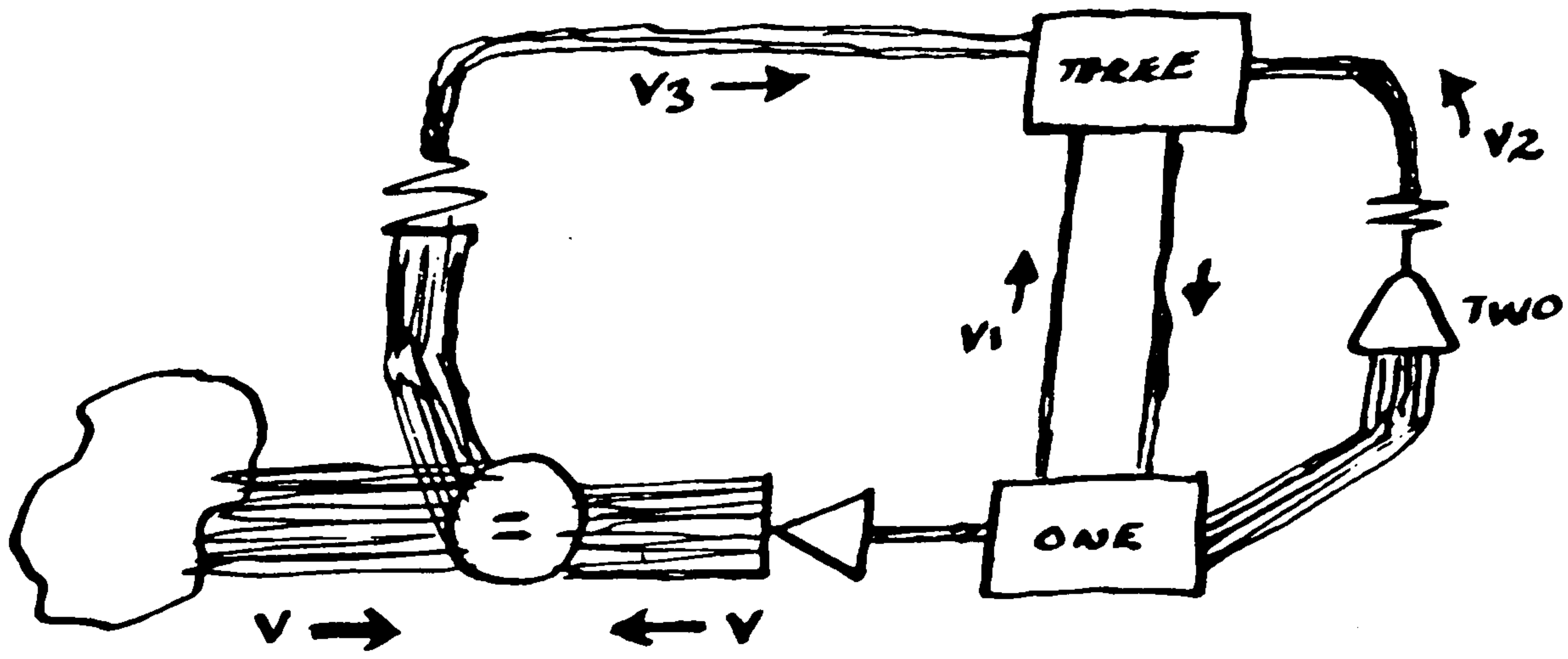
It has been shown before that, due to the freedom granted to System One, System Three's intervention is limited to what can be described as acceptable intervention through its command channels, and its link with System Two. But the reality of the existence of a corporate policy for the whole system necessitates that the metasystem, through System Three, should be in direct touch with the actual performance of the implementation subsystems (Systems One), because if any one of them is going to violate that policy, then System Three should inhibit its freedom. This can be made possible by a communication channel linking System Three, on behalf of the metasystem, directly to the implementation regions of the subsystems to monitor their actual performance. (More reasons for needing this channel are given in Section 4.3.5.)

4.3.5. The variety exchange

The power of the model is in its depicting of the functions of the viable system and the information (variety) flow inside such a system. The placement of the human and material parts of the system, within this context, is being dictated by the requirements of the functions, and has no consequences for the model as long as they are placed on the right channels of communication. This being the case, this model can then be a valuable tool for the investigator (system analysts, cybernetician, etc.) of any system. By mapping the organization of the investigated system on this model, the weaknesses of the various functions can be identified, and the information flow can be examined, which can certainly lead to finding the correct solutions and a better efficiency for the system. As a redundant conclusion from the previous one, a system designer can certainly use this model to build a new organization, thereby insuring a degree of stability right from the beginning. This is what is being done in this work.

The five functions of the viable system are in continuous interaction among themselves. This would mean a continuous variety exchange alongside the communication channels. Ashby's law also tells us that the variety content of each interacting function in the viable system must be matched by the partners of this interaction. If we look at the interaction taking place inside the implementation region, the first thing that comes to mind is that, since System One would naturally possess a massive variety to deal with its task, and since System Three is usually dealing with more than one System One at a time, then System Three must possess (or deploy) a huge variety that would inevitably lead to the extensive use of the command channel (if it happens to be the only available one). It would thus result in the inhibition of System One's freedom, and so the viable system would be violating its own cybernetic principle of freedom; thus the matching must be done by other means, and such means are available.

When we refer to the cybernetic freedom, we are by no means referring to an open-ended freedom. All the properties of the viable system mentioned before (Section 4.2) are there to fulfil the corporate purpose of the system. If it seemed to a subsystem that it is easier to depart from that purpose than to change and adapt in the face of the environmental influences, then it would be violating its relatedness to its system. Not only that, but the system's responsibility is to keep this subsystem within the 'whole', otherwise the viability of the whole system might be threatened (Section 4.2). Therefore, constraints are called for to safeguard this measure. Such constraints can in fact be exerted on the subsystems by defining the limits of their cybernetic freedom. They come forth when System One accepts the corporate policy, thereby accepting by implication the limits of its freedom. This would give System Three the opportunity to define the size of the variety set for each System One it can match. This is the first step in containing Systems One's variety, for using the command channel outside these limits is an acceptable intervention. The next step is to deal with the variety which is in between these limits without the extensive use of the command channel. This can be done by allowing System Three a direct interaction with the operations of System One, with the approval of their management, thereby establishing a communications channel that will not in fact exert any command function at all, and at the same time possessses a huge variety. A good example of such a channel is the AUDIT in the current accounting practice. Auditing has a very large variety content by its nature, yet is still done with no intervention—or at least belongs to the minimal intervention category. This is by no means a violation of the freedom rights of Systems One since this channel does not exert any intervention command. (It specialises in gathering information only and if it happens that a human is performing this role he should be aware of this fact and refrain from any act outside this role.) So it can be seen, therefore, that the command channels (i.e. the vertical channels in the model) need not be extensive, and at the same time Ashby's law is observed (see Figure 4.6). Both



Variety absorption by autonomy

Figure 4.6, source: ´The Heart of Enterprise´, Beer

these stages of variety attenuation are in fact served by the last-mentioned channel, which is called the ´monitoring channel´.

In cybernetic language, all that we have been describing above with respect to variety is what is known as ´attenuation of variety´. In Beer´s terminology, the attenuation can be done by ´variety engineering´ which involves the concepts of variety ´filters´ and variety ´amplifiers´. In this work, we are going to avoid any involvement in the filter-amplifier issue for fear of controversy in naming each of them, for what might seem to be an amplifier to the casual observer is in fact a filter and vis-a-vis. Still, we can assert two principles regarding the variety engineering: One is that the variety of an element in the system cannot be amplified from within but only by imported variety (i.e. computers, courses, databases, expertise, etc.). The other is that the variety can only be decreased through time or by constraints if there is no other opposing variety to destroy it.

The fact is that we do not need to use names in our variety engineering as long as we can reach the required attenuation. We only need to employ the cybernetic principles of variety to which Beer´s model is adhering, and to be aware that there are ways in which the variety of the element concerned can be increased, decreased, or both, by design.

4.4. Outline of the Suggested Organization

We are adopting here a convention in referring to certain terms such as: system, project, plan, etc., to distinguish between their general reference and the specific reference intended by us. The specific reference term will start with a capital letter. For example 'System' is referring to the one we are studying here, i.e. the Plan Follow-up System; 'Project' is referring to a ranking subsystem in the suggested organization, while 'project' is used in its usual every-day meaning (similarly is the case with Sector, Field, and Contract); 'Plan' is referring to a specific plan, i.e. the National Plan.

4.4.1. The mapping of the System

Our System is modelled on Beer's viable system, and breaks down into hierarchical ranks (recursions)—exactly four in our case/example.

Any planning system will be centred around one plan, and therefore follow the same principles of organization (be it centralized for the whole country or not). The example chosen here is the most complex case in the field of planning, that of the entire country.

The highest rank of the System has its operational elements (Systems One) as the Sectors of applications. In the example given (a rough system map is shown in Figure 4.7), there are four sectors of specialized application,

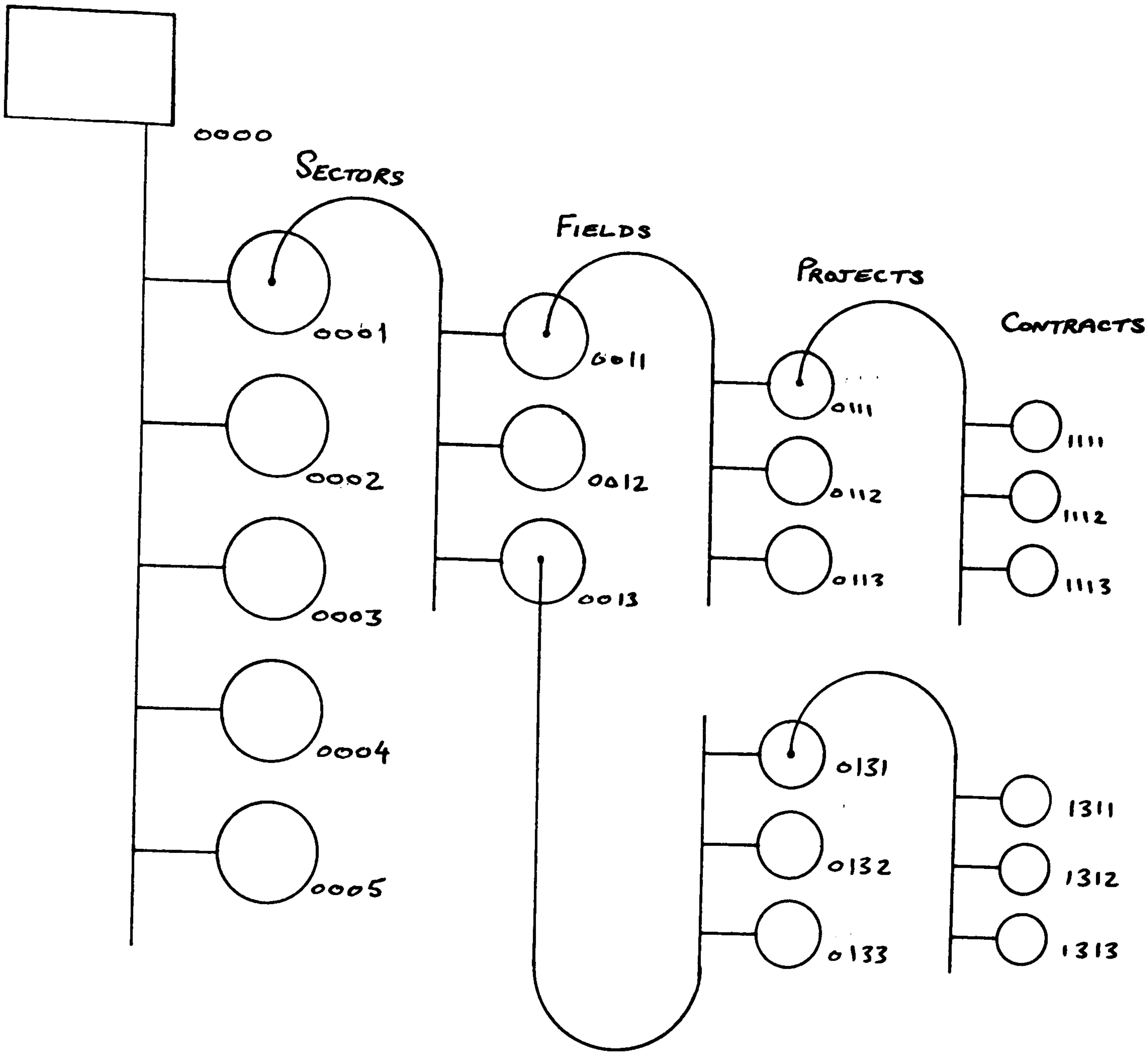


Figure 4.7, mapping the System

namely: Industry, Agriculture, Education, and Services. We add one technical support subsystem for computer systems programming. (This subsystem should not be confused with those belonging to System Two function since it is a productive system and has no role in the liaison between the subsystems.) The metasystem of this rank is the national body directing the National Plan.

Rank Two of the System (i.e. the next recursion after the highest rank) will be the subsystems of each Sector, which are the fields of application within each sector.

In the same way, Rank Three is all the projects involved in those Fields.

Rank Four, for the purpose of this example, is the last recursion, and consists of the contracts of each Project. These contracts are either national or international, and they also differ in the approach adopted towards them. While all the international contracts are planned and implemented by a different system (i.e. the contractor corporations) and can only be monitored, some of the national contracts (i.e. with suppliers) will require planning and implementation, and therefore come into the scope of the Plan operations as well. Translating this into the system map (Figure 4.8) will show that a certain contract, especially if with supplier, can in fact be a Project in another Field, and therefore in a higher rank and another subsystem. (It is very common in the multi-year

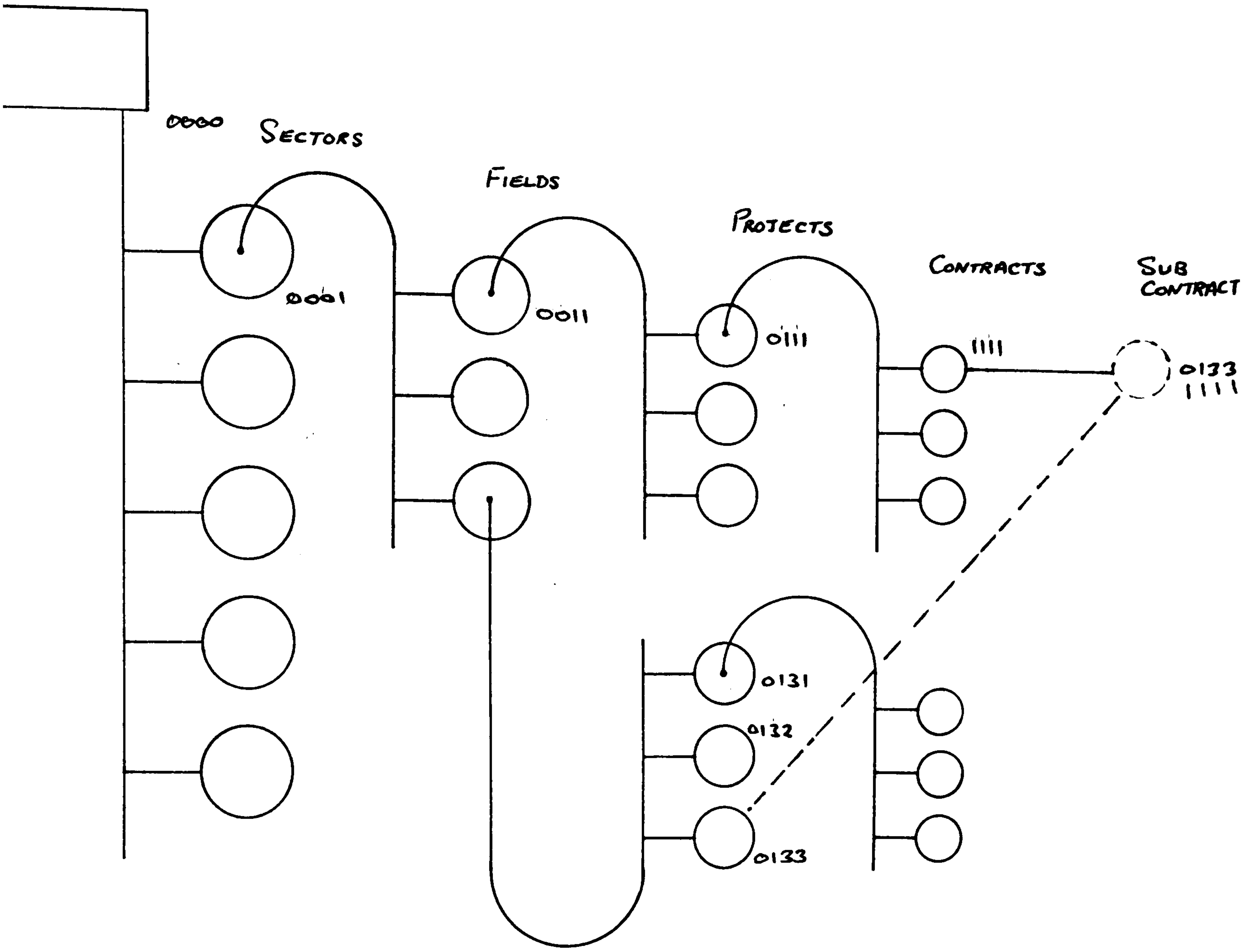


Figure 4.8, extending the mapping of the System

plans to have planned productive projects that can be finished in the early years so that their products can be used in finishing other projects in the later years.) In a situation like this, this contract is no longer a Contract subsystem, but is with a supplier to a contract (i.e. it is a sub-contract), and will appear on the map only in its capacity as a Project. The responsibility of the Contract subsystem, to which this Project is supplying, is therefore minimal and confined to the role of an external source of warnings about delays or quality of products transmitted through different channels. It should also be noted that the positioning in a certain rank represents the importance within the System and the degree of attention required by it from the higher authorities (in addition to the degree of both autonomy and authority assigned to it). The system map of Figure 4.8 also shows that it would not be feasible to extend the number of ranks to five (i.e. suppliers to the contracts) for some of the suppliers will be mapped twice or more on the system with different ranks and different subsystems.

4.4.2. The problem of the two planes of mapping

The system map shown in Figure 4.7 can only work properly and efficiently in the same plane of dimensional space--a case which it is very difficult, if not impossible, to realize in any conventional organization for this system.

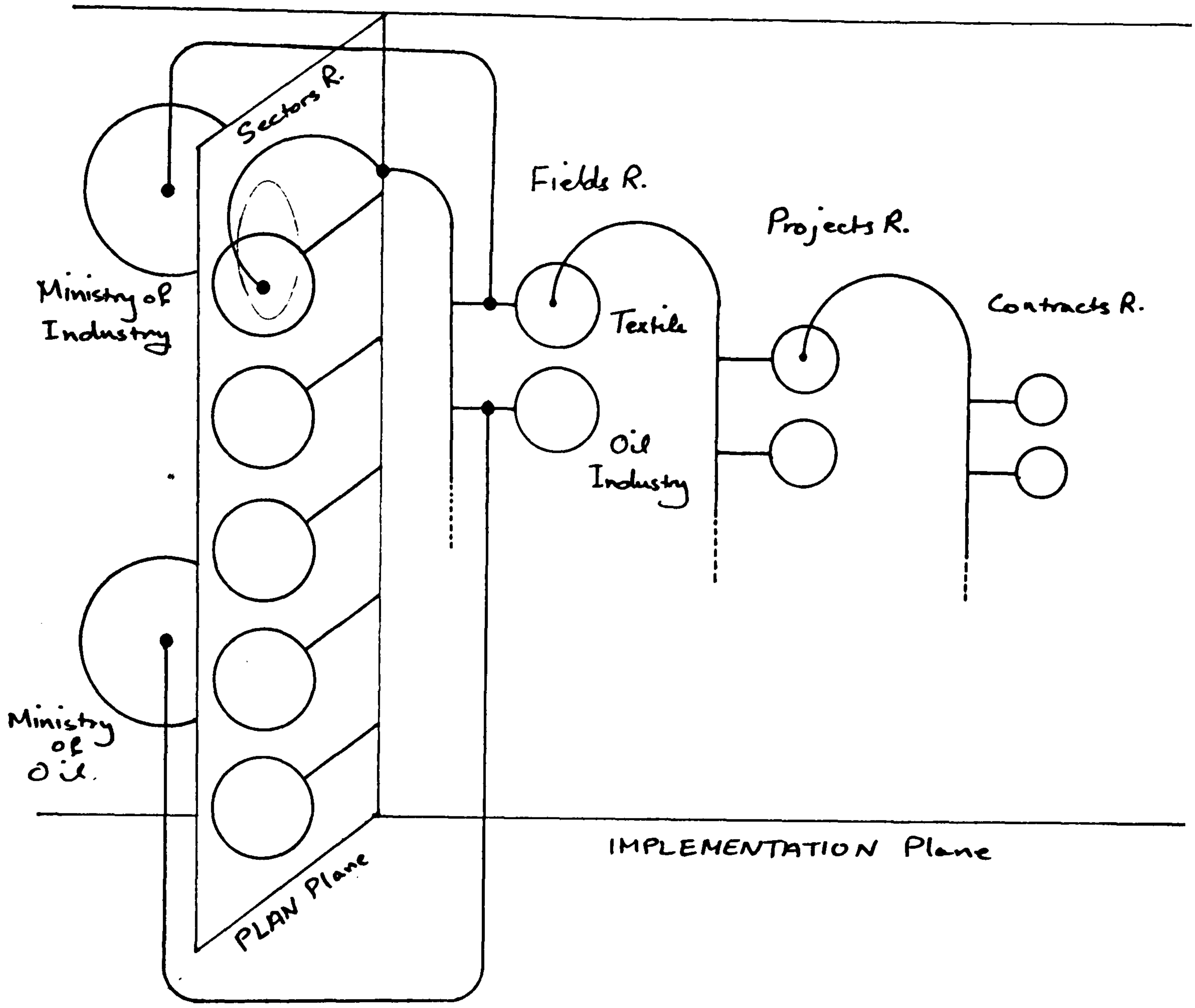


Figure 4.9, the problem of the two planes

To clarify this statement more, it should be apparent that from Rank 2 (or at most Rank 3) onwards, the physical existence of all the elements of the system will be scattered all over the country's establishments with their physical links and actual organizational control lying with the corresponding establishment (i.e. different systems acting as host systems). While Rank 1 will be in one mode (plan orientated), the other ranks will be in a different mode (real life orientated). This is a situation that will necessarily bring a conflict of interests, and therefore place the other ranks in a different dimensional plane to that of Rank 1. Figure 4.9 illustrates this situation, with the mapping showing two major deficiences arising from it: The first is that there will be no straightforward channels of communication with the lower ranks. The second is that a direct control by Rank 1 cannot be established over the lower ranks.

In practice, therefore, the system map of Figure 4.7 cannot be realized with all its elements existing physically in the same dimensional plane.

NOTE: Any major reorganization of the Global (government) System as a whole, to remedy this situation, will face the same problem (i.e. the conflict of interests); and any establishment of new elements within the other systems, to carry out the task of planning follow-up, will have the status of actual presence and artificial involvement, and therefore be rendered obsolete—which will mean in fact that the old map is working and not the new.

It is because of the nature of the system that we encounter such problems, and the solution lies in the nature of the system as well. Being a 'planning-related' system, it is confined to the tasks of manipulating situations, analysing events, redrawing tactics or even strategies, and containing performances within set boundaries (social, political, and financial), all of which are situated in other systems (outside our System), and finally to making sure that all the other systems will reach (safely) a set target at a set period of time.

What creates the problems is that all the previously mentioned tasks are highly authoritarian and, to make the situation worse, they are performed by elements which, although situated in the user system, belong primarily to another system, i.e. their loyalty is to the other system and not to the host system. Part of such a problem bears resemblance to that of the company accountant, but being a real part of the system the accountant's loyalty is not questioned, and furthermore, his task is appreciated by the metasystem and becoming increasingly understood by the subsystems. It is the actual presence in one system of a person or persons whose criteria of reference are in another system whose instructions will be highly authoritarian, and in addition to this are normally going from a lower rank to a higher rank. If we wanted to avoid this friction by making the findings and results of analysis made by the 'planning' unit in the user system go to the 'planning' system (Rank 1), and then making the instructions (which will certainly be coming from a higher rank) come back to the management of the user system, then the 'planning' unit will be seen as a spy nest—a situation which will bring us back to where we started.

It seems that everything is concentrated around the human involvement of one system with another; therefore if we eliminate this factor from our design, a reasonable settlement can be reached.

The solution we are proposing also means that our system will no longer be required to extend in a different dimensional plane.

To achieve this solution, our system may be extended artificially in the same dimensional plane by means of an electronic communications network (as long as the five functions of the viable system are performing properly it does not matter how and who is behind their performance) with the ranks of our system made able to talk to relevant computers that are put in relevant places in the other (real life orientated) dimensional plane. The computers are put there with an initial aim to help implement the planned projects, and are operated by a programming system that can support the operations of the implementation and at the same time extract the required information for the follow-up task. By talking to these computers, the upper corresponding rank's computer gets its information does the analysis, and reaches the proper decision. This is then fed back to the lower rank's computers in the form of a 'programming' instruction to alter the planning programmes accordingly, which had been fed initially to these computers as terms of reference at the time the plan was put into effect. The human workers in the user system, who depend on the computer at the beginning of each working period to give them the instructions, will receive new altered instructions and, even if they know there have been some changes, they also know by now that these changes have come from high up, where the interest is of a broader scope, and more importantly they know that they supplied the information that led to these changes in the first place. These computers can also talk to each other horizontally but within the 'System Two' context. That is, the computers belonging to one recursion are linked together to enable them to coordinate their activities directly.

An important issue here is the preservation of the user's autonomy within his own boundaries. Thus, clear boundaries should be drawn around the mentioned programs to make the access of our system of a 'read only' nature inside these boundaries, which should be sufficient to get the follow-up information and at the same time to prevent the user system from exceeding its autonomy, by reversing the situation outside these boundaries. Another technique with an even higher autonomy can be followed by letting the user fill in the required information on a specially designed form, which he will transmit through the communications network to our system without the need to access the user's information by our system. This technique would give the user greater freedom in producing his own computer programs to support the operations. With such a set-up it can be designed so that the user may allow access to his files by our system using a key under his own control (i.e. with his consent) for the purpose of extracting information or auditing.

The follow-up task is seen here as one of the planning system's essential tasks. Its responsibility, as well as the other tasks of the planning system, is to BENEFIT the other systems involved with the actual implementation of the plan (i.e the user systems), in addition to making sure that everything is going according to the plan.

A double-standard responsibility will arise when viewing the system from outside and from inside respectively. The user systems see from outside the purpose of our system as to benefit them by being an efficient reference to the plan and an informative side when it comes to schedules, timings, and budgeting. The viewer from inside will see the system, in addition to being all these things, as an overseer of the user system.

This clash in the viewing of purpose has a damaging effect on the performance of our system, and indeed on the implementation of the plan itself.

Each side in this problem has legitimate objections to the extremes of the other, but at the same time these extremes are justified in the eyes of each side. For the user system, autonomy is a very serious issue, especially when it is threatened by a foreign system. Control on the other hand is the safeguard of the plan for our system, and there is no compromising about it when a certain project (i.e. a unit outside the system's direct organizational control) is not obliging it.

To avoid this situation, our system is designed around the purpose of SERVICING the other systems with the minimal intervention possible, and at the same time servicing the metasystem of the Global System (i.e. the government).

For the tasks of our system to be seen as a service given by it to other systems, it is required as a first step that these tasks and the organization of the system should be laid out very clearly. It should be made clear that:

1) This system is serving the interest of the metasystem of the Global System; therefore there is a direct obligation towards it.

2) The principles of control applied by the system against the user system are of benefit to the user system in the first place—because these principles are applied to all the other user systems, and because if every other user system complied with the plan there should be no reason why a user system would not reach its target safely. Therefore, if a user system is denied a request (or prevented from doing something), it would be helpful if the information associated with this decision is made available to it.

3) Every step taken by our system is done according to an accounting system of some sort (money, material, men, equipment, power resources and most importantly the time).

4) The anti-corruption measures in our system should be protected (and monitored) by the user systems for their own good,and any gap in these measures would benefit only a minority.

The system programs could contain vigorous measures to prevent corruption. This obviously will not eliminate corruption from the system completely, but it can reduce it considerably by decreasing the circle of those who can manage to do it. The introduction of computers into the system itself means an increase in the sophistication of the system (i.e. the variety). By increasing the variety of the system we insure that less corruption variety can penetrate the system than before. Such variety engineering is ensured by:

1) Designing anti-corruption procedures in the system programs.

2) Only persons with computer expertise are now suspects.

The job specification for all the people involved in the working of this system, apart from Rank 1, whether they are placed in our system or the user system, is mainly that of an OPERATOR of a microcomputer. Basically, they all need to be technically qualified to operate the computer and they need no other technical qualification because they are using ready written techniques. What will differ is the work load at each Rank, where the higher ranks will spend more time dealing with greater traffic.

An effective rewarding system, if established, can serve as an incentive for those operators to develop the techniques they are using. This would make it more beneficial to publicise their discoveries rather than keep them to themselves. With the presence of the technical support subsystem, such techniques can be tested and verified, and finally introduced to the whole system. A test period can be allowed into the system to serve these experiments, by setting the whole system into a ′test mode′ run during the lunch time for example, and by specific permission of the higher rank.

V. REALISING THE SUGGESTED ORGANIZATION

5.1. An Outlook

Each subsystem on the system map at all ranks (i.e. every Contract, Project, Field, and Sector) is a crossroad in the communication grid of the system, and the information stream will be terminated at this organizational node either for local use or to be passed on. At this node there will be a section (be it a human or machine) at which the termination is done; this section is a TERMINAL POINT (TP).

At each TP of the total system, there should exist a communication point. This communication point (CP) is not considered here in the sense of hard wiring, rather it is an environment suitable for the act of communicating with the immediate environment of that TP´s site, plus a suitable communication channel (CC), again not in the sense of hard wiring, to transfer information to another TP in the system. The acts of communicating between the TPs, that are affected by the transfer of information, can be achieved by hard wiring (equipment) or otherwise (human). This is to allow for a feasible investment of resources, while retaining the required degree of complexity at each TP (i.e. the Law of Requisite Variety is observed at each TP).

In general, hard wiring will be involved, and in particular computers. The discussion which follows will assume this situation throughout the system but, whenever the application dictates otherwise, the same principles can be followed to organize the communication. The size and type of the computer used will depend entirely on the workload and type of application at each TP, but the use of microcomputers is envisaged for most ranks as sufficient. Judging from the rate of progress of microprocessor technology, it is envisaged that this technology will be capable in the future of meeting the demands of our system.

The discussion in this chapter will be divided into two major areas. The first area will be the acts (events and procedures) of communication at each TP and between the system's TPs as a whole. The second area of discussion will consider the equipment and software suitable for the job that are available for the user, and thus the modifications thereby required in the first area to make it workable by the available equipment. This enables us to preserve the original ideas until the time comes when development in the hardware and software make it possible, while making the system usable in the present day.

Initially the workload will become greater as the individual TPs are considered in the bottom-up direction through the ranks, but there is no reason for this situation to continue as the system settles down through time, and the autonomy designed for the subsystems takes full effect, at which time the work load should be distributed uniformly throughout the system.

As for the storage requirements, they will depend on the information requirements of the autonomous subsystems, where only locally-used information is stored. The storage organization envisaged is that the total information used by the system is going to be scatterd throughout the system, rather than concentrated in one or few points to supply the TPs with their information requirements. An exception is the one TP in the entire system called Terminal Zero, T0, which is situated at the directorate of Rank 1 and which will serve as a databank for the system and other systems in the country.

5.2. Analysis of the Information

Different types of information will enter the system, from which it will extract the knowledge necessary to maintain the operations. If we examine closely one typical operation, namely the drawing up of a job schedule, we will encounter three types of information. The TP responsible will have to process in the first stage:

1)The target which must be reached according to the plan schedule.

2)The time period allowed by the schedule to reach that target.

3)Material required for the scheduled period.

4)Equipment required for the scheduled period.

5)Human resources required.

6)Other specific details depending on the application.

From this information the user will know ´What is required to be done?´ during the next period.

In the second and final stage, the information needed should tell the operator ´How much can be done ?´ of what is required; this can be achieved by processing the following items:

1)Inventory (items available).

2)Equipment and its present location.

3)Items and their suppliers. } Usually
} considered
4)Contractors and their specializations.} by higher ranks

Hence, information about the availability of material, equipment, and those who can do the required work should yield knowledge about how much can be done. (There will follow a tentative movement between the two stages until the final decision can be made, but we are not going into this here, because it has no effects on the types of information involved.)

This same operation may be performed by any TP regardless of its rank. The only difference is the kind and scope of information required to be moved through the ranks, to be used by the authorized TP´s.

The group of information items used in the first stage of the operation can be discarded once the job schedule is finalized and only the final information about the decision may be stationed here. All the working information items which led to deciding on the individual items used in stage one, plus the items themselves, are of no value once the schedule is finalized and the decision is made. However, some reference information is needed here to assist (technically) in reaching the decision, such as specialized technical packages and expert systems.

The group involved in the second stage is essentially required to be stationed at the user site, and cannot be discarded at any stage (except by elimination from use). In addition, the same items of information might be shared all over the system regardless of the rank or subsystem. This is a distinct difference to that of the first group.

In the two stages, we can observe three different groups of information involved: one was needed as a reference, replicated at more than one location; the second was needed for the future once it had been created, but only locally; and the third was operational and was not needed for the future once it has been created.

Hence only the first two of these groups are required to be stationed in the system. The first group is relatively static and is not an intrinsic property of the site at which it is stationed (such as those in Items 3 and 4 of the second stage and the technical packages in the first stage). They belong instead to the total System, and are thus a 'Global' type, with their copies at the various sites being borrowed for the duration for which they are relevant to the user site. The user site borrows them from its node TP, which borrowed them from its node TP, and so on up to TO. They have in fact more of a reference nature—helping in making the decisions. The second group is dynamic and is a property of the site at which it is stationed. Its members are of a 'Local' type, with no copy stationed at any other site throughout the system. However, a summary of each will be stationed at each node TP, which is taken from each item stationed at each branch TP of that node TP, in the bottom-up direction.

N.B. The distinction between the local and global types of information is being made here with reference to the record's 'image' of the file which is holding that information. It is necessary to keep this in mind when we come to discuss the 'application' types in Subsection 5.5.2, for there will be a sizable volume of the local type information which is involved in the 'global' type applications.

In short, the types of information found inside the system are:

1)OPERATIONAL: Non-retrievable after summarization (we can consider the final decision as some sort of a summary), and not required to be stationed inside the system.

2)LOCAL: Retrievable after summarization (from the lower ranks), which is required to be stationed inside the system, and no other copy of it exists at any other location but one.

3)GLOBAL Retrievable after fragmentation, required to be stationed inside the system, and replicas of it may be found at more than one location.

After considering what information will be stationed inside the system, we now consider its distribution and its impact inside the system. The second, LOCAL type, will normally enter the system initially from the lowest rank (the Contract rank in our case). The third, GLOBAL type, will normally enter the system initially from the opposite direction, namely from the highest rank. Therefore, the first concentration of information will take place at Rank 4 (R4) terminals and at T0. This first impact of information can be easily handled by R4 terminals, due to the normally large number of them, coupled with the slicing of the whole into portions corresponding to the environment of each contract. The impact of the Global type on the upper-most rank (its point of impact) can also be easily handled. Terminal zero should be capable of taking the full load, whose fragments are spread down to the lower TPs, thereby proportionally reducing the volume required to be stationed, as the rank of the site becomes lower.

As a result of the decrease in volume of information stationed at each TP, which is happening in both directions of impact, the total information should spread across the system in a rather balanced way. Nevertheless, the volume stationed at each rank's TPs would become relatively larger as the rank becomes higher (see Subsection 5.2.1). But this excess (which is due to the Local type coming from the lower rank) can be ignored compared with the initial volume and the much lesser number of TPs at the upper rank.

5.2.1. Organization of data storage

The stationed data is stored in the form of files (tables) at the relevant TP. The records of each file are of a similar nature. The major characteristic of the local files is that no more than one copy of their record is stored in the system. They might be found at every rank (an example of which is the INVENTORY files which we will consider in detail in Ch. 6), or may be confined to certain ranks, due to sufficiency, as shown in Figure 5.1.

It can be seen from Figure 5.1 that each rank's TP stores its relevant table, the number of records of which is equal to the number of subordinate subsystems (branches), with each record holding information regarding a subsystem.

The number of records of an INVENTORY file is different; it is equal to the number of inventory items used by the subsystem, or its sites. Rank 3's TPs will each have their file's number of records slightly higher than that of a Rank 4's TP, and similarly for the upper ranks, because of the facts that each site might store an item which is not used by the others and that the node TP has to store all the items. But these items should not be many, since the projects of a certain field of application would have a similar nature, and so the majority of items used are the same.

CONNO	PRJNO	BUDG_AL	BALANCE	LAST_ENTRY	LOCATION	CONTR_NO	CONTR_NAME

Project Level Storage: CONTRACT TABLE

PRJNO	FIELDNO	BUDG_AL	BALANCE	LAST_ENTRY	LOCATION	NO_CONTS	REALIZATION

Field Level Storage: PROJECT TABLE

FIELDNO	SECTNO	BUDG_AL	BALANCE	LAST_ENTRY	NO_PRJTS	REALIZATION

Sector Level Storage: FIELD TABLE

SECTNO	BUDG_AL	BALANCE	LAST_ENTRY	NO_FIELDS	REALIZATION

System Level Storage: SECTOR TABLE

Figure 5.1, some of the database tables

Many forms of summarization can be performed on information to be stored at on rank to be passed on to the next higher rank for its own use. At the same time, to avoid duplicate storage of information as much as possible, if information is required in the same form by a higher rank, then it should be accessed by it through the lower rank's terminal and storage. Similarly, for the information which will consist of only one, or a few, records to a file, and is not used regularly, such as the contractor record at the Contract TP, storage will be at the immediately higher rank's TP (see Figure 5.1) with access provided to the lower rank's TP (it seems more appropriate to have one file with many records than to have many files with one record).

The type of detailed information, that can be summarized, can be summarized still further at each point of collection. At any TP from R3 onwards, a summarized batch of information will be received by it from each one of its subordinate TPs (branches). The summary of this collection will form the batch sent by this TP to its controlling TP (node).

Such distribution should uniformly decrease the volume of information, and therefore of storage, in the upward direction. This reduction is met by another from the opposite direction, i.e. the top-down direction, by fragmenting the GLOBAL information entering the system from the top. Information of this type includes the Supplier/Item tables, Equipment/Location tables, Contractor/Country tables, and Foreign Exchange Rates.

All information of the GLOBAL type can be stored at T0 and accessed by all TPs as required, but it would be preferable to store fragments of this information locally at each TP, while preserving the original at each higher rank TP. The preference of this method is due to several reasons (this area will be examined in more technical detail in Section 5.5):

1)Security against failures at T0, for other TPs can continue functioning regardless, at least for some time.

2)Less use of the communication network, which leads to:

a)less overheads;

b)less congestion on the lines, and;

c)more efficiency in performing the remaining tasks that are left to the network.

One aspect of this organization is the use of T0 as main storage without endangering the system, which at the same time plays an integral part in the security of the GLOBAL information against damage that might happen at any of the various TPs used. Due to this assigned task to T0, it is necessary for it to be equipped with large storage capacity and faster access time, plus a more sophisticated operating system to support the possible communication of a large number of users. (Compared with other TPs, the high value and versatility of the information accumulated at this terminal will draw a large number of users from outside the system.)

5.2.2. Validity of information

This distribution of information, combined with the high autonomy at each TP, gives rise to a major issue regarding the continuous validity of the information. Part of the distributed information will require changes. The majority of events that necessitate these changes are originated at the different subsystems and, with the facilities available to all TPs, the action of modifying the data making up that information is logically and psychologically better done locally. What is vital in this case is the necessity of update all the copies of the altered record which exist in the system. It is obvious that the total information of the system can be corrupted through time if there is repeated failure to modify the information. This could easily happen if the modifications to the copies (as a result of changing one TP´s record) are left to be done manually by their local users.

To secure this, a global command should be solely responsible for updating the GLOBAL information (no other local command should be made available which can do the job). The Global records of this kind are contained in files which can be isolated from the local updating command by an access code which can be used only by the global command. One example of such a command may be a subroutine, call it CHANGE, which should be capable of the following:

1) Copy the new data from buffer storage to the storage medium and supply the outgoing signal with the necessary information.

2) Transmit upwards and/or downwards a signal to the TP computer(s), containing:

a)a call command for CHANGE on that computer;

b)the contents of the buffer;

c)the name of the file to be updated; and

d)the key of the record to be updated as parameters.

3) On the arrival of the signal at its destination, the CHANGE command should perform the following:

a)open file, the name of which is supplied;

b)locate record, the key to which is supplied;

c)move contents of the buffer to the record allocated; and

d) repeat Step 2.

4) The end of the line should be taken care of by the communication network.

5.3. Analysis of the Communication

5.3.1. The terminal (communication) organization

Following the system map (Ch.4), each autonomous unit (designated as a subsystem on the system map) is considered as a TP, therefore an intelligence facility is required to provide computations and the automatic relay, which is essential for control, of information.

With regard to the system under consideration, there are four rows of computers involved (corresponding to the four ranks) plus one. They are the Contract row, the Project row, the Field row, the Sector row, and a single computer situated at the directorate of the Sector rank. The three in the middle are of the same sophistication, while the first and fifth are different but they all must be compatible with each other.

The fifth row computers (Contract rank) may be stationary or portable, but in both cases a communication line is necessary.

The connection links can be made literally in every direction both ways. Figure 5.2 illustrates a sample of the connections in the system. From this sample one can classify the directions into two general directions diagrammatically:

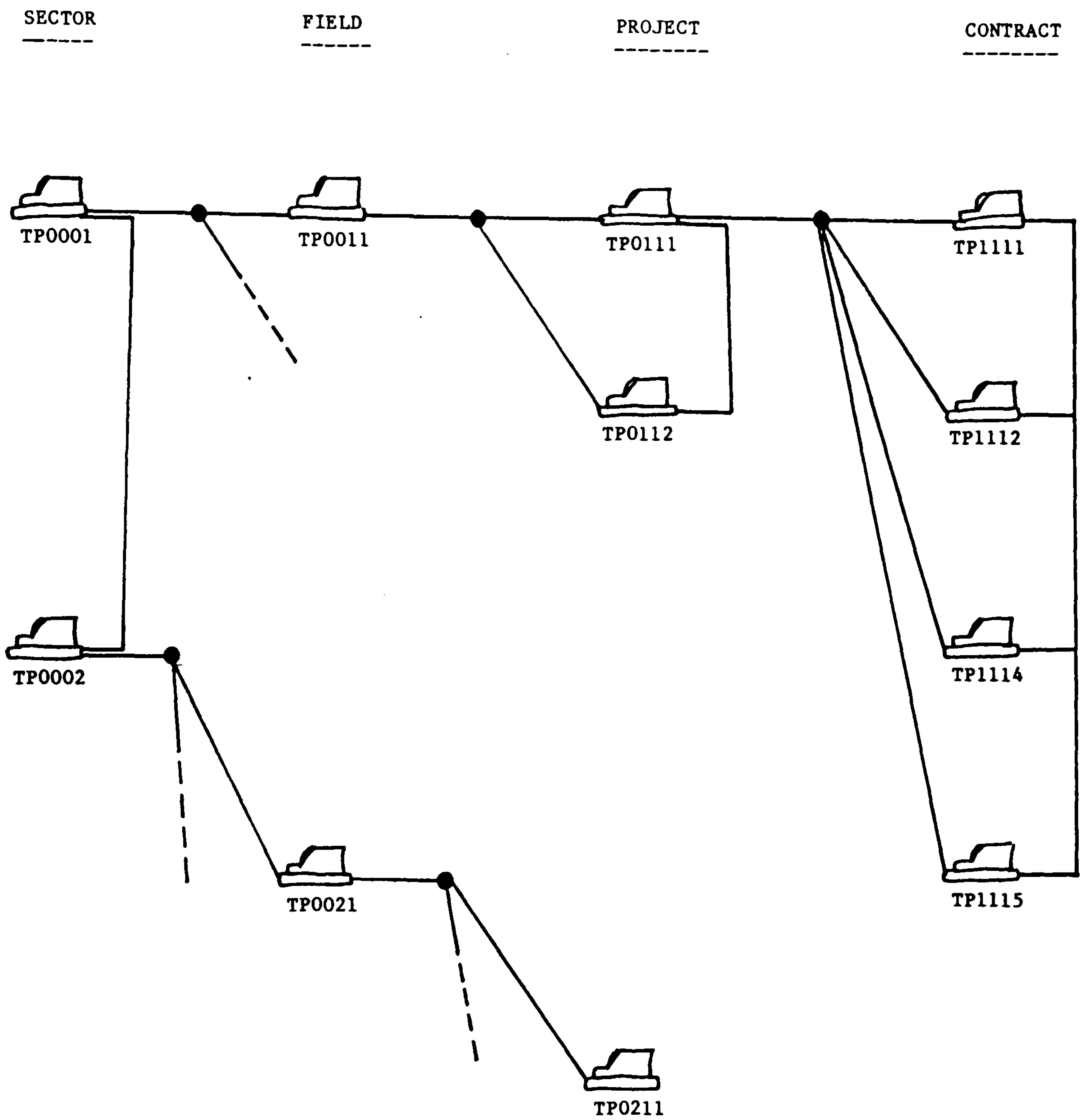


Figure 5.2, sample of the connections

1)Vertical direction.

2)Horizontal direction, divided according to the groupings of the subsystems into:

a)external communication; and

b)local communication.

The vertical direction links each Sector to its Fields, each one of these Fields to its Projects, and each one of these Projects to its Contracts. In other words, this linkage provides part of the enternal links between each Sector's subsystems (the complementing part comes from the Local-Horizontal link). This connection would serve the control and monitoring functions. A communication topology which is suitable here is the hierarchical topology (see Subsection 5.5.1).

The Horizontal direction (considered alongside the ranks of the system) links the TPs of one rank to each other (the fourth rank computers are excluded from this link in the initial stages of implementation). Such connection provides a communication link between all the projects of the system (regardless of their Fields), similarly, all the Fields and all the Sectors. This connection is made up of two separate connections. The first one, called 'local', is made between that group of subsystems which belong to one node at a higher rank (eg. a group of projects of a certain

Field) and will serve the System Two function, thus completing the internal connections of a complete Sector as mapped by the viable system model. A communication topology which is suitable here is the ´completely interconnected´ topology (Subsection 5.5.1). The second, called ´external´, is between any TP of the ´local´ group and any TP of another ´local´ group situated in the same rank. This connection is not suitable in our context except for ´read only´ applications within the freedom of information concept and the propagation of knowledge; otherwise the interaction allowed by such links would get out of hand, because it is not controlled by a System Three type function. We may allow it, however, in the inter-subsystem context as a ´postal service´, and prohibit it in the inter-rank context for fear of rank violation.

With such widespread connections throughout the system, it is imperative to make sure that each channel is used in the sense for which it is specialized, i.e. either command or co-ordination; otherwise it would cause great confusion.

The use of Locks may serve well in this area, where all the tables used can be locked up, except for authorized ´read only´ to allow for co-ordination, and a one-way ´write´ access in the top-down direction for the command chain. We have seen previously that the update action, which requires a write access in the direction opposite to the command direction, is being done by a global command that has a built-in security measure.

The last terminal to consider is the outstanding computer which we called Terminal Zero (T0) at the directorate of rank one, where we have two options to consider, but each depends on the equipment used. The first option is to provide direct access for each and every terminal in the system to this computer. The second is for the user TP to use those TPs which lie on the Vertical connection between itself and T0, as intermediaries.

Finally, an identification code (TPID) is given to each terminal to serve as a key to its computer. This key is capable of identifying the rank number of the computer by the number of digits including and following the first non-zero digit starting from the left of the code, and the subsystem to which it belongs by the numeric value of this code,where the first non-zero digit from the left is the Sector code, followed by the Field code, followed by the Project code, followed by the Contract code, if any exists. For example, the code ´0002´ belongs to a Sector terminal, namely, Sector 2; the code ´0012´ belongs to a Field terminal, namely, Field 2 of Sector 1; and ´0124´, belongs to a Project, namely, Project 4 of the previous Field; while

´1249´ belongs to a Contract, which is Contract 9 of the previous Project. With such a combination key it will be possible, therefore, to address computers and check the access rights of the user computer. The TPID can also be used as a suffix to the file name——a situation which will allow freedom of choice of names for the local operators without fear of a mix-up, especialy when using external central archives for the backups.

5.3.2. An illustrative case

With the proposed connections (communication network) we established the communication channels identified by the viable system mapping.

The automatic relay of information, from one TP to another, is a key issue in this organization and in the communication of the whole system, and can be used here as an example to demonstrate an aspect of the communication grid. The information exchanged inside the system between the different TPs is not entirely for the purpose of storage, since the queries are also involved (regardless of the direction and rank it came from). The autonomy and the communication network proposals will cut down the load on the higher TPs and shorten the response time to the query, while the relay service will make sure that a response to the query is achieved, whether directly or indirectly, by relaying the same message to the higher rank after the time allowed for the first response expires without a response.

Considering the distribution shown in Figure 5.2, a simple but frequent problem is examined. A consignment of supplies produced by Project (0.2.1.1) is scheduled to arrive at Contract (1.1.1.1) on the 15th of March; today is the 16th of March and there is no sign of the consignment. The envisaged activity would be:

1)Manual Procedure

a)TP1111 operator ´informs´ TP0111 operator about the delay (notice the preservation of the command chain), using the ´vertical´ line;

b)TP0111 operator ´asks´ TP0211 operator for reasons plus estimated arrival time of consignment, using the ´postal service´ line.

2)Automatic Relay

There will be a 24hr cycle for each query to be answered, during which time a record for this entry is stored on the Queries Table of the TP originating the query, then the following automatic steps take place:

a)one complete time cycle is allowed for TP0211 to find out and send back the answer; if the answer is received, the query entry is erased from the Q table;

b)if at the end of the cycle no answer is received by TP0111, it will retrieve the query from the Q Table and direct it to TP0011 automatically (the entry on the table is still safe), using the ´vertical´ line;

c)the same as ´a´ and ´b´ take place between TP0011 and TP0021;

d)the same as ´c´ takes place between TP0001 and TP0002;

e)if at end of day 3 (i.e. end of step ´d´) no answer is provided, then TP0001 will instruct TP0011 to implement a new short term tactic, similarly TP0011 to TP0111, and TP0111 to TP1111, each using the resources available to it or that directed to it by a higher rank TP;

f)if the situation persists, then TP0001 refers the situation to its directorate (the planning committee) to redraw the strategy of supplying TP1111.

If at any step mentiond above, the answer retrieved puts the reason as the lack of transportation (i.e. the material is ready to be moved), then the Local-Horizontal communication is used to the full and the External-Horizontal communication is avoided. The following procedure is then initiated to coordinate the resources:

a)TP0111 transmits to all TPs with a three (non-zero) digit code which starts with digits ´11´ (i.e. ´011?´ which is a wild card meaning all Projects belonging to the coordination net of Field 1 Sector 1), and stating its problem, and choosing the preferred response from those received (TP0011 must be informed at each step).

b)If no response is received at the end of the cycle (or none of those received is acceptable), then TP0111 will send the request to TP0011 (note that the message is now a request and not just a query), which may verify all the answers received by TP0111 through its vertical link (monitoring). On the discovery of inconsistency between reality and one TP´s answer, it will instruct (command) that TP to provide the requirements of TP0111.

c)If, however, these requirements still cannot be met, then TP0011 will transmit exactly like Step ´a´ to all TPs with a two digit code starting with ´1´ (i.e. ´001?´ which is a wild card meaning all the Fields of Sector 1).

d)Step ´b´ is activated if applicable between TP0011 and TP0001 (we referred here only to TP0001 in the first rank for simplicity of presentation, however the field TP will be transmitting in duplicate as will be explained in Section 5.4).

e)TP0001 will transmit exactly like step ´a´ to all TPs with one digit code, i.e. all Sectors.

f)Same as the previous ´f´.

At step ´d´, the flow of information will be running smoothly as it departs from the implementation plane to the planning plane with the help of the communication grid; thus no bureaucracy (see Section 5.4), that might delay the action or result in human conflicts is involved. The preceding example shows clearly that, in effect, there are two different systems acting concurrently, each progressing smoothly to its target, with only one shared factor, which is the information. Both systems operate on the same information to perform, most of the time, different actions, although initially both of them aimed at exactly the same result. It is also clear that, if we separated those systems from each other, then, in addition to having a redundancy of information of about 50%, we would observe the two systems moving further apart as time passes, and the conciliation between them becomes harder to achieve. This would only result in harming their progress towards the objective they both set out to achieve.

This example also reflects, to some extent, that the proposed links are capable of providing the channels for the three essential communication functions of the viable system: the command channel, the monitoring channel, and the coordination channel.

The activity we have just examined in the example is just one of many activities which are going on inside the system and making up the operations of the tasks assigned to each subsystem. Underlying this communication network, however, is a network of feedback loops which is made possible by the communication network itself. We may be able to infer the basic structure of this network in its lower levels, but it is almost impossible to map it in detail. Nevertheless, this is not a drawback to our task, as long as we provide the necessary channels for this feedback to take place (see Section 2.2).

Figure 5.3 shows a complete single Vertical line (of one task) to illustrate the basic feedback processes that will take place between the various TPs.

The diagram also gives the impression that the feedback loops (illustrated by arrows and comparators) are fairly easily recognized and are simple and straight forward, which is in fact the wrong impression. The reason for the apparent simplicity is that the diagram considers only one line of communication in isolation from the many other existing in the system which will pass through each TP. In addition to this, the arrows connecting the TPs shown represent two types of information at the same time, namely control and coordination.

With the real complex connection realized, knowledge is gained at each TP by processing the information reaching it, and is proportional to the size of the environment from which the information is gathered.

Each rank will treat the complex web of feedback circuits of its own subsystems as a blackbox, as shown in Figure 5.4, by measuring the total (collective) output of those subsystems at Point A. This is then matched against the plan (or reference) at Point B. As a result, one of three different responses by R3, is sent to its branch subsystems. These responses are either to discourage the deviation from the plan, or to encourage it, or to do nothing when the deviation does not deserve the attention of a higher rank (i.e. dose not call for an intervention). The response will pass through Point C to all the subsystems where each of them will act accordingly with freedom. It is necessary that each of the subsystems be aware of the whole intervention chosen by the metasystem so that its own position, with respect to the total action, is made clear.

Figure 5.3, a single vertical line of feedback

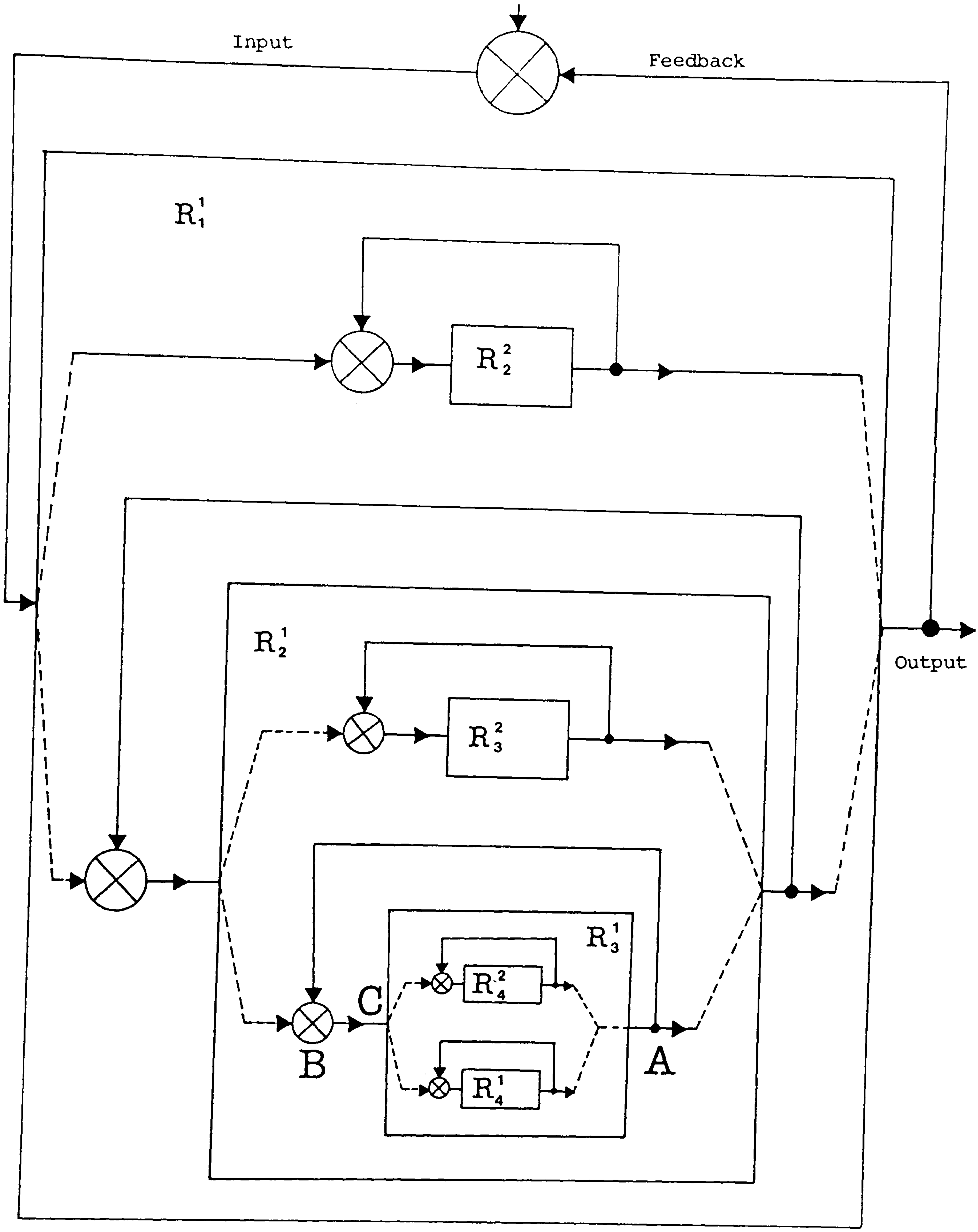


Figure 5.4, the more complex feedback ciircuit

4.4. Two Planes and the Interface Between Them

Until now the system has been presented as if it is lying in one dimensional plane. In fact this is because we are looking at the artificial extension we created before (through the electronic network). The system we have been considering in the ongoing discussion is in fact lying in two different dimensional planes. While Ranks 4, 3, and 2 are in the real life plane (implementation), Rank 1 is in the theoretical plane (planning).

In reality, all humans and equipment involved at the Terminal Points of Ranks 2, 3, and 4 do not belong to our system; on the other hand, all the information generated and activities at these TPs do. Therefore, we can say that we have been presenting the system ´informatically´, or in other words we have been considering the ´soft map´ of the system. The ´hard map´ is something else, where all the subsystems which are playing hosts to the TPs of Ranks 2, 3, and 4, do not necessarily have the same boundaries and ranks in their own real systems as they do in our system. But there is no harm in this for our organization, as long as the communication links do exist between our subsystems, and the boundaries and relative environments of each subsystem are marked by the application software, which has sufficient capacity (variety) for it to function.

For Ranks 3 and 4 (Projects and Contracts), there will probably be no differences between their boundaries and relative environments as subsystems on our system map, and their hard organizational mapping.

Rank 2 subsystems might differ in this sense as they correspond to the government establishments responsible for that Field's activity, as drawn on our map.

As for Rank 1, we have no need to make its subsystems correspond to any establishment, for they are real.

To establish the necessary connection between those two dimensions, we propose the following method: The real communication grid will branch out in two different directions after it leaves Rank 2 in the upward direction (Field to Sector connection on our cybernetic map), i.e. any transmission from Rank 2 will be in duplicate. One branch goes to the responsible Rank 1 subsystem, the other continues in the same dimensional plane (i.e. implementation) to the ministry responsible organizationally for that Field. These links are essential for providing the right decision, and at the same time the right implementation of that decision. Our Rank 1 subsystem should not be allowed to interfere with the detailed implementation, while the ministry should not be allowed to interfere with the plan administration. There are some priorities given to each link to facilitate this understanding.

N.B. It is proposed that the link between the two dimensions should be at one point only. This is due to the fact that the planning dimension has only one physically existing rank, i.e. Rank 1.

The final decision reached at the top will necessarily have two main components: (1) a 'directive' component formulated at the ministry and issued to its establishment (thus the Field) through their link, regarding the implementation of the decision (or solution), and (2) an 'instructive' component formulated at the Sector and passed to the same Field, but this time on their link, which is in line with the decision taken, and referring to the planning requirements. Both these components are received by the Field's TP, which needs to act according to them. If they are not compatible, the Field will have difficulty in implementing the decision, thus has a problem, and will need to refer back, in the same manner, to the top again.

The links between the Field's establishments and their Ministries are essential to complete the command chain of the real organization which is behind the plan implementation. At this point, therefore, it is essential to freeze the command link between Rank 2 and Rank 1, while maintaining a normal link with the Ministries. The Ministries in their turn will sort out any problem they receive, between themselves and the Sectors responsible. Since, when it comes to the Plan's projects, no Ministry would be allowed to issue commands that the Sectors have had no share in formulating or to disagree with them, it followes that the ´short cut´ presentation of the communication link between Rank 2 and Rank 1 TPs on the cybernetics map is not unreal, but was oversimplified for the purpose of the previous discussion. When the Ministry and the Sector disagree, the Planning Board, which is in fact the metasystem of our System, will decide on the final command.

In fact it is at this section of the communication line that the departure will take place from one dimensional plane to another. The transitions at this point are therefore smoothed out by the actual link shown on Figure 5.5.

The straight line link between Ranks 1&2 is in fact a four-cornered link, with its corners as a Field, a Sector, a Ministry and the Planning Board.

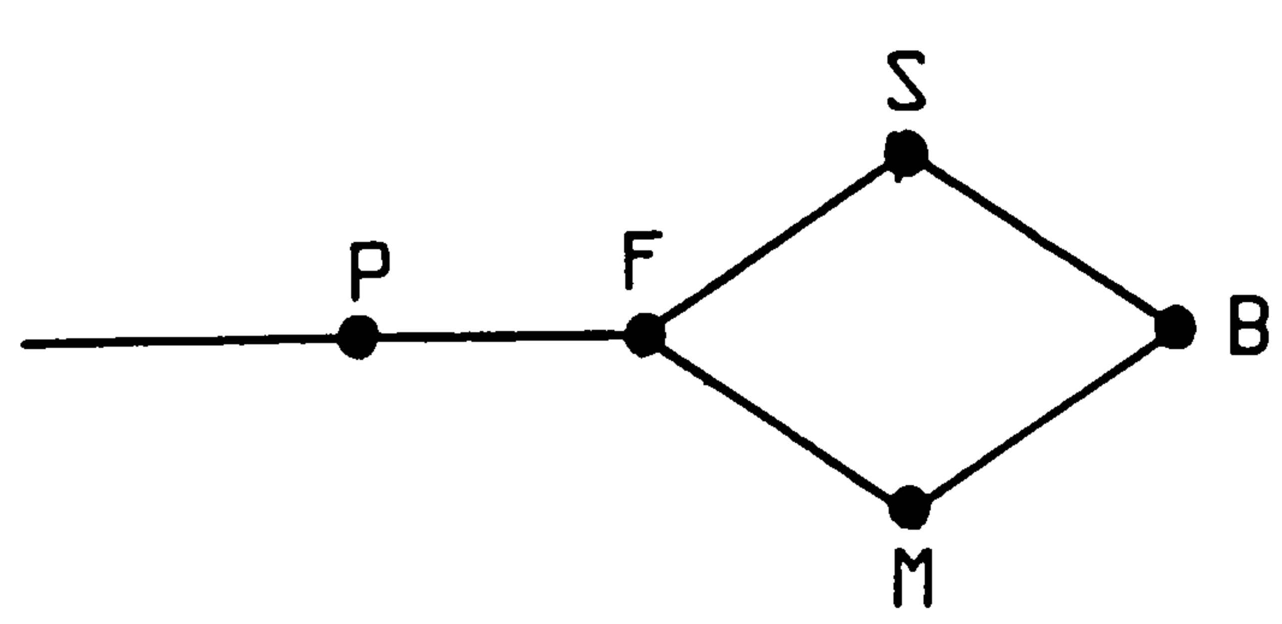


Figure 5.5, the four-cornered connection.

The final command reaching the Field subsystem on that straight line is in reality coming from the Ministry responsible on the MF line after it has been settled through the communication channels and activities of the BMS triangle. At the end of the day, the Field TP has received a command that the Sector would have issued, had it had the command channel.

N.B. The Planning Board has a higher authority than both the Ministry and the Sector which gives it the power to act as a metasystem for both of them, thus controlling and coordinating their interactions for the good of the National Plan.

5.5. Technical Considerations

The solution we are suggesting in this work is based heavily on the networking concept of communication and the use of databases. In this section we will review the technical aspects of both, although we did not actually construct a communication network or a complete distributed database system, for practical reasons. Instead, we simulated the solution on one microcomputer and used a relational centralized type database (dBASE II) as a local database, because of:

1) The unavailability of funds.

2) The time allowed, by the sponsor, did not permit going into such details as the construction of an actual network and the building of a proper distributed database, which would certainly have needed more resources (human and material) than was available to us.

Nevertheless, we see no reason to go into such detailed work at present. Our work is based on the idea of giving a comprehensive solution, which has integrated into it the means of tackling the specific issues in due time, with the proper expertise and specialities required to fit the specific requirements of the user.

5.5.1. Communication networks

The computer networks are physically linked according to various models of connections called 'topology'. The basic topologies, shown in Figure 5.6, are the star, the hierarchical, the ring, the completely interconnected, and the irregular topology (Ceri & Pelagatti,1984). The model we are suggesting in our work would require a combination of two of these topologies. The basic one is the hierarchical topology, which will support the command (and monitoring) channeling, with the completely interconnected topology (with some reservations) to facilitate the coordination service (i.e. System Two).

The communication network enables the computers which are linked by it to perform autonomously (which is why we chose to use it). These computers are called by us Terminal Points (TP); they are also called by other literature 'hosts' or 'sites' (Date, 1981 and Ceri & Pelagatti, 1984). The word 'site' will also be used by us to refer to a geographical area which is supported by a TP belonging to the network.

The communication network itself involves links of various kinds (telephone lines, coaxial cables, satellite links, etc.), and often includes several computers which are dedicated to the networking function, and therefore their function is not the same as the TP computer, so they will not be considered here.

The basic facility which is provided by the communication network is the following: ´a process running at any site can send a message to a process running at any other site of the network´ (Ceri and Pelagatti, 1984).

Three aspects of this basic function must be considered here:

1) The ´delay´ which will be due to two components: the first is time spent by the message to reach its destination and depends essentially on the performance characteristics of the network itself; the second is when the network is heavily used and the delay becomes longer, due to the waiting which is imposed on the message when other messages have to be transmitted before it.

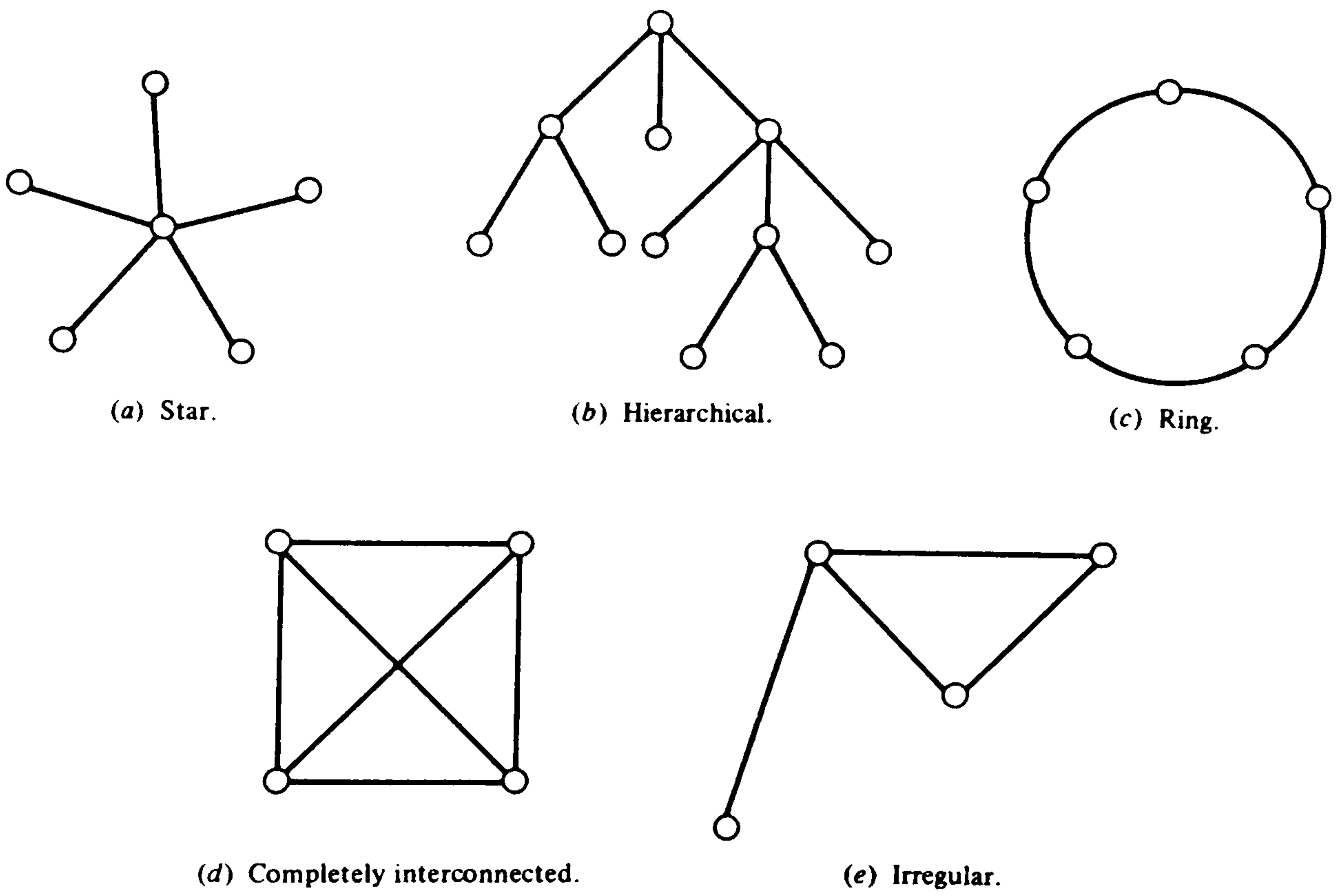
2) The ´cost´ of transmitting the message, which is incurred basically from a fixed cost associated with each message plus an additional cost which is proportional to the length of the message.

3) The ´reliability´ of the network, which is the probability that the message is correctly delivered at its destination. Failure of delivery will be due to various types of network failure which will be considered separately later.

In many applications it should be possible for one TP to send the same message to a group of TPs; this operation is called a ´broadcast´. It is possible to perform a broadcast by simply sending separately as many single messages as there are TPs in the group, thus repeating the basic message-sending operation. However, in several types of network, broadcasting the message once is much less expensive than sending the same message many times.

Communications networks may be categorized according to their physical structures and behaviour, such as the point-to-point, the common bus, and the satellite network types (Ceri and Pelagantti, 1984). Examples of these are shown in Figure 5.7.

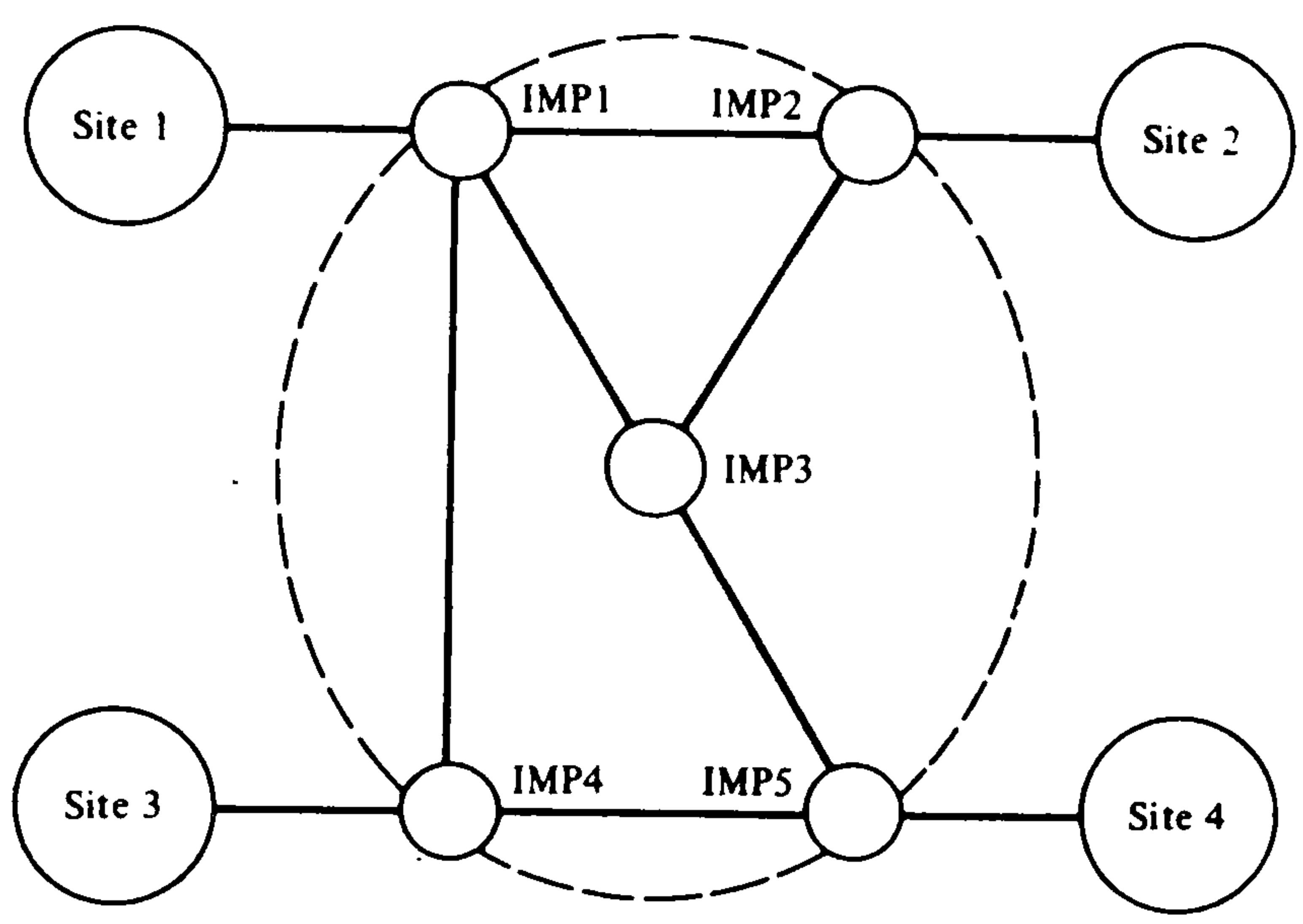
The communication network in Figure 5.7a is composed of a set of dedicated processors, called IMPs (Interface Message Processors), which are pairwise connected by a communication channel, with each TP connected to an IMP. Any communicated message is delivered and stored first in an IMP, which then sends it to another IMP, and so on, until the message finally reaches the IMP to which the destination TP is connected. In such networks, the IMP is responsible for choosing the path along which the message is transmitted. In a point-to-point network, the separation between the TPs and the IMPs tasks is a logical one, and not necessarily a physical one—the same computer could perform both functions.



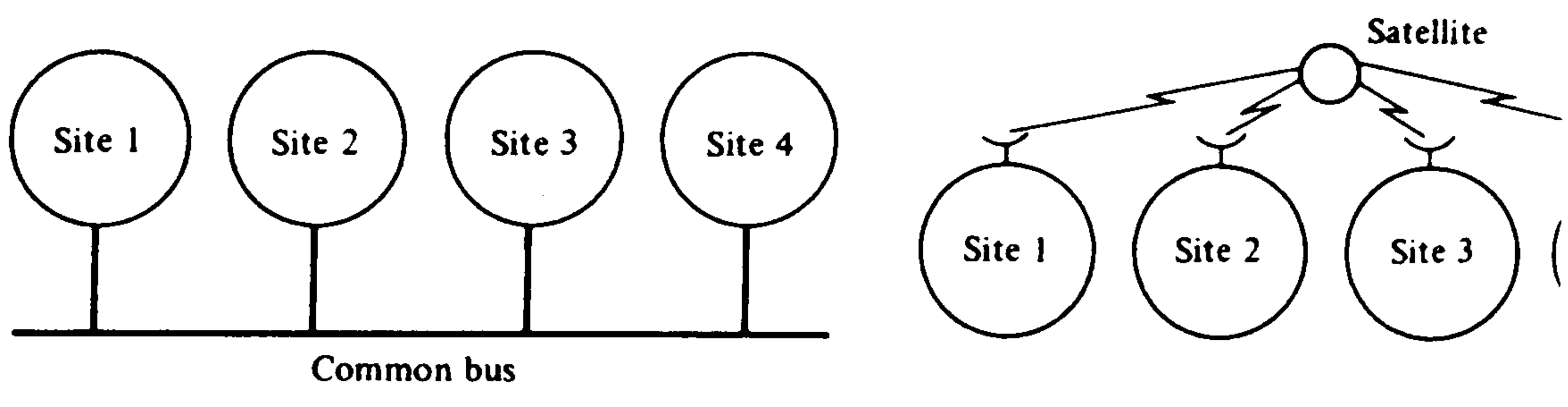
(*a*) Star. (*b*) Hierarchical. (*c*) Ring.

(*d*) Completely interconnected. (*e*) Irregular.

Figure 5.6, network topologies



(*a*) A point-to-point network.



(*b*) Two types of broadcast networks.

Figure 5.7, types of communication networks

Computer networks are also classified as 'local' and 'geographically distributed' networks. This distinction is based on the fact that the communication channels used for small distances are different from those used for long distance communication. The local networks are characterized by their short distance communication but very high throughput. They are typically of the type shown in Figure 5.7b.

As for the geographically distributed networks, which are featured mainly in our work, they are still mainly based on the telephone lines. They are characterized by very long distance communication but a very low throughput. This throughput is even lower in cases and the reliability is much less than that of the local network.

Any two processes which want to communicate with each other have to follow some rules, called 'protocol', in order to achieve this task. The protocol states how the sender and the receiver can reach an agreement on exchanging a message which would include the recognition and identification of each other, number of messages exchanged, whether an answer is required or not, and so forth. A protocol must take into consideration that as long as two TPs are exchanging information continuously it is not convenient to disconnect them. However, if one of them has to think about a problem, or has to run another process to find an answer, it may be convenient to disconnect and connect again at a later time, in order to free the resources involved here while they are not utilized. It must be noted here though, that the connection in a network may not be direct, as Figure 5.7a shows. The exchanged messages may follow different paths depending on the decision taken by the routing algorithm of the network type used.

A typical communication error is the loss of a message. When a message is sent from one TP to another, with a typical network, we would expect: the sender TP to receive an acknowledgment after a predefined time interval, the message to be correct, and the message to be received in the proper sequence with respect to other mesages exchanged between the two TPs. With respect to these considerations, a variety of possible failures might occur, but most networks are capable of eliminating most of them. Thus it would be possible to assume that, if a message is delivered at a destination TP, then it is correct and in the right sequence with respect to other messages exchanged between the TPs involved; and that, if the sender TP receives an acknowledgment, then the message has been delivered. However, if the sender TP did not receive an acknowledgment, then it cannot know whether the message has been delivered or not, for the message may have been delivered but the acknowledgment lost. The typical solution therefore, is that the same message is sent a number of times and, if the failure persists, it would be reasonable for the sender TP to assume that the destination TP has failed. (If the failure is due to the loss of the acknowledgment message, then the destination TP will become aware of this failure when it starts to receive the same message over and over again.)

The use of communication networks and database systems at the same time is marred by a serious danger to the integrity of the system's data, especially if the application involves a 'transaction' type process. Processes of the transaction type are where more than one TP is involved in the exchange (and change) of data, so that the changes at one TP must not be made unless its complementary on the other TP is also made, and vice versa, otherwise the whole process should not be committed. A failure in the communication network could happen while TPs involved in a transaction are still exchanging information (i.e. the transaction is not completed). Nevertheless, there are rigorous measures that can be implemented in the system to ensure that such processes are only committed the correct way (detailed examples can be found in Ceri and Pelagatti, 1984).

5.5.2. Databases

There are basically two models of database, relational and hierarchical, regardless of their nature whether centralized or distributed. As far as our work is concerned, our review will be biased towards the relational database model, since it is the most suitable model for the distributed data concept which we are promoting.

In relational databases, the logical image of the stored data is treated as a table (i.e. relations) called ´file´ by us, which has a number of rows called ´records´, and a number of columns called ´fields´. The names used here are taken from traditional data-processing language and have an approximate correspondence to the mathematical terms mostly used in the literature. The major features of relational files which distinguish them from the traditional (undisciplined) files (Date, 1981) are:

1) Each file contains only one record type.

2) Every record in a given file has the same number of fields.

3) Each record has a unique identifier (key).

4) Within a file, the records are stored either in a random order, or stored according to the value of one (or more) of its keys.

An important property of some of the fields of records is that they serve as keys for their records to identify them within the database. The key field must hold a unique value within the file, such as the ´supplier number´, thus the ´age´ of the supplier cannot serve as a key. If however, the database identifies the suppliers within their countries, and hence numbers them separately within their geographical areas, then a pair of fields must be used (country code and supplier number) as a key. We can also use keys on different levels of indexing so that we may have a ´primary´ key and a ´secondary´ key with the same file.

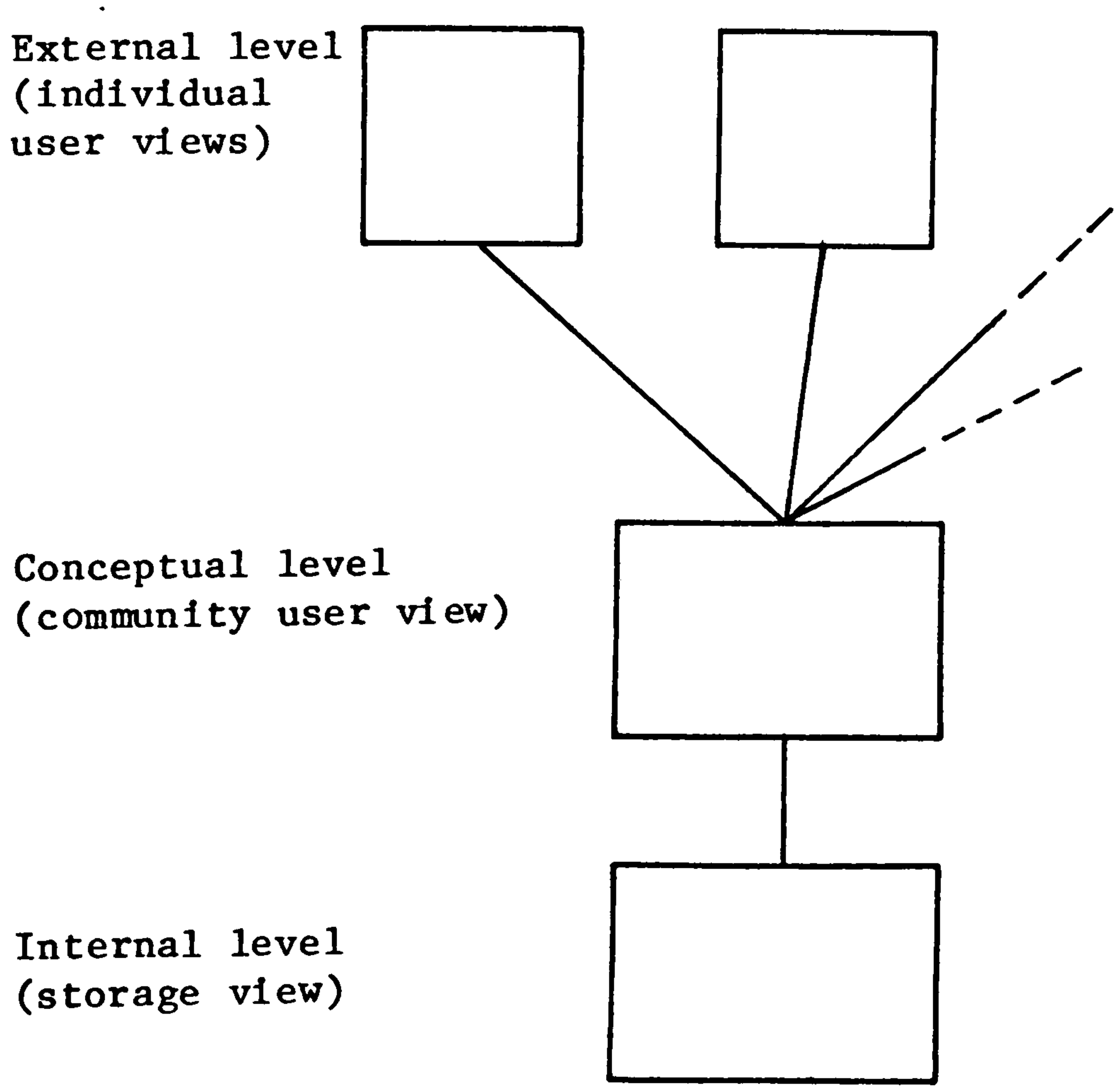


Figure 5.8, the three levels of the architecture

The database architectures follow a variety of different frameworks, but Figure 5.8 shows an architecture that seems to fit a large number of systems reasonably well (Date, 1981) and which gives a good idea about the general use of databases.

The architecture we are describing here consists of three levels of interfacing with the database. This is necessary to facilitate the broad requirements of the various users of the database (which is an essential property of the database). Each level will hold its own view of the stored data. The internal level is the one closest to the physical storage of the data and is concerned with the way in which the data is actually stored. The external level is the furthest from the physical storage and the closest to the user, and is therefore concerned with the way in which the data is viewed by the individual user. The third level, the conceptual level, lies between the previous two and, as would be expected, is concerned with reconciling the two views to enable the individual user, with his own view of the data, to access the physical storage.

Each user is able to access the database with an external programming language (which has to be supported by the database system) like COBOL, PL/I, FORTRAN, etc.. Each user may access certain fields while others may access the same, or different, fields in the database while using the declarations designed for the programming languages they are using. At the conceptual level there will be a reference to the data requested by all the users, such as the record name and all the fields declared by the users. The user may use different names (labels) in his declarations from those used in the conceptual level but he cannot use different lengths than those actually assigned to the data types (i.e. record or field) in the physical storage level (i.e. internal). For example, the users may use the labels EMP# or EMPNO while the conceptual level uses yet another name like EMPLOYER_NUMBER. At the internal level the same data type is stored, again with a different name possible, but this time it is declared in more detail, such as the ´offset´ of the field with respect to the record, the length of the data type in bytes, the index field if the file is indexed, and so on.

The transition from the user (external) level to the conceptual level, and from the conceptual level to the internal level, is helped by means of ´mappings´ on two levels: one between the external and the conceptual levels, and the other between the conceptual and internal levels. The conceptual/internal mapping defines the correspondence between the conceptual view and the stored data; it specifies how conceptual records and fields map on to their stored counter parts. If the structure of the stored database (i.e. storage structure definition) is changed, the conceptual/internal mapping must be changed accordingly, so that the conceptual schema remains invariant. Similarly, we have the external/conceptual mapping to define the correspondence between a particular external view and the conceptual view.

In addition to supporting the conventional programming languages used by the users, the database has its own language. It can be either a query language or a special purpose language tailored to the user's requirements and supported by an on-line application program. It may also have a special command language for the benefit of the user, to write his own applications.

The user's languages which are supported by the database system will include a 'data sublanguage' (DSL), which is a subset of the total language that is concerned with the database objects and operations. It is a combination of two languages: a data definition language (DDL), which provides for the definition or the description of the database objects (as they are perceived by the user), and a data manipulation language (DML), which supports the manipulation or the processing of such objects.

The final aspect of the architecture, which we should consider here, is the three components which are central in handling the actions of the database, namely, the database management system (DBMS), the database administrator (DBA), and the user interface. The DBMS is the software which handles all access to the database. Conceptually, what happens is the following:

a) a user issues an access request, using some particular data manipulation language;

b) the DBMS intercepts the request and interprets it;

c) the DBMS inspects, in turn, the external schema, the external/conceptual mapping, the conceptual schema, the conceptual/internal mapping, and the storage structure definition; and

d) the DBMS performs the necessary operations on the stored data.

The second component, the DBA, is the person (or group of persons) responsible for the overall control of the database system. The DBA's responsibilities include the following:

a) deciding the information content of the database;

b) deciding the storage structure and access strategy;

c) liaising with the users;

d) defining authorization checks and validation procedures;

e) defining a strategy for back-up and recovery;

f) monitoring performance and responding to changes in requirements.

Any practical database system would provide the DBA with utility programs (special system-supplied applications), such as loading, recovery, and statistical analysis routines.

The third component, the user interface, is at the external level and may be defined as a boundary in the system below which everything is invisible to the user.

Typical failures in distributed database systems are of two distinct types: local failures and global failures. Failures of the global type are due to failures in the communication between the sites (see Subsection 5.5.1). The local failures are those typically associated with centralized database systems, and their most important characteristic is the amount of information lost because of the failure. These failures could happen without any loss, with loss of volatile information, or with loss of nonvolatile information, or with loss of stable storage.

A failure could happen without any loss of information and that is when all the information in the main memory is available for recovery. The loss of volatile information happens when the failure results in a loss of the content of main memory, such as in a system crash, while all the information involved is safe on disks. But when the contents of the disks are lost (due to media failure as in the head crash) then we have a loss of nonvolatile information.

There are some measures which can be taken against the loss of nonvolatile information. The information stored on one disk can be protected by replicating it on several disks with independent failure modes and using the so-called 'careful replacement strategy' (Ceri and Pelagatti, 1984): 'at every update operation, first one copy of the information is updated, then the correctness of the update is verified, and finally the second copy is updated'. This strategy leads to what is called a 'stable storage'. A failure may also happen with loss of stable storage because of several, simultaneous failures with loss of nonvolatile information. However, the probability of such failure can be reduced by increasing the number of replica disks used, but it cannot be reduced to zero.

To recover from such failures, the idea of a 'log' is used, which contains information for undoing and redoing all actions which are performed by transactions. There are recovery procedures which will read this 'log' and perform certain operations to recover the lost information. Whenever the database is updated, a log record is written in the log file. The log record will contain the necessary information for recovery, such as an identifier of the transaction, the type of action performed, the old value and the new value, and an auxiliary information for the recovery procedure (such as a pointer to the previous log record of the same transaction). Some security measures are also needed here, for it is possible, if the log record is written after the update has been written, that the failure could occur in between the two operations, and we would end up with no log record to perform the recovery procedure. To avoid this, the log record can be written in two stages: first, the ´undo´ portion of the record is written before performing the update; and second, the rest of the record is written before committing the transaction; this is called ´log write-ahead protocol´.

In our work, we promoted the use of database systems to support our applications for several reasons which are advantageous when it comes to the large scale handling of information. Some of these advantages are reviewed here:

1) The redundancy of stored data can be reduced (when necessary) and the waste in storage can be eliminated. By contrast, in the non-database systems, each application has its own private files, which can often lead to considerable redundancy in stored data.

2) The inconsistency can be avoided, as a result of the preceding point, there will not be several copies of the same record to worry about every time we update one of them. Even in the event of requiring the existence of these copies (i.e. when redundancy is not removed), then it would be easier with database systems to propagate that update to cover all the records concerned.

3) Probably the strongest advantage, from the informational point of view, is that the data can be

shared, not only by the existing applications, but also by new applications which can be developed to operate against that same stored data. This means that the data requirements of the new applications may be satisfied without having to create any new files.

4) We can ensure that all sorts of standards designed by the user are followed in the representation of the data in the database. Standarizing stored data formats is particularly desirable as an aid to ´data interchange´ or migration between systems.

5) Security restrictions can be applied by ensuring that the only means of access to the database is through the proper channels, so that checks can be carried out whenever access to sensitive data is attempted.

6) Integrity of the data can be maintained with the use of databases. Even after removing redundancy (Point 1) and eliminating inconsistency between replicated records (Point 2), the database can still contain incorrect data, due to errors of storing values which are outside the natural limits of their fields, or even a non-existing value. The database can help to avoid these situations, in so far as they can be avoided, by carrying out validation procedures whenever any update is attempted.

7) The database system can be structured to provide an overall service for the corporate requirements of the

user.

The advantages we have just reviewed are in fact drawn from the practice of the centralized database. The distributed database structure has, in addition to the preceeding advantages, a few of its own, which we will be discussing shortly.

Distributed database technology is a comparatively recent development and is still in the prototype stages. A distributed database is (Date, 1981 and Ceri & Pelagatti, 1984), a database that is not stored in its entirety at a single physical location (as is the case with the centralized database), but rather it is spread across a network of computers that are geographically dispersed and connected via communication links. The data is stored at a location at which it is frequently used but is still available, via the network, to users at other locations. Each site of the network has autonomous processing capability and can perform local applications. Each site also participate in the excution of at least one global application, which requires accessing data at several sites using communication subsystem.

It is clear from this description that there are different software considerations involved here (as compared with centralized databases). With the distributed structure, the applications of the system are essentially of two types:

one type is a ´local´ application, the other is a ´global´ application. In the case of the local application there may be no major departure from that of the centralized structure. (As a matter of fact most available distributed database systems are using existing centralized database systems.) With the global application, the user would need to access data at more than one location of the network, and the ideal situation here is that the user does not need to know where any particular piece of data is physically stored (as if the user is running a local application). This is in fact the key objective of the distributed database approach and is the source of the chief technological problems facing the designers of such systems.

The classification of the applications as local and global suggests also that we can have a local autonomous processing capability, in addition to the organizational autonomy, to support the local processing needs.

Acquiring such systems commercially is however not easy at the present. One commercially available system is a homogemeous distributed database management system, called ENCOMPASS, which is built upon Tandem´s NonStop computer architucture and its operating system. Based on this computer architecture and operating system, the distributed database management system provides data distribution, query processing, and distributed transaction management. The system however is a mainframe and we have to exclude it from our considerations. On the other extreme is an IBM facility. IBM has developed and is developing tools which facilitate building distributed database systems on top of homogeneous and hetrogeneous local systems. The most important of these tools are the System Network Architecture (SNA), and the Inter System Communication (ISC). This system offers the best hope for us since it is designed to handle end-user terminals. The IBM micros are now capable of linking to IBM mainframes as terminals.

IBM however has developed system R* as a prototype distributed database system. The most important objective of R* is to provide site autonomy. This is achieved when the site is able both to control accesses from other sites to its own data and to manipulate its data without being conditioned by any other site. Many of its features are now operational, including data definition and manipulation statements of SQL, transaction management, deadlock detection, and recovery mechanisms.

We have already demonstrated in the discussion of the previous chapter why we choose (by implication) the distributed database and not the centralized concept. To recapitulate, there were organizational considerations which resulted in rather autonomous subsystems for which the distributed data concept seems to fit naturally. There were also economic considerations. These would come from the reduction of communication overheads, as a result of using the local databases most of the time by means of the local processors (which gives us at the same time a lower rate of communication failures). Gains would also come from the benefit of recent developments in computer technology (micros and minis ranges) that would eliminate the need for larger computer centres with multiprocessing capabilities. This would give us the benefit of using single processors resulting in higher freedom, better performance, and fewer technical complications.

With the distributed data concept we can continue using ´existing´ databases employed by the organization (as locals) and create the distributed database bottom-up from them. By the same token, the incremental growth of the organization can be easily and smoothly supported, whereas with the centralized data structure it would need a major reconstruction if its initial dimensions did not consider such growth (which is an impractical, if not impossible, task to achieve).

The fact which must be stated here is that the failures can be more frequent with the distributed database because of the greater number of components, but the compensation is that these failures will be confined to those applications which use the data of the failed TP (site) only, thus a ´complete´ system crash is rare.

The last comment, about the benefits of using databases, is that the trend of software houses today is to develop complete specialized databases with full data stored in them regarding the choosen subject, in addition to empty databases. This is especially important to those who did not have the chance to accumulate such information.

VI. THE SOFTWARE SPECIALLY DEVELOPED FOR THE SUGGESTED ORGANIZATION

6.4.2.3. Updating the quantity received

6.4.2.4. Updating the quantity ordered

6.4.2.5. Reporting

6.1. About the Demonstration

By the end of the last chapter the proposals should be complete and ready to be implemented. The only remaining ingredient is the software required to make the system fully functional. To demonstrate the workability of the proposed solution needs, however, at best a full scale software preparation. This is necessary if the users do not happen to be cyberneticians. They need to observe the functioning of the system when all the tasks involved are working, and interacting together, before they accept it as a valid alternative. To achieve this would certainly mean a longer time to be spent on this work and a number of programmers in addition to the necessary hardware. Meanwhile, it could be a valid argument to say that, as the whole work is built on principles whose initial development had stemmed from the aim to procedurize the events they are governing, then writing the software for these procedures must be a straightforward task. However, we have at least to attempt what is possible, which is still of paramount importance in demonstrating a major aspect of the proposals. We have been proposing that the user may abandon current dependence on mainframe computers in favour of the microcomputer. The capability of the micro in handling the various tasks involved in the operations of the System must be therefore demonstrated. Even though this is possible, up to a point, which can be seen from the next sections of this chapter, the ambitions of eliminating human corruption and reducing his inefficiency cannot be fully demonstrated for two main reasons: (1) writing the software that would demonstrate these things is a highly specialized and technical task that depends fully on the operating system used, and (2) to write this software, the problem has to be a real one so that its components (psychological, sociological, and technical) can be studied in detail prior to writing the procedure, otherwise the whole thing may be dismissed as a pure speculation.

For the purpose of this demonstration, three distinctly different jobs which are involved in the System are outlined in the next sections of this chapter. This outline is concentrating on the file structure of the data used to: (1) give an idea about the storage requirements, (2) show how sophisticated the application can be and whether this sophistication can be increased, and (3) examine the updating moves which are required to preserve the integrity of the System's data. The 'inventory control' job was then chosen for full demonstration for various obvious reasons, chief among them is the possibility of doing it without the need for a communication network (which is not available). At the local level, a database system (dBASEII) is used to handle the jobs involved. Another program, an 8080 assembly language program, is to be used by each TP node of the System for the updating of its own data.

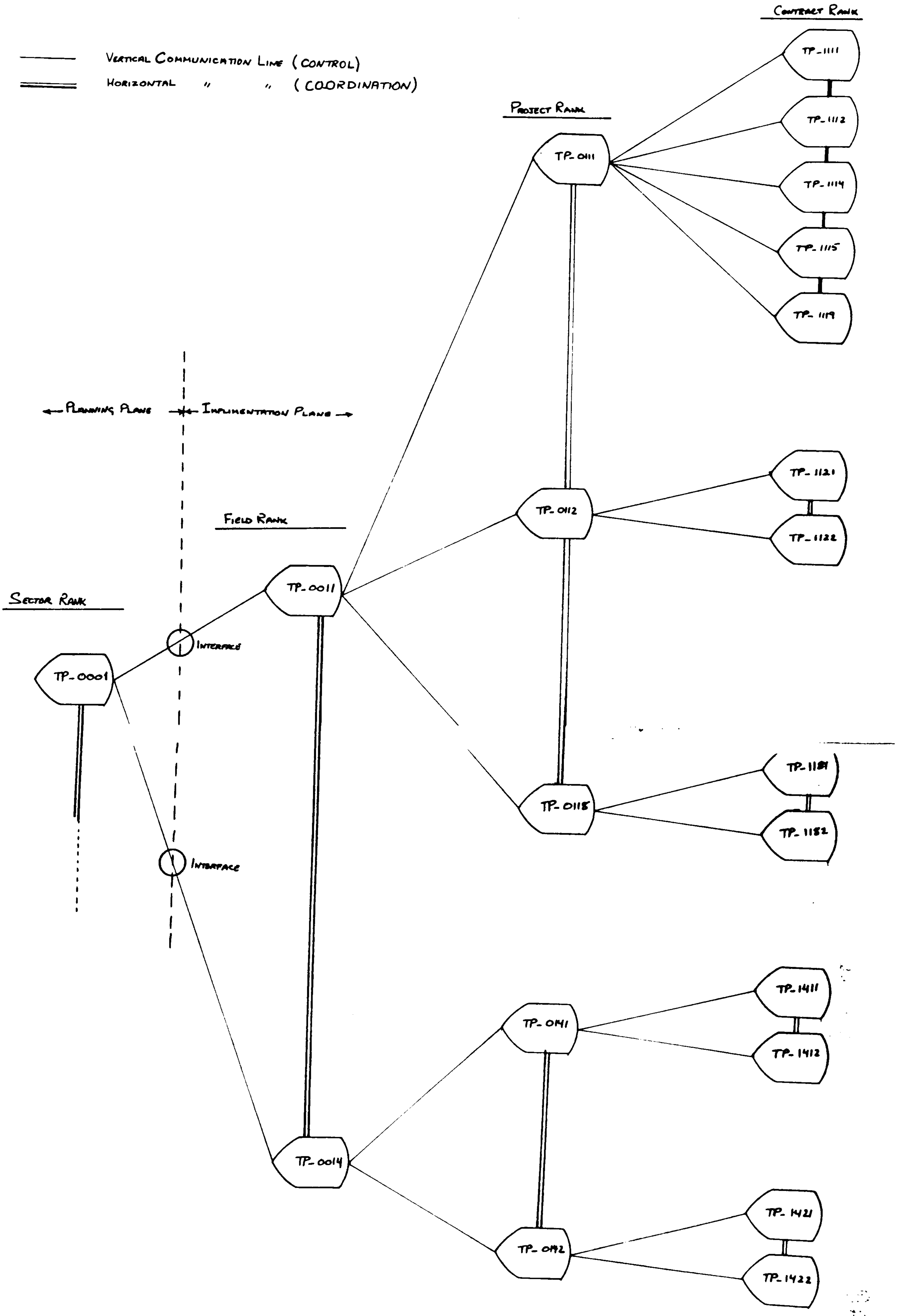


Figure 6. , the demonstration's sites and their links

This program can be looked at in a different light— it is affecting the flow of information in the down-up direction through the ranks of the System. It should demonstrate: (1) the sophistication of the micro's programming, (2) the possibility of using the same program at each node throughout the System regardless of its rank, and (3) all the other typical traits of the micro, and computer in general. The two programs together demonstrate the natural proposal of using the spread data approach, with one of these programs playing the role of local processing facility and the other as a global application.

We chose one section, which is a minimum representative sample, of the whole System's connections that can be demonstrated on one micro, as shown in Figure 6.1. In this section we consider a complete vertical line going through four ranks to examine a continuous line of communication between Contracts in the fourth rank and the Sector they belong to in the first rank, passing as it were through the third and the second rank's node subsystems in the communication grid of the chosen Sector. The horizontal links of this section are also examined by considering the various possiblities of configuration, eg. gaps in the coding numbers of subsystems, that are sufficiently representing the accumulation of information within that vertical line. Figure 6.1 shows a diagram of the subsystems considered in the demonstration, each with its TP code that will be used as a code part in the filename of each data file residing at the TP. The dotted line on the left of the diagram represents the interface between the two planes mentioned in Subsections 4.4.2 and 5.4. At the interface point, the communication is shown to go into two directions: one to the Sector and the other to the implementing Ministries, this however is not demonstrated by our programs.

The updating (i.e. information and data maintenance) in our system is required on two levels: Local and Global, in the sense of subsystem and parent system, propagating in the recursive manner which is characteristic of the System. Thus we have, at the local level, updating activities that have two aims: one is to maintain the local information for internal use and the second is to maintain the local information so that the global information may be maintained.

This fact should be emphasised here to show that the main responsibility for the integrity of the data will lie with the local subsystems. This is the one drawback of the decentralization of information gathering which can be balanced, to a degree, by imposing some local constraints in the form of control measures that would bar any illegal updates from taking place. In this context, and in addition to those control measures, a firmer method of monitoring should be employed in the form of random checks done manually (whenever the technology cannot do it for us) in addition to the traditional method of the yearly stock taking.

The updating processes in our example are carried out by two programs: one is an assembly language program, the other is a database program. A detailed description of each is found in Subsections 6.4.1 and 6.4.2 respectively. The first program, the assembly language program, is assigned global updates which are routine and determined beforehand and do not require the constant intervention of the operator. The second program, the database program, is assigned the local updates which require a constant intervention by the operator. This is, therefore, a conversational program taking care of the daily (or on-line) updates required to restore the data of the database to their real values after a day's work, and also extracting information messages, regarding the state of affairs, which are relevant to the operating of each task involved in the system.

The reason for writing the first program in assembler language is to make it more isolated from the operator, who has no decision with regard to the routines carried out by it. There are also some technical reasons, eg. to manage the global updates, neccessating the direct manipulation of the data files, is done more easily by this language.

The recursive nature of the System makes it possible for the programs to be written for each job, rather than for each subsystem, resulting in a uniformity of operations, documentations, and expertise that is not readily achieved if the programming is done for each subsystem. This method means uniform file layouts throughout the System, which is an essential requirement if the information is to flow freely across the System's internal boundaries, which is in turn an essential requirement for linking the subsystems to each other (according to the model) to produce the synergy and oneness envisaged for the System.

The various jobs involved in the operations of the example system are varying in complexity; and it is suggested here that a combination of Machine-Human can match this complexity effectively if the programmer of the machine takes into consideration that there will be a human brain (in the sense of a high level reasoning and decision-making power) working for him as an extension to the machine--hence the conversational nature of some of the programs involved. This allows for a very cheap and powerful extension to the processing and logical powers of the existing personal computer when used in that machine-human combination.

What makes the human the extension to the machine and not the the other way around, as is normally considered the case, is the fact that as long as the operator decides to stay within the environment of the pre-written programs, the machine will lead the conversation and will branch only to and from those branches allowed by its program.

6.2. Outline of Some Jobs Involved

In Section 5.2, we referred to a group of information items, which were involved in what we called a ´second stage´. In this section, we will consider them separately and try to outline, as much as possible, their file structures, together with the major processes (updating moves) associated with them. There will be a variety of dataprocessing activities associated with them, but we will mainly consider the updating moves since they reflect the two major aspects which concern us most, namely changing the stored data and preserving its integrity. A common characteristic of all the data involved is that they are stored in the form of tables. The table´s name (label) starts with enough characters to indicate the nature of its data (i.e. INV, SHIP, EQP, etc.) followed by the TPID at which the table is stored to identify the owner.

6.2.1. Inventory control

All the material used by the system will be physically stored either at the site which uses it or at a depot, belonging to a higher rank subsystem, to be distributed, basically, to its subordinate subsystems. The point here is that there will be a detailed data record, for each of these items at the location of their physical storage, which is stored at the TP of that location to form the inventory file used there; the inventory table structure is shown in Appendix C.

The inventory files may, therefore, be found in their detailed form at each contract TP as user sites, or at a higher rank's TP if and when it is responsible for a depot.

The TP at which location the material is physically stored is directly responsible for maintaining the inventory data in a semi-online manner (i.e. during the working day). A summary of this data, according to each item, is stored at each node TP, which is taken from the data of their branch TPs. Updating the inventory data at the node TPs is done indirectly and periodically (say every week). At the end of the set period, a summary of the data at each branch TP is passed up to the node TP starting from the lowest rank's TP (at which the data is maintained directly).

The major updating moves regarding the inventory control are concerned with maintaining the integrity of the stored data to reflect, as accurately as possible, the state of the items involved. This is an essential condition for getting the right information about the inventory status. To get this right information at the right time is also important and can be done with a stroke of a key. The major updating moves investigated here are required whenever a quantity of a particular item is:

1) Received.

2) Used (consumed).

3) Ordered.

4) Moved (not consumed).

The first two are updates of the transaction type, while the other two are a simple update process (changing one field of data only).

The design of the database table used here can be extended to include more data types, or less, depending on the degree of complexity chosen for the subsystem. In our case, we chose a design which would reflect the exact quantity ´available´ of each item, plus those expected to be delivered at the same location (the time of delivery can be worked out from the shipment table of Subsection 6.2.3). From this same table we can extract other auxiliary information, such as the quantity used todate from each item. Storing the projected quantity to be used for the whole contract (or project) would also help in giving information about the rate and efficiency of the work progress when compared with the used (up-till-now) quantity.

The fourth updating move is not simulated in the programs of Section 6.4 in order not to complicate the simulation programs. Updating of the ´moved´ quantity is similar to that of the used quantity except that this quantity is not added to the ´used´ quantity field.

In conjunction with the inventory table, there will be used some other tables, such as the SHIPMENT, PARTS, and

SUPPLIERS tables, whose items can be traced via a common field, from one table to another, to build up the total information available about the traced item. Other auxiliary tables would be needed as well when deciding on the specific issues and requirements (details in Section 6.4).

6.2.2. Equipment control

This is a different case in nature from that of the inventory, in that the used equipment is normally not consumed by the site; thus a provision must be made for the return of the equipment after finishing with it. The physical presence of the equipment and its data storage is exactly like that of the inventory (i.e. either at a depot or a site). The envisaged data table for equipment control is shown in Table 2.

EQP:NO	ACT:LOC	ORD:LOC	EQP:MK	EQP:USG	SEC	INUSE?	STATUS

Table 2, EQP: table

Our major concern, as can be seen from Table 2, is to retrieve the essential items of information about each item of equipment, such as those identifying it (registration number, make, usage), its essential specifications (those of relevance to the user), its status, its location, and the essential dates on which it is in use or not. In the table, we have two different location fields, one for the present location at which that particular equipment is found (ACT:LOC), and one to which it is ordered, by the node TP, to be moved.

The major updating moves will be confined to the location, the INUSE?, and the status fields. The INUSE? field is supposed to reflect the dates from which the equipment will be in use until it is not, or it is not presently used but required in the future, or referring the user to the STATUS field if and when the equipment state is neither of the previous two. We can adopt three letters to precede the dates used in the INUSE? field to reflect each of the three previous situations (as one alternative of design). The letter ´R´ could stand for ´required until´ the projected date shown after it. The letter ´N´ could mean a ´not in use until´ the projected date shown. If the equipment concerned is not being used for the time being, or it is not sitting idle until a future date on which it is to be used, then it is of no benefit for the enquirer who may want to borrow it. In which case the letter ´S´ would inform him so, and at the same time would refer those who want to know more, to the STATUS field. The situations that the STATUS field reflects may be: in a garage for repair, on a transfer to another location, borrowed for a duration by another site, etc..

The letters used in the INUSE? field were chosen in an ascending order of their ASCII values to increase the variety of the field itself (S>R>N). This is to allow for a simple manipulation to gain more information, so that the search for free equipment by an indicated date will be done simply by searching for values which are ´less´ than ´R´ followed by that date. The item, which would return a true value, will be all those required to be used by their present users until an earlier date than the one indicated by the search routine, plus all those starting with the letter ´N´ (i.e. those not required at the present time). The next step is to exclude all those which are not required until an anticipated date on which the enquirer needs that equipment (by a simple ´greater´ than ´N´ followed by that date). Similarly, if the value in that field is ´S´, then no matter how far ahead the date is, the return is false. To consider an example of information retrieval, suppose a user is searching the database for a particular item of equipment (say a crane), which he needs on the 22nd of March 1985 and until the 30th of August 1985. This means that the search would be done for any ´crane´ which is not presently in use and until 30/08/85 (i.e. any value which is greater than N850830), and not required by the present user site after 22/04/1985 (i.e. any value less than R850422). A sample database program may be:

```
SELECT    EQUIP        (selecting the equipment file)
WHERE     EQP:USG=CRANE
```

AND   INUSE?<R850422 (from 22/04/85)

AND   INUSE?>N850830 (until 30/08/85)

In plain English, this is to say: select from the table whose name is EQUIP, an item whose usage is CRANE, which will be free to be borrowed from 22/04/85 until 30/08/85. The user will get a list of all the equipment whose INUSE? field's value is not 'S', less than R850422, but greater than N850830, from which he can choose the one which is most convenient by making an application to his node TP to order the move.

One essential difference between the equipment record and that of the inventory is that, while the inventory record is unique for the site, the equipment record may have a replica at another site. This could happen if and when the design requirements (of the system) demand that the node TP stores exactly the same image of this record for the purpose of its own control. However, we can avoid the typical updating difficulties in such cases, by storing the first part of the record only (i.e. the INUSE? and STATUS fields excluded). In this case, the first part would give reasonable information and at the same time supply the information which would lead to the detailed record at the site TP via the ACT:LOC and EQP:NO combination field, if further information is needed. At the node TP there remain only those updates which are essentially its own responsibility (i.e. the locations´ updates), and are at any rate less global updates.

The location fields are also involved in another way in the global applications. When a particular equipment is finished with at one site, it has to be moved to the new location indicated by the ORD:LOC (ordered location) field. This equipment will remain the responsibility of the old site until it is received by the new site. A record of this equipment has to be stored at this site at the time the equipment was ordered to it, of which the first part is an image of the first part of the record at the old site´s TP, but the second part holds the new site´s own information. On receipt of the equipment by the new site, the steps that should take place are:

1)Change ACT:LOC field of the new site´s record to contain the new site´s TPID and erase it from the OLD:LOC field.

2)The same changes should be done to the location fields at the node´s TP record.

3)Erasing the old record from the old site´s TP.

These three steps must be carried simultaneously for, in addition to updating the data to conform with the new reality, these steps would act as a formal confirmation of the receipt of the equipment by the new user (i.e. a transfer of responsibility from one site to another). In such an arrangement the old site´s record should not be erased from the equipment table (at the old site) unless it is officially received by the new site. This would demand two measures: the first is a global program which will be activated by the new site´s TP to execute the three steps mentioned previously; the second is to prevent the old site´s user from erasing the old record individually, which he may do by using the local database DELETE command. We can prevent this by modifing the local DELETE command to first check a particular byte at the beginning of the data file, to see whether it is an equipment file or not. If it is an equipment file, it can be made to refuse excution of the instruction.

6.2.3. Shipment control

This subject relates strongly to the inventory control tasks, in addition to its own outstanding importance, since it provides information regarding the ordered material. The envisaged data table for the shipment control is shown in Table 3.

SHIP:INFO	OWNER	PRT:NO	SUPP:NO	DUE:DT	REQRD:BY	

Table 3, SHIP: table

Working out a schedule would require information regarding the availability of material to be used in the scheduled job. The inventory table can provide the planner with such information as we have seen previously. If, however, the available quantity of material required is not sufficient, but a sufficient ordered quantity is earmarked for the site, it would be possible to work out that schedule. There would only remain some uncertainty about the availability of the ordered quantity at the time it is scheduled to be used. Such information can be extracted from the shipment table, and the projected delivery can be worked out from the DUE:DT (due data) field, which is a fixed date set by the supplier. The RQURD:DT (required date) field is a date set by the user, initially when the detailed plan is decided and the first estimate of quantity required has became known. However, this date may be changed by the user, forward or backward, as the circumstances dictate. This field would work for the benefit of the user, indirectly as well, for example if this table was available to the port authority they could then speed up the handling of the shipment if and when the RQURD:DT is very close. The existing practice is to give priority (by the government) to the National Plan projects, thus in consequence the handling of material due for these projects would enjoy the same priority (there would be some inner priorities as well which are also known to the port authority). The mutual benefit from this table between the site and the port can be extended by adding other fields to hold the estimated unloading time and the port of arrival, to be maintained by the port and used by the site.

It is apparent from the use of this table that, unlike the previous tables, the shipment record is needed in its same image in at least two different locations; a higher rank (i.e. at a node TP) may also store a replica of this table. Any update to this table, although done locally, must be propagated to every replica record existing in the system, thus putting it firmly in the global applications category

The other fields in this table would also supply the user with relevant information, such as those of the PRT:NO and the SUPP:NO fields, which would serve as pointers to the detailed records of information regarding the shipped material and its supplier. These detailed records are stored in reference tables maintained by the system, namely the PRTS and SUPP tables, as shown in Tables 4 and 5.

SUPP:NO	ADDRESS	STATUS	SPECIALITY

Table 4, PRTS: table

PRT:NO	SPECIFICATION	SUPP:NOs	

Table 5, SUPP: table

There are other important uses for these tables, especially when it comes to deciding on orders and tenders, when they would serve as a quick reference. There would be no local updating involved with these tables, and the data types are regarded as fixed unless the system intervenes (from the highest authorized rank) to change them.

6.3. Human factor in Software Design

What has preceded gives rise to the importance of the conversation itself (as the inner mechanism of the machine-human combination) and the way it will be conducted to make the optimum use of the human power. As the machine side of this conversation is robust in its behaviour, and does not have the human's flexibility, while at the same time the burden of leading the conversation lies heavily on it, then the clarity of purpose behind the machine's prompts during that conversation becomes a decisive factor in reaching a successful conclusion to the conversation.

Another decisive factor is the human decision that is made in response to the machine prompts.

6.3.1. The clarity of purpose factor

The main burden of the clarity of purpose lies with the machine for the simple reason it is the leader of the conversation. Its prompts are the main means of communication with the human side (besides the operating manuals), and it is upon these prompts that the operator is going to act, and therefore supply inputs to the machine, on which it will decide the next step of the process. Any misunderstanding or misinterpretation of the prompts from the machine (and it is possible that the prompt may mean nothing at all to the operator) will definitely lead to the wrong results and may also mean a disaster for the integrity of the database.

The first consideration, and the simplest in this case, must be given to the contents of the prompt, and is twofold:

1)Clear language sentences reflecting the true meaning of the prompt are often sacrificed by condensation because of the obsession of the programmer with saving space--it should be recognised here that the clarity of purpose is of no less importance than an IF statement or a DO loop in the program.

2)Explicitly or implicitly, by the contents of the prompt, all the possible reactions allowed to the operator on that prompt should be understood, together with their effects. The prompt should therefore be of a closed- and not open-type question.

As a consequence of implementing those two measures it would be possible to guarantee correct progress through the program and trap any illegal requests by the operator. It is therefore, necessary to inform the operator about the error he has just made in order to take a corrective measure. This would also be done through the machine prompts, to which the same considerations apply.

The second consideration should be given to the operator's sense of position within the operation being run, because there is normally a multi-step procedure to perform a particuler task, whose increase in complexity is accompanied by yet more complicated steps (sense of position: the appreciation of the relation of the present position, as land marked by the current prompt, to the starting point and the ending point of the running procedure). This issue is in fact a very important one for all people communicating with the machine, whether experienced or inexperienced (experience is taken here to relate to running the program concerned or to the profession in which the machine is helping). Three areas to which we must pay attention are drawn from the natural behaviour of the human who will operate the machine. In almost any process in which the human is involved, he needs to have an idea at the beginning of the process what to expect from it, then he should be able to follow the process through continuously and, finally, everything should seem logical to him. Otherwise he would turn into a machine himself, whereby we would lose the benefit of his reasoning powers, and as a result destroy the Machine-Human combination. Another effect is the inhibition of learning. These three areas combined would, if treated properly, provide the operator with the sense of position required, thereby reducing a great deal of the uncertainty—— the 'friendliness' of the system would contribute a great deal to solving this problem. Systems like ours, in general, must be prepared to communicate with the two extreme types of operators (with respect to experience), namely the new and the experienced, and those who fall in between will follow. For the new operator, the sign-on and error messages, plus the help messages, are especially important to get him started with the system, and respond in the correct way to its prompts. They are also of benefit to the highly experienced operator, especially the error messages; and there is no danger of destraction from the sign-on messages since they appear only once at the beginning of the run while the help messages are displayed on request only.

To follow the process through continuously there is the continuous conversation between the operator and the machine, which at the same time will emit the logic of the process.

To ensure the transparency of the procedures involved, for the purpose of learning and for them to be logical, a simple but basic rule should always be observed by the programmers of the system. They should ask the operator (via the prompt) for particulars of the problem, and when and where the program is going to process each of them, so that the operator will view single steps presented in a logical sequence, where it would be possible for him to infer the procedure that is being performed by the machine. If this is done efficiently, then the experienced operator will know where he is and, at the same time, the procedure would be transparent for the new operator to learn from it about the mechanics of the system.

Taking care of these considerations, on the other hand, means a constant stream of messages bombarding the operator after each move he makes. All these friendly messages, error and recovery messages would soon become a disturbing factor to the operator and not as helpful as at the beginning—a situation that might produce a negative effect. This danger could be effectively reduced if the following points are observed:

1)Minimum use of these messages, especially those which appeared earlier in the run, or those which become logically not necessary at the point in time considered. An example is when a TRY AGAIN...(Y/N)? message appears while control is inside the subprogram shell, it is not necessary to return to the main program for the main menu, or even to the menu of the same subprogram, unless the error that prompted the message is related to them.

2)By avoiding using the same type of inputs for successive quizzing messages, using alternate input types of alpha and numeric inputs, thereby trapping any omission by the operator in the error trap, a situation which is easier to recover from than entering an unknown program area.

NOTE: The Use of Menus

The use of menus is very natural to the human (and therefore, friendly) and at the same time a very good programming technique for the machine, but it is very inefficient to include the quitting option as one of its entries. This is, firstly, because it is incompatible with the nature of the menu as a contents table, while QUITting is a terminating decision (although for the programmer and the experienced operator it is logical enough since QUIT is still a procedure just like the other entries in the menu) that should be compatible with the STARTting procedure; preferably the same opening message like DO YOU WISH TO CONTINUE...(Y/N)? could be used for proceding and terminating the run. The second reason (a practical one) is, if the operator chooses the QUIT entry in error, then the recovery might be very costly. This type of error may happen very easily, especially when normally each entry of the menu is given a ´number´ as its key, while these numbers are situated adjacent to each other on one row of the keyboard. This is an added reason why it is better to use an explicit message for QUITting.

6.3.2. The human decision factor

In this subsection we refer to the ´operator´ as the human side of our Machine-Human combination; some literature reserves this word for a special case of computer users but this is not the sense adopted here.

The operator will be responsible for taking many of the critical operational decisions involved in the subsystem´s operations, while the machine is programmed to decide whether or not that human decision is within the constraints of the system (the constraints of the system are in fact the answers allowed to its prompts by the running program) and allowing only those which are. In addition to this, the machine is capable of evaluating a situation and suggesting alternatives (provided that the relevant information is there) from which the operator can decide.

In this respect we must look into this factor in terms of timing and the kind of decisions which are to be taken by the operator, and his capacity to handle them. This could be covered by answering the following questions:

**WHAT the operator is to decide for the machine?

**WHEN the operator is to decide for the machine?

The WHAT? question can be looked at in terms of three sub-questions:

1)What can the machine decide?

To this question the answer is that the machine can be made to decide many things, but we are not looking for its maximum capacity right away. It all depends on the complexity that is decided for the system; as the designed complexity is increased, the machine is then pushed closer to its limits. A major exception here are decisions that are within the designed complexity but not assigned to the machine—they are those with psychological aspects. Another exception is when the system needs a natural dumping mechanism, rather than a tightly designed one, for disturbances encountered.

If we revised this question in the light of the foregoing, then it would become: What can the machine decide for us, that we want it to decide? then the answer in general terms is that we want the contribution of the machine to our Machine-Human combination to be in those fields which the machine is known to be better than the human. In other words, we want the machine to be an aid to regulate the complexity of the environment that is facing the operator. Regulating the environmental complexity (variety) is done by amplifing the operator's variety and filtering that of the environment. One of these fields, in which the machine is accepted by many to be better than the human, is in fact-finding activities. (The term 'better' used here means faster and more accurate than the human.) It can amplify the human's capacity to retain a huge volume of data without fear of losing accuracy, and retrieve it very fast. Both are done in a capacity exceeding that of the human, in a manner that cuts short the time gap between the environment changes and the operator's actions on them, and at the same time puts at the operator's disposal a wide range of information to help him deal with these changes. Other activities accompanying the retrieval process are the processing of the stored data and information, and filtering the information to allow only those relevant to pass through; in all such activities, the machine is known to be more efficient than the human.

At any point in the future, provided that the metasystem decides to increase the complexity of the System, or new development in technology came through, or both, then the changes required could be met without changing the infrastructure of the System. Only the contribution of the two sides of our combination would be revised, where the human contribution, at a certain level, would be decreased while that of the machine would be increased.

2)What can't the machine decide?

This question is similar to (1.a.) in the sense of its openness to review, in the light of future developments in technology. Its answer will show (at the time of the examination) all the technological limitations to which we should find an alternative in the available technology (whether in processing or communication).

3)What is predetermined with respect to control?

Here we are looking at a wider scope than that of the subsystem. For control is a factor that integrates the subsystems of the System in addition to its internal duties. These should be determined as non-compromising activities that would be assigned to the machine. They include all those activities that are not covered by the autonomy privilege.

After deciding what decisions the operator is going to contribute to the combination, it is important to determine the timing of the machine's prompts, that are guiding the operator towards these decisions. This is where the WHEN? question has to be considered; the timing here is taken relative to the running time of the program procedure concerned.

A decision cannot be made at a time when the required information is not available. In our situation (i.e. having a computer running and waiting for an input) we should then assume that the operator, as the need rises, should only be asked to make decisions whose relevant information is available at hand. A question by the machine that incorporates all the correct considerations mentioned before, but pops up at the wrong time, could prove costly for the system. An example of such a question that may arise is when a delivery of a certain item takes place while the subsystem has not ordered exactly the amount received, or had never ordered that item beforehand at all. The machine, noticing this inconsistency, then asks the operator: IS THIS

QUANTITY MEANT FOR YOU...(Y/N)?. In order for the operator to answer such a question, he would have to halt all operations to make the necessary enquiries to find out the right answer; this might take days, if he can ever get it at all. A question of that type has, in addition to the delay factor, no safety (security) measures; for if the operator wishes to avoid any delay and any extra work, he would answer YES (or NO), terminating thereby any further processing, and corrupting the system as a result if his answer does not agree with the reality. It can be argued here that the experienced operator may know by now how to deal with such situations, but if we went with this argument we would be violating our principle of allowing for the inexperienced. In cases like this, two methods can be pursued. The first is to makes the machine check the orders to see if any combination of them makes up the delivered amount. If not, or if no previous order exists, then the machine should reject this quantity and refuse the delivery (many mechanical devices exist which can be linked to the computer in use to take orders from it whether to stamp or not), forcing the operator to call upon another subroutine where there is no danger of corrupting the data, if he wishes to enter the quantity to the stock (the operator has no freedom where the interests of the wider system are concerned). By running this second subroutine it would be possible to stamp the delivery papers as DEFERRED (not

RECEIVED), send a message to the subsystem concerned informing it about the error and stating the need for that item (with reasons), and wait for a specified period of time before actually using it (unless a reply message came through before that). The second method, which would be used in conjunction with the first one, is to offer the operator help regarding the situation and, if he wishes take him step by step through the proper procedure. The implementation of the second method by the system is in itself an enhancing factor to the accumulation and distribution of knowledge (and experience) inside the System. A detailed example is given in Subsection 6.4.2.3 where it is recommended that various types (in speciality and complexity) of expert systems may be installed in such cases.

There are other activities which are regarded as trivial decisions when examined in the context of the system´s operations, but are as important as the major ones, if the psychological aspects of the human side are to be emphasized. These decisions come under the heading of freedom of choice for the operator, which is very likely to be compromised for the benefit of efficient or tight control programs by the system designers and programmers.

These decisions are: When to start?; Where to start?; What task to choose?; and What item to choose? to name but a few. It is very easy to compromise these activities by a program if it is designed around a sequential logic that would take these matters one by one whether the operator likes it or not. A job where logic follows a predetermined sequence of processes is not involved here, for the simple reason that doing the job has an overriding priority. It is also preferable, in such cases, for the sequence to be taken care of by the machine, to ease the burden of the operator having to know it. If, however, there is no predetermined sequence governing the job, it is always possible to design the program in a manner which would give the operator a substantial freedom in deciding on his next action. The real value of these decisions is moral rather than material, in that no matter what sequence is chosen, the final result is the same for the machine. It is therefore very important that these decisions should always be left for the human side of the Machine-Human combination, no matter how much we increase the complexity of the program.

6.4. The Inventory Updating Programs

The simulation needs two separate application programs, one for the daily updates which are done locally, and the other for updating the inventory files of each rank for the benefit of the higher ranks. The total number of inventory files operated on in this demonstration is 21 files. Those belonging to Rank 4 were updated by INVUPDT.CMD program, and the rest by INVUPDT.COM.

INVUPDT.COM is an 8080 Assembly Language Program employed to carry out the global summarization of the inventory. Some of the programming techniques used in this program are taken from Barbier (1983), and the Digital Research CP/M Users Manual (1976).

INVUPDT.CMD is a dBASEII command language program, run on a SUPERBRAIN microcomputer, which is a CP/M based system. dBASEII is a trade mark of Ashton-Tate.

6.4.1. INVUPDT.COM

This is the assembly program responsible for inter-rank updates. It will operate at an upper rank taking the inventory file of that rank's subsystem as its master file and all the inventory files of all the sub-subsystems of that rank's subsystem as its transaction files. The transaction files will be called one by one in the ascending sequence of their code number to update the master file and at the same time rearrange them for future use. The operator's responsibility is to load this program at the beginning of the run by giving the machine a command line containing the name of the program with the name of the master file as its first parameter, while the second parameter is the transaction file with the lowest code number in that group of transaction files, each preceeded by its drive number (or code).

A command line would take the form:

A>INVUPDT B:INV-0111.DBF B:INV-1111.DBF

telling the operating system to find the application program INVUPDT on drive (A:), and to take as the master file the file named INV-0111.DBF which is stored on drive (B:), while the transaction files group will start with the file named INV-1111.DBF which is on drive (B:) also.

A confirmation message is then displayed allowing the operator to cancel the run if he so wishes. The program will not interact with the operator after this message, except for displaying informative messages tracing the progress of the run, until all the updates are finished, when it will ask the operator if it is alright to store the updated version of the master file instead of the old one, whereby it will destroy the old version on the disk. Only nine transaction files in one group are allowed by this program.

This same program can be used at any rank(except Rank 4 where the second program will be sufficient for its updates) at any subsystem. The only changes required are to the first and second parameters of the command line. The example above showed a command line entered at Rank 3 and operating on transaction files at Rank 4 starting with file INV-1111.DBF. Another command line entered at Rank 2 would take its transaction files from Rank 3 only, and will be as follows:

```
A>INVUPDT B:INV0011.DBF B:INV-0111.DBF
```

In general,a typical command line will take the form:

```
A>INVUPDT B:INV-0???.DBF B:INV-???1.DBF
                00??            0??1
                000?            00?1
```

The program (INVUPDT.COM) as a whole can be divided into seven logical blocks:

1)Housekeeping.

2)I/O handling of master file.

3)Matching master and transaction records.

4)Updating the matched records.

5)I/O handling of transaction files.

6)Utilities.

7)Memory allocation and variable definitions.

6.4.1.1.Housekeeping

This block starts from the beginning of the program and ends by the last statement of FINISH: subroutine. The first part of this block will initialize several variables (START:), then confirm the run of the program (START1:), when on the operators confirmation a subroutine that will read the whole of the master file into memory is called. The current transaction file is then opened (START:2) and made ready to be accessed from an assigned buffer, and pointers are set to the first logical records in each file, when a subroutine to match them is called. The last part of this block (FINISH:) is branched to when the end of each transaction file being processed is encountered, to increment the transaction file code by ´1´ and reset the master file to make it ready again for updating by the new transaction file.

6.4.1.2.I/O handling of master file

This block will locate the required master file and try to open it successfully (GET:), otherwise, it will inform the operator and stop the run. If opening the master file is successful, then it is read into memory (GET1: and GET:2) sector by sector (GET3:). During this time the checking is continuous on the read process, branching when applicable to the apropriate error routine, and at the same time keeping track of the available space in memory with the appropriate error routines. When the opening of the master file, reading it, then writing it to the memory is completed sucessfully, a pointer is set to the first logical record in the master file (GETEX:), and the master file is then prepared for updating (PRPMSTR:).

The second part of this block writes back (PUT:) the updated version of the master file from memory to an assigned disk, after erasing the old version from that disk if it does exist.

To access the logical records of the master file after it resides in memory, the (GETM:) subroutine is called.

6.4.1.3.Matching the master and transaction records

The (COMP:) subroutine goes through both files sequentially and stops whenever two records are matched, at which point a call to the updating subroutine is issued. All the activities associated with sifting through both files are taken care of here, especially for the transaction file where it is required, by the technique of using two buffers, that the sectors finished with (i.e. the physical records) are written back to disk before writing a new sector in their assigned buffer in order to preserve the changes made to them during the run.

An additional field may be added here to the record to serve as an ´approximate´ flag. If the transaction file at one of the sites was not executed (for any reason), and an estimate has been worked out and added by the program, then this flag is set ON. This same flag can be designed for a third situation, that is in addition to the 'normal' and 'approximate' situations, where it can also be set for a 'not normal but not approximate' situation, when the file for the site is not executed but no estimate is made either.

6.4.1.4.Updating

When the two records are matched from both files, there are five different updates performed by this block on five different fields in the master file. Three of them are quantity fields and the other two are flags, plus the necessary calibration for the transaction record. The three quantity updates are:

1)The quantity available to bring it up-to-date (QTY:AVLBLE).

2)The quantity used to bring it up-to-date (QTY:USED).

3)The quantity ordered to bring it up-to-date (QTY:ORDRD).

Another two updates are done, to set and reset indicator fields (flags) in the record, and they are :

1)Set or reset the quantity availability indicator to YES or NO in the master file record depending on the conditions gathered from all the processed transaction files.

2)Set or reset the orders indicator to YES or NO in the master file record, depending on the conditions gathered from all the processed transaction files, but set to YES if one or more transaction record indicated a YES.

The purpose of these updates is to make these five fields reflect the situations they represent as they now stand. In the ´available quantity´ case, the master field ends up containing the sum of all the ´available quantity´ fields at each subsystem belonging here (i.e. those situated in the transaction files). Therefore, the old quantity of the master field does not belong here any more except as an indicator of what was available prior to this updating, and it is thus moved to another field (AV-LST-WK) and the field is zeroized to prepare it for the summation process. The old quantity of the transaction record should stay as it is except for moving it to the last week field (AV-LST-WK). The final decision on updating the availability flag is actually made at the end of the run (DONE:) where, if all the transaction records processed had indicated a YES, then the master is set to YES, otherwise it is set to NO.

As for the ordered quantities, the updating is done by totalling all the transactions into the master´s field after zeroizing it, but without first moving the old quantity to another field, as is the case with the other two quantities, since it can be produced mathmatically if required as follows:

Quantity ordered as at last week

= (QTY:RCVD) + (QTY:ORDRD)

where (QTY:RCVD)

= quantity received in between two updates

= (QTY:AVLBL)-(AV-LST-WK)+(QTY:USDE)-(USD-LST-WK)

The updating of the orders flag is done in a straight forward manner by scanning all the transaction records so that, if the flag under consideration is YES in any of the scanned records, then the matching master record is set to YES, otherwise it is set to NO. The arrangements for updating the two flags cannot show the higher rank which of the subsystems is in fact has items on order or below level, but that information could easily be obtained from the reports sent by the subsystem concerned, if such details are required.

All the additional operations of the updates are handled by the (ADDA:) subroutine, which is supplemented by the (CARRY:) subroutine.

6.4.1.5.I/O handling of the transaction files

The operations performed by this block are reading and writing the transaction file sectors (physical records) stored on the disk through a memory buffer, after opening the file (OPENFL:). The technique used here is totally different from that used for the master file.

The technique used in handling the master file is a standard one which is usually used in the micro, because of its simplicity and easy maintanance. Its only disadvantage is the vast amount of memory space which the second technique would save. This is however not an efficient utilization for the micro's capacity which is needed in our type of work. The technique used in the mainframe is more efficient in that it uses a memory 'buffer' to hold one physical record (sector) at a time. We followed the same idea and wrote the necessary instructions for the micro to demonstrate that even the sophisticated traits of the mainframe can be emulated by the micro. The two buffers technique, which we wrote, uses 128 bytes of memory space for each of the buffers. The use of two buffers is necessary to capture a complete logical record in memory when one record overflows from the end of one physical record to the next. The buffers are filled, after being done with, with new data in an alternate sequence to secure the capture of complete logical records all the time. Meanwhile, the updated buffer is written back to disk after updating the whole of its last logical record (i.e. including its overflowing part in the second buffer). The updated physical record is placed back on disk in its original sector.

In our demonstration we used the data files that are created and maintained locally by a database system. One of the features of this system is the maintanance of ´last update date´ which is stored in the first sector of each file, which we did not alter during this program´s manipulation of the transaction files since we have not changed its data. (This date should however be altered for the master file.)

The reading and writing back of the transaction file´s updated record are each done in two different stages. One deals with the physical records of the file, due to technical reasons, and the second deals with the logical records as the updating process requires. The first stage is reading or writing (GETREC: & PUTREC:) a physical record (sector) from or to its disk storage, to or from a memory buffer (GETT:), while the second stage is the updating of the logical record after moving it (MOVREC:) to a work space in memory called (RECORD).

For this purpose, two memory buffers are used to ensure that if a logical record is extended into the next sector, we can capture the whole of it in memory. The two buffers are filled initially as we access the transaction file for the first time (START2:). From then on, it will be the responsibility of(OUTPUT:) subroutine to write back to disk the finished buffers. (A buffer is considered finished with when all the logical records in it have been updated, especially if it contains a partial record at the end of it, when the writing back to disk has to wait until the whole record is updated and written back from RECORD to this buffer and the next one.) Once a buffer is written back to disk it is free to accept the last-reached sector on the disk. This operation involves keeping track of the corresponding buffer to each sector on the disk, so that the physical records are written back to disk in the right order, and at the same time keeping track of the current active buffer.

This technique demands the continual checking for the end of file flag and the continual updating of the record number of the file control block of the disk (DCRFCB: & INCFCB:).

6.4.1.6.Utilities

The utilities are mainly to display messages (error or informative) and to accept the operator's response to these messages. Apart from the usual error messages, they involve displaying the full name of the master file being run, and its drive number.

There are subroutines considered here as utilities but these should be kept in the library disk, and are used for debugging purposes only. They include dumping subroutines for several significant memory locations that can be called in between the statements of the program to trace an error.

6.4.1.7. Memory allocation and variable definitions

The variable definitions are situated at the beginning of the program listing while the majority of the memory allocation statements are situated at the end (apart from a few statements left within their subroutines for easier recognition of their use).

6.4.2. INVUPDT.CMD

This is a database program responsible for the local updates within the subsystem, that will operate on the inventory file of that subsystem, whose name must be supplied by the operator. It consists of a main program (INVUPDT.CMD) that will call four other programs, QTTYUSED.CMD, QTTYRCVD.CMD, QTTYORDRD.CMD, and REPORTER.CMD, to perform the various updates required locally, and provide a reporting facility on the status of the inventory which is extracted from the inventory file. A sixth program, RCVDINFO.CMD, is called by QTTYRCVD.CMD if help information about one of its prompts is required. (Other ´help´ files can be written also to be associated with other operations.)

These updates can be considered as semi-on-line updates except for the fact that they are done manually (i.e. the operator calls these programs hence deciding the actual time of the updating process). Some of these activities may be considered as on-line, such as updating the received quantity and the ordered quantity, since they are made at the time of arrival, in the case of the received quantity, or at the time the order is confirmed in the case of the ordered quantity. Updating the used quantity and that still available is done once in a set period of time, to be decided by the management of the subsystem. In the latter cases, on-line updating is not necessary, and is not indeed feasable.

The operator calls the main program only, by entering its name, while the other five are called by the main program, depending on the operator´s choice of options. But, before the operator can do so, he first has to enter the environment of the database. Therefore, there will be two command lines to be entered: the first is a system command line to call the database, the second is a database command to call the updating program.

```
A>dBASEII
**** a sign-on messages here ****
.DO INVUPDT
```

The six programs are considered here together as one program, as they logically are, and that program can be divided, according to its main functions, into five different parts:

1)Main calling program.

2)Updating the used quantity (and adjusting the available quantity as a result).

3)Marking the quantity received.

4)Marking the quantity ordered.

5)Reporting facility.

There are various techniques common to all six programs, designed to make them as friendly as possible to the operator. To start with, each program has a sign-on message describing the services given by it, thereby giving the operator an idea of what to expect.

On entering any subprogram from the main program, control will remain within that subprogram, allowing the operator to perform the same function for more items. Otherwise, on the operator response, control is returned to the main program, allowing the operator to choose any of the options provided.

On error within any subprogram, a choice is given to the operator if he wishes, to try again (TRY AGAIN...(Y/N)?) after a clear explanation of what was wrong.

Two of the subprograms are menu-driven programs (the main program and REPORTER) where each option is selected by a number.

6.4.2.1.The main program

This part is performed by INVUPDT.CMD which will prompt the operator to decide the file name and indexing key on which the operations are going to be performed. After the requested file is indexed and ready, the program will prompt four different options, representing the four functions performed by its subprograms. On the chosen option number this program will call the appropriate subprogram.

6.4.2.2.Updating the quantity used

This function is performed by the QTTYUSED.CMD subprogram. The operator is required to enter the key code of the item being updated, after which the program will locate its logical record and prompt the operator to enter the amount used. On the operator's answer, this program will add the entered amount to its field (QTY:USED), deduct the same amount from the 'available quantity' field (QTY:AVLBL), and check the new value of QTY:USED against QTY:PLAND so that, if it is greater, a message is displayed to the operator to that effect. The availability level of reorder is also checked, so that if the new value of QTY:AVLBL has passed this level (taken as zero in this program) then the availablity flag (AVAILABLE) is set to NO. If, however, that value is less than zero, then an error has occurred somewhere and an appropriate message is displayed to that effect.

6.4.2.3.Marking the quantity received

This function is performed by the QTTYRCVD.CMD subprogram with the help of the RCVDINFO.CMD subprogram. After the operator enters the key of the item being updated, the program will locate the required logical record and prompt the operator to enter the received amount, where it will be deducted from the 'ordered quantities' field (QTY:ORDRD). If the result of the subtraction is not negative, it is stored in (QTY:ORDRD). If the received quantity is exactly what has been ordered, then the orders flag (ORDRD) is set to NO, and the value of the field (QTY:ORDRD) becomes zero. In both cases, the received quantity is added to the available quantity (QTY:AVLBL), and the availibility flag (AVAILABLE) is set to YES (if it is not already).

If, however, the result of the first subtraction was negative, then this means that an amount has been received but had not been ordered previously, indicating an error on which the updating should not be allowed to take place. The RCVDINFO.CMD program is on hand in such cases to help the operator deal with this sort of situation, by responding with YES to the prompt MORE INFORMATION...(Y/N)?, where QTTYRCVD.CMD will call RCVDINFO.CMD.

The RCVDINFO.CMD program is a primitive expert system which can be replaced by a proper expert system. It is being used here as a simulated situation where expert systems can be integrated in the application programs of our System. The main purpose, therefore, is to provide expert advice on the more complicated situations facing the operator who is only familiar with straightforward tasks of updating and reporting. With large scale systems of the kind we are studying here, there is a definite possibility of becoming involved with mistakes that have originated at higher ranks.

Such mistakes are better left to be dealt with by a developed acceptable norm originating from accumulated experience; the rigid control of correction may prove more costly than the mistake itself. The situation chosen for simulation here is one such mistake, where a load of a certain item arrives by transport at a site which uses it in its operations, and has a need for it currently. The usual procedure for the operator is to enter the amount of the quantity arrived in the database. Suppose the database tells him that the received quantity has not been ordered by the subsystem beforehand and, therefore, the initial assumption is that there has been a mistake. Allowing him to enter the quantity now delivered would corrupt the system, and this update is therefore refused. The initial reflex reaction of the operator is to check the paper work of the delivery and make sure of the information entered in the database; if the results are still the same, then there is a definite possibility that this load is meant for another subsystem. At this point serious consideration should be given to the correction procedure. The rigid control-orientated alternative is to turn the shipment back to its sender, a situation that would result in:

1)An average loss of one day of delivery time.

2)A definite loss of transportation cost.

3)If the transportation system is being used to the optimum, then what should be an empty vehicle cannot now be used to transport back another shipment from the surrounding area.

While, on the other hand, leaving the load at this site will free the transportation vehicle, provide the site with required material, and compensate for the time lost in delivering this material, by making the delivery.

6.4.2.4.Marking the quantity ordered

This is performed by the QTTYORDRD.CMD subprogram to mark any amount ordered by the subsystem running it. After locating the relevant record (as in previous programs), the amount is entered and added to whatever is stored in the ´ordered quantities´ field (QTY:ORDRD) while making sure that the orders flag (ORDRD) is set to YES. No other processes are involved.

6.4.2.5.Reporting

This facility is performed by the REPORTER.CMD subprogram which will display any of the four main reports chosen by the operator to reflect on the status of the inventory. The reports provided by this program are:

1)A listing of all items that are marked as ´not available´ (i.e. passed the reorder level). This is done by checking the AVAILABLE flag for NO value and, if it matches, then the contents of NAME and CODE are displayed on one line.

2)A listing of all items on order, given by checking the ORDRD flag for YES value and, if it matches, then the contents of NAME, CODE, QTY:ORDRD and QTY.PLAND are displayed on one line.

3)A listing of all items, usage of which has exceeded planned figures. This is done by checking QTY:USED against QTY:PLAND so that if it is greater then the contents of NAME, CODE, QTY:USED and QTY:PLAND are displayed for that item on one line.

4)A listing of all items whose amounts are considered not available and have not been ordered yet. This is done by checking the AVAILABLE and ORDRD flags so that if both of them are NO, then the contents of NAME and CODE fields are displayed for that item on one line.

VII. CONCLUSIONS

The problems that are hampering the progress of developing countries, which are directly intervening in their own development process, are typically found in the area of control and communication. This is evident in their failure to achieve their own planned progress, due to difficulties in managing the implementation of their Plans. The multi-dimensional contents of these problems proved the feebleness of orthodox managerial techniques.

Cybernetics is the right approach in this case for two main reasons: (1) it is the science of control and communication, and (2) it provides a unified approach to world activities and is therefore comprehensive.

Analysing the developing countries situation, as an opening step to finding the solution, led to the examination of the three fronts of technology, management, and viability of existing institutions. This showed weaknesses and shortcomings on each front that are affecting each other, but can be put right through effective organization and intervention techniques.

Technology has proved to be essential to the development process, and information technology is the core of the new technology. Possessing it involves two main requirements: purchasing power and technical know-how. We looked upon the know-how as having two components: (1) technical expertise essential for utilizing the new technology with respect to operating it, and (2) conceptual understanding of this technology with respect to managing it and organizations using it. To purchase this technology, existing wealth needed to be maintained. For this reason, highly technical people had been transfered to managerial posts. This, understandably, did not improve the management at all on the one hand and, on the other, it deprived the technological front of much-needed people. The continual dependence on orthodox management on all fronts disillusioned people even more. For these reasons we had to look for a solution which is not expensive (proportionally) and at the same time offers ways of overcoming the shortage of know-how.

We started our search with the problem of technology. To begin with, we considered the impact on the host country. We expect no social upheaval associated with the introduction of this technology within the scope we are suggesting here—there will be no redundancies for example. But the serious problem will be in obtaining the missing experience which forms a gap that has to be filled if an efficient utilization is to be achieved. Such gap had been crossed by the advanced countries with natural graduation. The experience gained from this graduation had spelled some changes to the society together with a grasp of utilizing the technology. The social changes which resulted from that may not be welcome in the developing countries, but the technical know-how is. Transferring the know-how may not necessarily be accompanied by changes which are not welcomed by the developing countries. This is the inevitability of change, and it is a price which has to be paid for progress. At any rate, the scope to which we are proposing to use this technology is a limited one, and therefore can be treated as an experimental field which can be isolated if need be. At the same time, this limited scope forms the foundation on which the whole country is being built. A successful performance in this field would result in transforming the whole country and deliver it into progress. This fact then justifies the risk.

A close examination of the issue of ´experience´ showed that an alternative is found in cybernetics, and in particular in the principle of intervention. Managing in cybernetic terms is based on clear principles which are equally efficiently applicable to both simple and complex systems. (All systems are naturally complex but the distinction is made here with respect to the user´s comprehension.) The alternative to immediate expertise with respect to operations can be found in the software packages which are available in the open market, and for very reasonable prices.

The answer to providing new technology started with the issue of 'information technology'. This then was divided into two main components: the computer as a data processing machine and Information Systems. The computer, although a machine, has many traits that would compensate for its lack of human intelligence. Interfacing the human with this machine in a designed and disciplined combination will certainly amplify the powers of the human.

The rapid development of the computer meant its inevitable involvement in industry and management, thereby resulting in further improvements to the intervention techniques in these areas. The microprocessor is the recent development in computer technology which has changed, yet again, the approach to computerization. With it, processing became easier to distribute, and the limitations of, and difficulties associated with, centralized processing have now been overcame. The proliferation of the powers of the micro offers the managers more solutions for their present difficulties, at affordable prices. With this, a whole range of application packages have become available, again with negligible prices compared to those of the mainframe. These are essentially up-to-date techniques which are the result of a long and well-tested experience in the field of application concerned, which is the very thing needed by the developing countries.

Solving the problems of hardware and software should be accompanied by a successful utilization in order to reap their full benefits. This can be done by an effective Information System. An initial step to building such system was the analysis of information involved. Ways of reducing the volume of the information concerned were established as a result. Such analysis also helped in identifying the scope of operations (with regards to information) of the various subsystems to decide on the monitoring requirements. As for the best way to go about building such system, it appeared that it must start from scratch. This will take two tasks into consideration: (1) learning from present information, and (2) accumulating present information for future use.

In our work, we used a real-life example to help illustrate our proposals. One aim of this thesis was to enhance the speed of information-gathering regarding the performance of the Plan implementation which is carried out by such system. This system is also playing the role of a reporting system to the main governmental body responsible for both finalization and monitoring of the National Plan. Both these tasks needed reliable information about the performance of the whole implementation process, and in the right time. To monitor, and therefore to take the proper decisions, needed representative information with respect to time, in order for the decisions to be correct. This will by itself become valuable information when it comes to finalizing the next National Plan. This is being solved but there still remains the problem of time lags. Effective organization with reliable communication are all that was needed.

We followed the cybernetic approach of identifying the ´system´ as an initial step in our organizational study. We aimed at achieving partial knowledge about the system that is complete within itself, but is sufficient for our ultimate practical purpose. It started by defining the ´system´ as: consisting of a group of elements dynamically related in time according to some coherent pattern, and having a purpose attributed to it by the observer. The environment of such a system includes objects and changes which exert considerable influence on it without being part of it, and this environment includes all objects and phenomena which feel the strong and direct (or not too remotely mediated) effects of the system.

Any model for the system which is based on the cybernetic definition and principles is therefore the right tool to produce the comprehensive solution we were after, i.e. the cybernetic solution. This was provided by the ´viable system´ model, to examine and to reinforce the present organization.

The mapping of the envisaged system showed the need for no more than four recursions (ranks) in order for the System to be effective. The identification of the recursions was based on the specialization, and then on the scope of operations. This mapping declared a major problem on which is otherwise a problem area that was hidden by the traditional organizational modeling. The power of the used model was not only in such identification, but, and what may be its real power, it provided for an easily recognized solution for the problem. Our system is abstraction orientated, while other systems involved in the implementation of the Plan are real-life orientated. The involvement of our System in the affairs of other systems while performing its task, was a major source for human conflicts as a result of direct contacts between people belonging to different systems that have different purposes. The need therefore was to reduce this contact as much as possible. This was provided for by the suggested organizational structure. In addition to that, this organizational structure took into consideration the cybernetic principle of freedom also, which will lessen the amount of intervention in the subsystems. This is a positive measure, among others, which is helpful in spreading the responsibility and reduce its burden, and in eliminating the need for management by dictation.

The establishment of autonomous subsystems meant the presence of local computational needs. The microcomputer is powerful enough, and cheap enough to provide every one of these subsystems with a local processing facility, and thus automating the information-gathering. This had laid the foundation for linking together the whole System electronically, and thus integrating the whole process.

With the automation of the information-gathering, and the electronic linkage, plus the fact that our System is not responsible for the ´direct´ intervention in the operations of the subsystems beyond its first recursion (except for the information-gathering task), the extension of our System only electronically beyond its first recursion was then: (1) possible, and (2) eliminates human contacts. This approach showed the following benefits: (1) The speeding-up of information-gathering —— in fact it can be described as close to on-line within the context of the System. (2) Provided the means for an efficient assignment of functions, both for our System and the systems implementing the National Plan (this cannot be clear with the present organization), thereby enabled the establishment of an efficient control and co-ordination inside the System. (3) Eliminated human conflicts. (4) Other local tasks of the subsystems can now be automated, thereby increasing the local efficiency. (5) Opened the way for the use of packaged software, thereby benefiting from other people´s experience.

Another negative aspect of the present organization was that any observer would see separate systems trying to perform the same task (i.e. the implementation of the National Plan is being done according to the specifications). This is due to the confusion surrounding the real functions of each of those systems which is definitely contributing to the inefficiency of both of them. (While these functions are clear in the mind of the ultimate metasystem of these systems, the organizational structure is killing this clarity.) Our proposal resulted in viewing those separate systems as a ´single system´ within which each of them is performing a separate and clear function, which is the real one.

With the main proposals completed, it remained to examine the possibility of implementing them. Two main considerations were studied and they were the information-handling and the communications structure. Analysing the types of information involved in the operations of the System helped in deciding on the way their storage has to be organized. This analysis showed two distinct types of information: ´local´ and ´global´. This had risen from the need of the subsystems to maintain their viability, and hence the existance of a distinctly local type of information; and from the need of the System as a whole to keep the inter-subsystem interaction dynamic to ensure the holistic effect, and hence the existence of a distinct global type.

The stream of this information will enter the System from two directions: either from its highest rank or from its lowest rank, and spread inside throughout the ranks. The number of terminal points (i.e. the subsystems) envisaged at the lowest rank were considerably more than those at the highest rank. The volume of information entering from this rank, which can be handled by its large number of terminals, has to be gathered at the upper ranks' terminals. This volume can be reduced via various techniques without affecting its informative value to the respective rank. Several benefits point here to the alternative of 'storing' the relevant information, to each subsystem, at the site of that subsystem, i.e. spreading the information over the system. This being the case, a major problem was met in the area of preserving the integrity of stored data due to the presence of global items that are stored at more than one site. The updating of any global item must be propagated to cover all its replicas. The global commands should be capable of ensuring this update. This propagation is also needed to channel information in general, and the control and co-ordination messages in particular, and therefore demanded the establishment of communications links between the various subsystems. The model of the viable system determined these channels, and had to be followed therefore to ensure the viability of the suggested organization.

Two main types of connection had resulted from the functions of these channels: ´vertical´ and ´horizontal´. The vertical direction was to serve the channelling of the command-type messages, and also handling the main information flow entering the system from both directions. The horizontal connection was to serve the channelling the co-ordination-type messages, plus a minor transfer of information items.

With the establishment of this network of connection channels between the subsystems, a wider door was opened for utilizing the computer powers of automation that can improve the overall performance of the System. One aspect made possible by this is the automatic relay of messages, which when examined closely showed a proper utilization for the command, monitoring, and co-ordination channels which are provided by our terminal organization and communications network.

We had to look at (and treat) the two, differently natured, organizations (i.e. the real-life and the planning-orientated) as ´one system´, because in effect that is what they are. Organizational necessities demanded that they had to be two separately organized systems. In reality, only the first rank of the CPFS has to exist physically. All the other ranks are information-gathering points. In the ´implementation plane´ however, all the mapped establishments (i.e. those excuting the Plan) had to exist physically. The two existing organizations can go different ways but they had to meet at a point where the ´ultimate purpose´ is common to them both. This point exists at the highest-most ranks of these different organizations. As we considered the ´abstraction plane´ and the ´real-life plane´ together with the ´Planning Board´ (this is the highest ranking—in authority and power—body in the country with respect to planning) and examined their situation, we observed a situation similar to that existing within the metasystem region of the viable system. The Planning Board mapped exactly on System Five function, while Rank 1 of the Plan (CPFS) was heavily involved in a System Four function, and the implementation establishments (which would be mapped at the same rank) were heavily involved in a System Three function. (The interaction between them is necessarily similar to that explained by the model.) We have used here the term ´heavily involved´ since each of these organizations is not, as they are now, exactly mapped at the function concerned. Ideally they should be, but practically they are not, due to the existence of a well-established organizational structure. Each of them was heavily biased towards one function and performing part of the other. This was due to the ´forced´ breaking-up of the natural organization, with each side having the necessary components to perfom what they are actually performing now. In our approach we considered them together (in their natural form) and, with the help of the model, we have channeled each component through its appropriate function. The principle which we had to consider carefully at this stage is that, while Rank 1 of the Follow-up has the authority to reach any information in the other ´plane´, it should not issue commands to the lower ranks there. These commands had to come from the higher rank of the intended subsystem. This condition correctly implied that there is the possibility that the command originating from the Plan´s rank does not have the approval of the implementation rank responsible for the subsystem receiving the command. The proposed establishment of the electronic communications network offered a realistic link which was capable of serving the various requirements resulting from this constraint. The final connection at the top rank had to be made in duplicate. A ´four-cornered´ link was then established with the Planning Board, the implementing Ministry, the Sector, and the Field establishment (which belongs organizationally to the Ministry) at each corner. The effectiveness of such a link, in joining the two planes, was demonstrated by two typical cases of intervention: (1) each Sector has to know how its subsystems are performing, and needs to intervene when it is necessary, and (2) a subsystem (say a Project) faces difficulties, its Field decides to intervene but needs the authority from its superordinate (i.e. the action required or suggested is outside its autonomy). In the second case, the Field was made to send the suggestion to its Ministry (outside our System), but the established link forced a replica of the request to go to the corresponding Sector. In both cases however the Sector became aware (through its ranks) of the need to intervene and had its own opinion about what this should be. The Sector and the Ministry will then exchange opinions (interact) where they may: agree, in which case the final decision will reach the Field from its links with the Mininstry and the Sector; or disagree, in which case an exchange on the triangular link between the Ministry, Sector, and Board, is activated under the control of the Board. This is resulting in the final decision which is to be sent to the Field. From there on, the flow will be effectively in a straight line to the other ranks.

The examination of the latest development in technology suggested to us that the technical requirements for the implementation of the foregoing proposals are available, or can be made available. This development is offering the efficient support we needed for the main proposals' requirements: (1) local processing capability, (2) local data storage , and (3) proper communication links between the subsystems.

The distributed database concept appeared to be an ideal option, and a natural one when considering our approach. Employing this concept can be done on the basis of two extremes. One is a complete database which is highly integrated in an homogeneous system, such as that of Tandem´s ENCOMPASS. The other is to utilize local (centralized) databases and connect pre-existing software components in a much more complex environment, comprising several database management systems. (IBM has developed and is developing tools which facilitate building distributed databases on top of homogeneous and hetrogeneous local systems. The most important of these tools are the SNA and the ISC.)

Currently however, no single, commercially available, distributed database management system possesses all the features of a typical distributed database, but the industrial world is moving very fast in this direction. Nevertheless, this does not mean that we cannot employ existing systems (however incomplete) to facilitate our proposals. By the time the complexity of our System increases, and the need arises for a more sophisticated system, the current prototypes will necessarely be fully operational. (IBM´s system R* prototype has many of its features operational now, including data definition and manipulation statements of SQL, transaction management, deadlock detection, and recovery mechanisms.)

The benefits from the distributed database concept, in addition to the typical advantages of databases, are: (1) It fits naturally with our organizational approach of cybernetic freedom. (2) Economically means less overheads. (3) We can still use existing databases. (4) It supports easily and smoothly the incremental growth of the System. (5) With this concept, a total system failure is rare.

To demonstrate the workability of the proposals, and to present them in their operational form, was seen however as an impractical undertaking with reference to time allowed by the sponsors of this work for the completion of the thesis, and the availability of funds to provide the necessary equipment for such demonstration. We however opted for the possible, and the minimum, demonstration to gain the necessary initial acceptance for these proposals. This was hoped to be done by convincing the users of: (1) the capability of the micro in replacing the mainframes in performing the operations of the subsystems, and (2) the possibility of performing global updates on it. This was achieved with a local database system for the micro (dBASEII) running on a Z80 micro (SUPERBRAIN QD), which was capable of performing an efficient inventory program (INVUPDT.CMD) with clear potential for performing other operations needed locally at each subsystem. The updating of the local files which were created at 21 sites, spreading over the four ranks of the System, was carried successfully by an assembly language program on the same micro. This was started by updating the inventory file of each Rank 3's subsystem from the files of its branch sites at Rank 4. The same program then used to update Rank 2's and Rank 1's files in the same manner. This had resulted in each subsystem having available at its site an up-to-date and sufficient information for its operations (i.e. no need to use the communication network to access branch subsystems files later).

The whole suggested proposal here has the necessary ingredient as whether to implement it manually or with the aid of today's technology. It is obvious though that the manual implementation is not as efficient as, and not as comprehensive as, when today's technology is employed to the full. (A partial use of technology, i.e. the use of microcomputers only without the communications network, is possible but it will be understandably less efficient than the full use. The 'floppy disk' of each branch site can be sent daily, by post or by a courier, to their node site.)

It should be emphasised here that the proposed solution is capable of being implemented in 'complete portions' as upgrading stages, in proportion to the required comlexity for the System. The complete System should then evolve (over several years), and adapt naturally to the user environment.

Appendix A

Listings of the database programs:

a) main program INVUPDT.CMD

b) subprogram QTTYUSED.CMD

c) subprogram QTTYRCVD.CMD

d) subprogram QTTYORDRD.CMD

e) subprogram RCVDINFO.CMD

f) subprogram REPORTER.CMD

INVUPDT.CMD

```
ERASE
SET TALK OFF
?'****************************************'
?'* THIS IS THE INVENTORY UPDATING PROGRAM *'
?'* YOU CAN RUN IT ANY TIME AND ANY DAY    *'
?'* IT WILL OPERATE ON THE INV- FILES      *'
?'* YOU WILL HAVE TO SUPPLY THE NAME OF    *'
?'* THE FILE AND THE INDEX KEY---THIS      *'
?'* KEY IS THE FIELD YOU WILL USE TO       *'
?'* SELECT THE RECORD TO BE UPDATED        *'
?'****************************************'
?'        '
?'         '
ACCEPT "ENTER FILE NAME" TO F
USE &F
STORE 'THIS PROGRAM WILL OPERATE ON ' TO A
?'    '
ACCEPT "ENTER INDEX KEY NAME" TO X
INDEX ON &X TO INVENTRY
?'     '
?'===>===>'
? A+F
?'TO PERFORM THE FOLLOWING:'
?'                      '
STORE 'Y' TO I
DO WHILE I='Y'
?'          (1) UPDATE THE QUANTITY USED '
?'          (2) ENTER ANY QUANTITY RECEIVED'
?'          (3) REGISTER ANY QUANTITY ORDERED'
?'          (4) REPORTING FACILITY'
?'                               '
?'        '
ACCEPT"ENTER OPTION NUMBER" TO O
EJECT
ERASE
   IF O='1'
      DO QTTYUSED
   ELSE
      IF O='2'
         DO QTTYRCVD
      ELSE
         IF O='3'
            DO QTTYORDRD
         ELSE
            IF O='4'
               DO REPORTER
            ELSE
               IF O<='0'.OR. O>'4'
                  ?'                  '
                  ?'WRONG OPTION NUMBER'
                  ?'                   '
               ENDIF
            ENDIF
         ENDIF
      ENDIF
   ENDIF
?'                      '
?'DO YOU WISH TO CONTINUE WITH INVUPDT'
ACCEPT "   (Y/N)? " TO I
ERASE
EJECT
```

QTTYUSED.CMD

```
?'===>'
?'This option will subtract the used quantity from QTY:AVLBL'
?'Updating QTY:USED and checking if we have exceeded the Plan'
?'IF the available quantity becomes zero'
?'THEN the AVAILABLE field is set to (N)'
?'                '
STOR 'Y' TO I
?'                         '
SET EXACT ON
DO WHILE I='Y'
ACCEPT "ENTER KEY (NUMBER/NAME)...." TO N
SET TALK ON
FIND &N
IF #=0
   ?'** requested record does not exist **'
   ACCEPT "TRY AGAIN...(Y/N)?" TO I
          IF I='Y'
             LOOP
          ENDIF
ELSE
   SET TALK OFF
   ?'                          '
   ACCEPT "ENTER QUANTITY USED..." TO Q
   ?'                         '
   ?'                            '
   STORE QTY:USED+VAL(Q) TO S
   STORE QTY:AVLBL-VAL(Q) TO B
   IF B>=0
      REPLACE QTY:USED WITH S
      REPLACE QTY:AVLBL WITH B
      IF S>QTY:PLAND
         ?'                           '
         ?'!!! quantity used exceeded Plan !!!'
      ENDIF
      IF B=0
         REPLACE AVAILABLE WITH 'N'
      ENDIF
   ELSE
      ?'                      '
      ?'*** error in quantity: updating cannot proceed ***'
      ?'Quantity Available Becomes Negative '
      ?'       '
      ACCEPT "TRY AGAIN...(Y/N)?" TO I
            IF I='Y'
            LOOP
            ENDIF
   ENDIF
   ?'                         '
   DISP
ENDIF
   ?'                             '
   ACCEPT "ANOTHER USED ITEM....(Y/N)? " TO I
ENDDO
```

```
  QTTYRCVD.CMD
?'===>'
?'This option will subtract the value of the '
?'received quantity from the ordered one'
?'and add it to the available quantity'
SET EXACT ON
SET TALK OFF
STORE 'Y' TO I
DO WHILE I='Y'
?'                    '
ACCEPT "ENTER KEY (CODE/NAME)" TO N
SET TALK ON
FIND &N
IF #=0
   ?'** requested record does not exist **'
   ACCEPT "TRY AGAIN...(Y/N)?" TO I
          IF I='Y'
             LOOP
          ENDIF
ELSE
SET TALK OFF
?'          '
ACCEPT "ENTER QUANTITY RECEIVED" TO Q
STORE QTY:ORDRD-VAL(Q) TO A
IF A<0
   ?'                '
   ?'*** error in quantity updating cannot proceed ***'
   ?'Quantity Received Exceeded The Ordered Quantity'
   ?'                              '
   ACCEPT "MORE INFORMATION (Y/N)? " TO M
   IF M='Y'
      ERASE
      DO RCVDINFO
      ERASE
   ENDIF
   ACCEPT "TRY THIS ITEM AGAIN...(Y/N)?" TO I
          IF I='Y'
          LOOP
          ENDIF
ELSE
   REPLACE QTY:ORDRD WITH A
   IF A=0
      REPLACE ORDRD WITH 'N'
   ENDIF
   STORE QTY:AVLBL+VAL(Q) TO A
   REPLACE QTY:AVLBL WITH A
   REPLACE AVAILABLE WITH 'Y'
   ?'                '
   DISP
ENDIF
?'                '
ACCEPT "ANOTHER RECEIVED ITEM...(Y/N)? " TO I
ENDIF
ENDDO
```

QTTYORDRD.CMD

```
?'===>'
?'This option will add the ordered quantity   '
?'to the QTY:ORDRD field and set  '
?'the ORDRD field to (Y)'
SET EXACT ON
STORE 'Y' TO I
DO WHILE I='Y'
?'           '
ACCEPT "ENTER KEY (CODE/NAME)" TO N
SET TALK ON
FIND &N
IF #=0
   ?'** requested record does not exist ** '
   ACCEPT "TRY AGAIN (Y/N)?" TO I
          IF I='Y'
             LOOP
          ENDIF
ELSE
SET TALK OFF
?'                       '
ACCEPT "ENTER QUANTITY ORDERED" TO Q
STORE QTY:ORDRD+VAL(Q) TO A
REPLACE QTY:ORDRD WITH A
REPLACE ORDRD WITH 'Y'
?'                 '
DISP
?'                   '
ACCEPT "ANOTHER ORDERED ITEM....(Y/N)? " TO I
ENDIF
IF #=0
   ACCEPT "TRY AGAIN....(Y/N)? " TO I
   IF I='N'
      ACCEPT "ANOTHER ORDERED ITEM...(Y/N)? " TO I
   ENDIF
ENDIF
ENDDO
```

RCVDINFO.CMD

```
?'This may be wrong and therefore would not be updated'
?'by this program. A manual procedure is used in these cases'
?'to insure that the data is properly checked before proceeding'
?'with the updating'
?'                    '
?'There are two possible reasons why this data is wrong:'
STORE 'Y' TO E
DO WHILE E='Y'
?'   (1) The Consignment is Meant for Another Contract But Landed'
?'       Here by Mistake'
?'   (2) Your Calculation is Wrong  '
?'                         '
ACCEPT "MORE ON ANY OF THESE POSSIBILITIES..(Y/N)? " TO A
?'                          '
IF A='Y'
   ?'                    '
   ACCEPT "ENTER POSSIBILITY NUMBER" TO C
   DO WHILE C='1'.OR.C='2'
   IF C='1'
      ?'                    '
      ?'The delivery papers may be correct but have accompanied the wrong'
      ?'consignment which means that your consignment may be on its'
      ?'way to another place.'
   ENDIF
   IF C='2'
      ?'                    '
      ?'Your calculation may be wrong because different items are'
      ?'actually packed together, so make sure first that all the '
      ?'items counted are of the same type'
      ?'But if you have been using the figures from the delivery'
      ?'papers then the papers might be wrong, while the actual'
      ?'quantity received is  correct, SO CHECK ACCORDINGLY'
      ?'                    '
      ACCEPT "MORE ON HOW TO CHECK THE PAPERS..(Y/N)? " TO D
      IF D='Y'
         ?'                    '
         ?'......A PROCEDURE FOR CHECKING PAPERS.....'
      ENDIF
   ENDIF
   IF C<='0'.OR.C>'2'
      ?'                    '
      ?'WRONG OPTION NUMBER***'
   ENDIF
   ?'                         '
   ?'ENTER (1) OR (2) TO REVIEW THE PREVIOUS POSSIBILITIES'
   ACCEPT "ANY OTHER NO. WILL EXIT YOU FROM HERE" TO C
   ENDDO
?'                    '
```

```
?'IF the consignment is definitely for you'
?'and your calculations are correct,THEN'
ENDIF
?'You may use this item*** PROVIDED THAT:'
?'    (1) You have ordered the same item before'
?'    (2) Your starting date to use this item is DUE'
?'                    '
ACCEPT "MORE ON AUTHORIZED USE..(Y/N)? " TO B
IF B='Y'
   ?'                    '
   ?'Send a Message to the Upper Rank Containing :'
   ?' 1-The Quantity of the Received Item'
   ?' 2-Your Ordered Quantity'
   ?' 3-Name and Code of the Item'
   ?'                    '
   ?'And Wait for One Week;'
   ?'IF there will be NO OBJECTION'
   ?'THEN and only then you can use it'
ENDIF
?'                    '
ACCEPT"MORE ON MANUAL UPDATE...(Y/N)? " TO B
IF B='Y'
?'                    '
?'.....A MANUAL PROCEDURE.....'
ENDIF
?'                    '
   ?'                    '
   ACCEPT "DO YOU WANT TO ADD ANY PROCEDURE" TO X
   IF X='Y'
      ?'                    '
      ?'YOU MAY ADD ANY USEFUL INFORMATION IN HERE'
      ?'BY FOLLOWING THIS PROCEDURE:'
      ?'......A DESCRIPTIVE DETAIL ON......'
      ?'......HOW TO ADD THE DATA...........'
   ENDIF
?'                    '
?'DO YOU WISH TO REVIEW THIS INFORMATION AGAIN'
ACCEPT "...(Y/N)?" TO E
ENDDO
```

REPORTER.CMD

```
?'===>'
?'THIS OPTION  PROVIDES A REPORTING FACILITY'
?'ON THE FOLLOWING:'
?'                  '
SET TALK OFF
STORE 'Y' TO I
DO WHILE I='Y'
?'          (1) EVERY NON-AVAILABLE ITEM'
?'          (2) EVERY ORDERED ITEM'
?'          (3) EVERY ITEM EXCEEDED THE PLANNED LIMIT'
?'          (4) EVERY ITEM WHICH RAN OUT OF STOCK'
?'              AND HAS NOT BEEN ORDERED YET'
?'       '
   ACCEPT "ENTER OPTION NUMBER" TO O
   ERASE
   EJECT
   IF O='1'
      ?'                '
      ?'THE NON-AVAILABLE ITEMS ARE:'
      ?'..........ITEM NAME.........CODE........'
      DISP NAME,CODE FOR AVAILABLE='N'
   ENDIF
   IF O='2'
      ?'                  '
      ?'THE ORDERED ITEMS ARE:'
      ?'..........ITEM NAME.........CODE......ORDRD....PLAND...'
      DISP NAME,CODE,QTY:ORDRD,QTY:PLAND FOR ORDRD='Y'
   ENDIF
   IF O='3'
      ?'               '
      ?'THE ITEMS THAT EXCEEDED THEIR PLANNED USE ARE:'
      ?'..........ITEM NAME.........CODE......USED.....PLAND...'
      DISP NAME,CODE,QTY:USED,QTY:PLAND FOR QTY:USED>QTY:PLAND
   ENDIF
   IF O='4'
      ?'               '
      ?'THE ITEMS THAT ARE NOT AVAILABLE AND NOT ORDERED YET ARE:'
      ?'..........ITEM NAME.........CODE....'
      DISP NAME,CODE FOR AVAILABLE='N' .AND. ORDRD='N'
   ENDIF
   IF O<='0' .OR. O>'4'
      ?'              '
      ?'WRONG OPTION NUMBER'
   ENDIF
   ?'              '
   ?'DO YOU WISH TO CONTINUE WITH REPORTER'
   ACCEPT "   (Y/N)? " TO I
   ERASE
   EJECT
ENDDO
```

Appendix B

Listing of INVUPDT.COM

```
                     ; ASCII CHARACTERS
000D =               CR      EQU     0DH        ;CARRIAGE RETURN
000A =               LF      EQU     0AH        ;LINE FEED
001A =               CTRLZ   EQU     1AH        ;OPERATOR INTERRUPT

                     ; CP/M BDOS FUNCTIONS
0001 =               RCONF   EQU     1          ;READ CON: INTO (A)
0002 =               WCONF   EQU     2          ;WRITE (A) TO CON:
000A =               RBUFF   EQU     10         ;READ A CONSOLE LINE

                     ; CP/M DISK ACCESS FUNCTIONS
000D =               INITF   EQU     13         ;INITIALIZE BDOS FUNCTION
000F =               OPENF   EQU     15         ;OPEN FILE FUNCTION
0010 =               CLOSF   EQU     16         ;CLOSE FILE FUNCTION
0011 =               FINDF   EQU     17         ;FIND FILE FUNCTION
0013 =               DELEF   EQU     19         ;DELETE A FILE FUNCTION
0014 =               READF   EQU     20         ;READ ONE RECORD FUNCTION
0015 =               WRITF   EQU     21         ;WRITE ONE RECORD FUNCTION
0016 =               MAKEF   EQU     22         ;CREAT FILE FUNCTION
001A =               SDMAF   EQU     26         ;SET DMA FUNCTION

                     ; CP/M ADDRESSES
0000 =               RBOOT   EQU     0          ;RE-BOOT CP/M SYSTEM
0004 =               DRIVE   EQU     4
0005 =               BDOS    EQU     5
0007 =               MEMAX   EQU     7          ;MSB OF TOP OF MEMORY
005C =               TFCB    EQU     5CH
0065 =               FCBTY   EQU     TFCB+9
0068 =               FCBEX   EQU     TFCB+12
006A =               FCBS2   EQU     TFCB+14
006B =               FCBRC   EQU     TFCB+15
007C =               FCBCR   EQU     TFCB+32
0080 =               TBUFF   EQU     80H        ;TRANSIENT BUFFER
0100 =               TPA     EQU     100H

                     ; CP/M FLAGS
0000 =               BDAOK   EQU     0          ;BDOS RETURN FOR ALL OK
0001 =               BDER1   EQU     1          ;BDOS RETURN ONE
0002 =               BDER2   EQU     2          ;BDOS RETURN TWO
00FF =               BDERR   EQU     255        ;BDOS RETURN ERROR FLAG

0100                         ORG     TPA


0100 31910A          START:  LXI     SP,STAK
0103 3A0400                  LDA     DRIVE
0106 32D108                  STA     DRSAV
0109 AF                      XRA     A
010A 324D0A                  STA     FINFLG
010D 324F0A                  STA     FLAG
0110 324E0A                  STA     DIVER

0113 CD1C08          START1: CALL    CCRLF
0116 CD4F08                  CALL    SPMSG
0119 5550444154              DB      'UPDATES INV FILES ONLY',CR,LF,0
0132 CD5B08                  CALL    GETYN
0135 C2C701                  JNZ     DONE
0138 3E30                    MVI     A,30H
013A 32DA08                  STA     NUMFILS
013D CD7F05                  CALL    SAVFCB
0140 CD9705                  CALL    GET        ;READ MASTER FILE IN RAM

0143 CD4005          START2: CALL    OPENFL
0146 AF                      XRA     A
0147 32500A                  STA     EOF
014A CD1C08                  CALL    CCRLF
```

```
014D CD4F08              CALL    SPMSG
0150 50524F4343          DB      'PROCCESSING FILE : ',0
0164 CD6F07              CALL    SHOTFN
0167 CDBC03              CALL    GETREC
016A CDBC03              CALL    GETREC
016D CDBC03              CALL    GETREC
0170 CDBC03              CALL    GETREC
0173 CDBC03              CALL    GETREC
0176 21FF08              LXI     H,INBUFR1
0179 110900              LXI     D,9
017C 19                  DAD     D
017D 22BA03              SHLD    HSAVE
0180 118009              LXI     D,INBUFR2
0183 CDBF03              CALL    GETR
0186 CD1303              CALL    GETT
0189 2AD408              LHLD    NEXT
018C CD0102              CALL    COMP

018F CD1C08     FINISH:  CALL    CCRLF
0192 CDDE07              CALL    CPDMA
0195 21920A              LXI     H,BUFFER
0198 110902              LXI     D,521
019B 19                  DAD     D
019C 22D408              SHLD    NEXT
019F 210000              LXI     H,0
01A2 22E708              SHLD    FCBSAV+12
01A5 22E908              SHLD    FCBSAV+14
01A8 AF                  XRA     A
01A9 32FB08              STA     FCBSAV+32
01AC 32EB08              STA     FCBSAV+16
01AF 324D0A              STA     FINFLG
01B2 324F0A              STA     FLAG
01B5 324E0A              STA     DIVER
01B8 3AE308              LDA     FCBSAV+8
01BB 3C                  INR     A
01BC FE3A                CPI     3AH
01BE CAC701              JZ      DONE
01C1 32E308              STA     FCBSAV+8
01C4 C34301              JMP     START2

                ;THIS IS TO UPDATE THE AVAILABLE FIELD OF THE
                ;MASTER FILE SO THAT IF ALL THE TRANSACTION
                ;FILES PROCCESSED HAD (Y) IN THEIR AVAILABLE
                ;FIELD THEN THE MASTER'S MADE EQUAL TO (Y)

01C7 21920A     DONE:    LXI     H,BUFFER
01CA 110902              LXI     D,521
01CD 19                  DAD     D
01CE 22D408              SHLD    NEXT

01D1 114900     NXTR:    LXI     D,73       ;POSITION REG HL
01D4 19                  DAD     D          ;ON FLAG
01D5 3ADA08              LDA     NUMFILS
01D8 BE                  CMP     M
01D9 3630                MVI     M,'0'
01DB C2EB01              JNZ     LOOP
01DE 2AD408              LHLD    NEXT
01E1 111F00              LXI     D,31       ;POSITION REG HL
01E4 19                  DAD     D          ;ON AVAILABLE
01E5 3659                MVI     M,'Y'
01E7 112A00              LXI     D,42       ;REPOSITION REG HL
01EA 19                  DAD     D          ;ON FLAG

01EB 23         LOOP:    INX     H          ;POSITION REG HL
01EC 22D408              SHLD    NEXT       ;ON NEXT RECORD
01EF 7E                  MOV     A,M
01F0 FE1A                CPI     CTRLZ
01F2 C2D101              JNZ     NXTR

01F5 CD8606              CALL    PUT
01F8 3AD108              LDA     DRSAV
01FB 320400              STA     DRIVE
```

```
01FE C30000              JMP     RBOOT

                ; COMPARE MASTER AND TRANS KEYS

0201 010A00     COMP:    LXI     B,10
0204 2AD408              LHLD    NEXT
0207 111500              LXI     D,21
020A 19                  DAD     D
020B 11160A              LXI     D,RECORD+21

020E 1A         COMP1:   LDAX    D
020F BE                  CMP     M
0210 C22A02              JNZ     COMP2
0213 23                  INX     H       ;OTHERWISE CURRENT BYTES MATCH
0214 13                  INX     D       ;SO TRY NEXT BYTES
0215 0B                  DCX     B
0216 79                  MOV     A,C
0217 B0                  ORA     B
0218 C20E02              JNZ     COMP1   ;IF NOT ZERO:FIELD NOT FINISHED
021B CD4202              CALL    UPDATE  ;IF ZERO: WE HAVE A MATCH
021E CD2604              CALL    WRTREC  ;WRITES BACK TO BUFFER UPDATED RECORD
0221 CD4D03              CALL    OUTPUT  ;OUTPUTS UPDTED BUFFER & GETS NEW ONE
0224 CD1303              CALL    GETT    ;AND GET TWO NEW RECORDS
0227 C3F702              JMP     GETM    ;FROM TRANS AND MASTER FILES

022A 1A         COMP2:   LDAX    D
022B 96                  SUB     M
022C F23502              JP      GTMSTR
022F CD1303              CALL    GETT
0232 C30102              JMP     COMP

0235 2AD408     GTMSTR:  LHLD    NEXT
0238 114900              LXI     D,73
023B 19                  DAD     D
023C 7E                  MOV     A,M
023D 3C                  INR     A
023E 77                  MOV     M,A
023F C3F702              JMP     GETM


                ;THIS SUBROUTINE ADDS EVERY TRANS QTY-AVLBL TO THE MASTER'S
                ;FIELD WHICH HAS BEEN ZEROED AT BEGINING OF RUN
                ;AND ADDS ALSO EVERY TRANS QTY-USED TO THE MASTER'S FIELD AND
                ;SAVES IT IN USD-LST-WK FIELD
                ;THEN IT PROCEED WITH CHECKING THE AVAILABILITY OF THE CURRENT
                ;ITEM AND UPDATES THE QTY-ORDRD IF APPLICABLE.


0242 2AD408     UPDATE:  LHLD    NEXT            ;ADD TO MASTER'S QTY-USED
0245 112F00              LXI     D,47
0248 19                  DAD     D
0249 0E05                MVI     C,5             ;THE QTY-USED
024B 11300A              LXI     D,RECORD+47     ;FROM THE TRANS RECORD
024E CDAB02              CALL    ADDA
0251 2AD408              LHLD    NEXT
0254 112A00              LXI     D,42
0257 19                  DAD     D
0258 112B0A              LXI     D,RECORD+42
025B 0E05                MVI     C,5             ;NOW ADD QTY-AVLBL
025D CDAB02              CALL    ADDA
0260 21270A              LXI     H,RECORD+38
0263 010A00              LXI     B,10
0266 11400A              LXI     D,RECORD+63     ;MOVE QTY-USED TO
0269 CD8C05              CALL    MOVE            ;USD-LST-WK
026C 2AD408              LHLD    NEXT
026F 111F00              LXI     D,31
0272 19                  DAD     D
0273 11200A              LXI     D,RECORD+31
0276 1A                  LDAX    D
0277 BE                  CMP     M
0278 CA8F02              JZ      UPDAT2
```

```
027B FE59               CPI    'Y'
027D C28D02             JNZ    UPDAT1
0280 2AD408             LHLD   NEXT
0283 114900             LXI    D,73
0286 19                 DAD    D
0287 7E                 MOV    A,M
0288 3C                 INR    A
0289 77                 MOV    M,A
028A C38F02             JMP    UPDAT2

028D 364E      UPDAT1:  MVI    M,04EH

028F 2AD408    UPDAT2:  LHLD   NEXT
0292 112000             LXI    D,32
0295 19                 DAD    D
0296 11210A             LXI    D,RECORD+32
0299 1A                 LDAX   D
029A FE59               CPI    'Y'
029C C0                 RNZ
029D 3659               MVI    M,059H
029F 2AD408             LHLD   NEXT
02A2 112500             LXI    D,37
02A5 19                 DAD    D
02A6 0E05               MVI    C,5
02A8 11260A             LXI    D,RECORD+37

02AB 1A        ADDA:    LDAX   D
02AC FE20               CPI    20H
02AE C8                 RZ
02AF 7E                 MOV    A,M
02B0 FE20               CPI    20H
02B2 C2BA02             JNZ    ADDB
02B5 1A                 LDAX   D
02B6 77                 MOV    M,A
02B7 C3C402             JMP    DCRMNT
02BA 1A        ADDB:    LDAX   D
02BB 86                 ADD    M
02BC DE30               SBI    30H
02BE 77                 MOV    M,A
02BF DE39               SBI    39H
02C1 F4CB02             CP     CARRY
02C4 2B        DCRMNT:  DCX    H
02C5 1B                 DCX    D
02C6 0D                 DCR    C
02C7 C2AB02             JNZ    ADDA
02CA C9                 RET

02CB C8        CARRY:   RZ
02CC C62F               ADI    2FH
02CE 77                 MOV    M,A
02CF 2B                 DCX    H
02D0 7E                 MOV    A,M
02D1 FE20               CPI    20H
02D3 C2DA02             JNZ    ADDC
02D6 3631               MVI    M,31H
02D8 23                 INX    H
02D9 C9                 RET
02DA C601      ADDC:    ADI    01H
02DC 77                 MOV    M,A
02DD DE39               SBI    39H
02DF F4CB02             CP     CARRY
02E2 23                 INX    H
02E3 C9                 RET

02E4 7E        UPDAT3:  MOV    A,M
02E5 FE4E               CPI    'N'
02E7 C8                 RZ
02E8 11260A             LXI    D,RECORD+37

02EB 1A        COMPD:   LDAX   D
02EC FE30               CPI    '0'
```

```
02EE C0                 RNZ
02EF 1B                 DCX     D
02F0 1A                 LDAX    D

02F1 FE20               CPI     020H
02F3 C0                 RNZ
02F4 364E               MVI     M,04EH
02F6 C9                 RET

              ; GETS ONE MASTER RECORD

02F7 2AD408   GETM:     LHLD    NEXT
02FA 114A00             LXI     D,74
02FD 19                 DAD     D
02FE 7E                 MOV     A,M
02FF FE1A               CPI     CTRLZ
0301 CA0A03             JZ      OVER     ;EXIT IF END OF FILE
0304 22D408             SHLD    NEXT
0307 C30102             JMP     COMP     ;OTHERWISE MASTER HAS BEEN
                                         ;INCREMENTED SUCCESSFULLY

030A AF       OVER:     XRA     A
030B 3C                 INR     A
030C 324D0A             STA     FINFLG
030F CD0404             CALL    OK
0312 C9                 RET

              ; GETS ONE TRANS RECORD ONLY

0313 2ABA03   GETT:     LHLD    HSAVE
0316 224B0A             SHLD    BSAVE
0319 11010A             LXI     D,RECORD
031C 014A00             LXI     B,74
031F 7E                 MOV     A,M
0320 EE1A               XRI     CTRLZ
0322 CA9903             JZ      FIN
0325 7E       MOVREC:   MOV     A,M
0326 FE24               CPI     '$'
0328 CA7603             JZ      NXTBUF
032B 12                 STAX    D
032C 23                 INX     H
032D 13                 INX     D
032E 0B                 DCX     B
032F 79                 MOV     A,C
0330 B0                 ORA     B
0331 C22503             JNZ     MOVREC
0334 22BA03             SHLD    HSAVE
0337 7E                 MOV     A,M
0338 FE1A               CPI     CTRLZ
033A CA4503             JZ      FILEEND
033D 114A00             LXI     D,74
0340 19                 DAD     D
0341 7E                 MOV     A,M
0342 FE1A               CPI     CTRLZ
0344 C0                 RNZ
0345 3A500A   FILEEND:LDA       EOF
0348 3C                 INR     A
0349 32500A             STA     EOF
034C C9                 RET

034D 3A4E0A   OUTPUT:   LDA     DIVER
0350 FE00               CPI     00H
0352 C8                 RZ
0353 AF                 XRA     A
0354 324E0A             STA     DIVER
0357 3A4F0A             LDA     FLAG
035A FE00               CPI     00H
035C CA6903             JZ      GTBUF2
035F 11FF08             LXI     D,INBUFR1
0362 CDE603             CALL    PUTREC
0365 CDBC03             CALL    GETREC
```

```
0368 C9                      RET

0369 118009     GTBUF2:  LXI     D,INBUFR2
036C CDE603              CALL    PUTREC
036F 118009              LXI     D,INBUFR2
0372 CDBF03              CALL    GETR
0375 C9                  RET

0376 3A4E0A     NXTBUF:  LDA     DIVER
0379 3C                  INR     A
037A 324E0A              STA     DIVER
037D 3A4F0A              LDA     FLAG
0380 FE00                CPI     00H
0382 CA8F03              JZ      BUFR2
0385 AF                  XRA     A
0386 324F0A              STA     FLAG
0389 21FF08              LXI     H,INBUFR1
038C C32503              JMP     MOVREC

038F 3C         BUFR2:   INR     A
0390 324F0A              STA     FLAG
0393 218009              LXI     H,INBUFR2
0396 C32503              JMP     MOVREC


0399 3A4D0A     FIN:     LDA     FINFLG
039C 3C                  INR     A
039D 324D0A              STA     FINFLG
03A0 3A4F0A              LDA     FLAG
03A3 FE00                CPI     00H
03A5 CAB103              JZ      BFR1
03A8 118009              LXI     D,INBUFR2
03AB CDE903              CALL    PUTR
03AE 33                  INX     SP
03AF 33                  INX     SP
03B0 C9                  RET

03B1 11FF08     BFR1:    LXI     D,INBUFR1
03B4 CDE903              CALL    PUTR
03B7 33                  INX     SP
03B8 33                  INX     SP
03B9 C9                  RET

03BA            HSAVE:   DS      2



                ; GETS A RECORD FROM A DISK FILE
03BC 11FF08     GETREC:  LXI     D,INBUFR1
03BF 0E1A       GETR:    MVI     C,SDMAF
03C1 3A500A              LDA     EOF
03C4 FE00                CPI     00H
03C6 C0                  RNZ
03C7 CD0500              CALL    BDOS
03CA 11DB08              LXI     D,FCBSAV
03CD 0E14                MVI     C,READF
03CF CD0500              CALL    BDOS
03D2 FE00                CPI     BDAOK
03D4 C8                  RZ
03D5 FE01                CPI     BDER1
03D7 C8                  RZ
03D8 CD8407              CALL    REMSG
03DB C3C305              JMP     ERREX

03DE 3AFB08     DCRFCB:  LDA     FCBSAV+32
03E1 3D                  DCR     A
03E2 32FB08              STA     FCBSAV+32
03E5 C9                  RET

03E6 CDDE03     PUTREC:  CALL    DCRFCB

03E9 CDDE03     PUTR:    CALL    DCRFCB
```

```
03EC 0E1A                 MVI     C,SDMAF
03EE CD0500               CALL    BDOS
03F1 11DB08               LXI     D,FCBSAV
03F4 0E15                 MVI     C,WRITF
03F6 CD0500               CALL    BDOS
03F9 FE00                 CPI     BDAOK
03FB CA0404               JZ      OK
03FE CDA207               CALL    WEMSG
0401 C33707               JMP     PUTEX

0404 3A4D0A     OK:       LDA     FINFLG
0407 FE00                 CPI     00H
0409 CA1E04               JZ      INCFCB
040C CDDE07               CALL    CPDMA
040F 11DB08               LXI     D,FCBSAV
0412 0E10                 MVI     C,CLOSF
0414 CD0500               CALL    BDOS
0417 CD1C08               CALL    CCRLF
041A CDDE07               CALL    CPDMA
041D C9                   RET
041E 3AFB08     INCFCB:   LDA     FCBSAV+32
0421 3C                   INR     A
0422 32FB08               STA     FCBSAV+32
0425 C9                   RET

0426 2A4B0A     WRTREC:   LHLD    BSAVE
0429 EB                   XCHG
042A 21010A               LXI     H,RECORD
042D 014A00               LXI     B,74

0430 1A         WRTR:     LDAX    D
0431 FE24                 CPI     '$'
0433 CA4104               JZ      CHKBUF
0436 7E                   MOV     A,M
0437 12                   STAX    D
0438 23                   INX     H
0439 13                   INX     D
043A 0B                   DCX     B
043B 79                   MOV     A,C
043C B0                   ORA     B
043D C23004               JNZ     WRTR
0440 C9                   RET

0441 C5         CHKBUF:   PUSH    B
0442 017F09               LXI     B,INBUFR1+128
0445 7B                   MOV     A,E
0446 B9                   CMP     C
0447 CA5004               JZ      BUFTWO
044A 11FF08               LXI     D,INBUFR1
044D C35304               JMP     WRTR1

0450 118009     BUFTWO:   LXI     D,INBUFR2

0453 C1         WRTR1:    POP     B
0454 C33004               JMP     WRTR

0457 21FF08     PRNBUF:   LXI     H,INBUFR1
045A 116F04               LXI     D,STRING
045D 018000               LXI     B,128
0460 CD8C05               CALL    MOVE
0463 CD1C08               CALL    CCRLF
0466 0E09                 MVI     C,9H
0468 116F04               LXI     D,STRING
046B CD0500               CALL    BDOS
046E C9                   RET
046F            STRING:   DS      128
04EF 24                   DB      '$'
04F0 21010A     PRNREC:   LXI     H,RECORD
04F3 014A00     PRNREC1:  LXI     B,74
04F6 11A504               LXI     D,STRING+54
04F9 CD8C05               CALL    MOVE
```

```
04FC CD1C08             CALL    CCRLF
04FF 0E09               MVI     C,9H
0501 11A504             LXI     D,STRING+54
0504 CD0500             CALL    BDOS
0507 C9                 RET
0508 217F05     PRNFCB: LXI     H,SAVFCB
050B 11CE04             LXI     D,STRING+95
050E 012100             LXI     B,33

0511 CD8C05             CALL    MOVE
0514 CD1C08             CALL    CCRLF
0517 0E09               MVI     C,9H
0519 11CE04             LXI     D,STRING+95
051C CD0500             CALL    BDOS
051F CD1C08             CALL    CCRLF
0522 C9                 RET

0523 CD1C08     PRNMEM: CALL    CCRLF
0526 2AD408             LHLD    NEXT
0529 116F04     PRNM1:  LXI     D,STRING
052C 018000             LXI     B,128

052F CD8C05             CALL    MOVE
0532 E5                 PUSH    H
0533 0E09               MVI     C,9H
0535 116F04             LXI     D,STRING
0538 CD0500             CALL    BDOS
053B CD1C08             CALL    CCRLF
053E E1                 POP     H
053F C9                 RET


                ; OPENS TRANS FILES

0540 11DB08     OPENFL: LXI     D,FCBSAV
0543 0E0F               MVI     C,OPENF
0545 CD0500             CALL    BDOS
0548 FEFF               CPI     BDERR
054A C27705             JNZ     RETURN  ; OPEN OK
054D CD1908             CALL    TWOCR
0550 CD4F08             CALL    SPMSG
0553 2A2A2A2043         DB      '*** CANNOT FIND  : ',0
0567 CD6F07             CALL    SHOTFN
056A 3ADA08             LDA     NUMFILS
056D FE00               CPI     00H
056F CAC701             JZ      DONE
0572 33                 INX     SP
0573 33                 INX     SP
0574 C38F01             JMP     FINISH

0577 3ADA08     RETURN: LDA     NUMFILS
057A 3C                 INR     A
057B 32DA08             STA     NUMFILS
057E C9                 RET
                ; SAVES MASTER FCB FOR FILE OUTPUT

057F AF         SAVFCB: XRA     A
0580 32FB08             STA     FCBSAV+32
0583 216C00             LXI     H,TFCB+16
0586 011100             LXI     B,17
0589 11DB08             LXI     D,FCBSAV

058C 7E         MOVE:   MOV     A,M
058D 12                 STAX    D
058E 23                 INX     H
058F 13                 INX     D
0590 0B                 DCX     B
0591 79                 MOV     A,C
0592 B0                 ORA     B
0593 C28C05             JNZ     MOVE
0596 C9                 RET             ;FCB IS SAVE NOW
```

```
                ; READ A FILE FROM DISK INTO "BUFFER"

0597 21920A     GET:     LXI     H,BUFFER
059A 22D408              SHLD    NEXT
059D 115C00              LXI     D,TFCB
05A0 0E0F                MVI     C,OPENF
05A2 CD0500              CALL    BDOS
05A5 FEFF                CPI     BDERR
05A7 C2C905              JNZ     GET1
05AA CD1908              CALL    TWOCR
05AD CD4F08              CALL    SPMSG
05B0 2A2A2A2043          DB      '*** CANNOT FIND',0
05C0 CD3E07              CALL    SHOFN
05C3 CD1908     ERREX:   CALL    TWOCR
05C6 C3C701              JMP     DONE

05C9 AF         GET1:    XRA     A
05CA 32D208              STA     RECCT
05CD 2AD408     GET2:    LHLD    NEXT
05D0 EB                  XCHG
05D1 0E1A                MVI     C,SDMAF
05D3 CD0500              CALL    BDOS
05D6 115C00              LXI     D,TFCB
05D9 0E14                MVI     C,READF
05DB CD0500              CALL    BDOS
05DE FE00                CPI     BDAOK
05E0 CAEE05              JZ      GET3
05E3 FE01                CPI     BDER1
05E5 CA1E06              JZ      GETEX
05E8 CD8407              CALL    REMSG
05EB C3C305              JMP     ERREX

05EE 3AD208     GET3:    LDA     RECCT
05F1 3C                  INR     A
05F2 32D208              STA     RECCT
05F5 2AD408              LHLD    NEXT
05F8 118000              LXI     D,128
05FB 19                  DAD     D
05FC 22D408              SHLD    NEXT
05FF 3A0700              LDA     MEMAX
0602 3D                  DCR     A
0603 BC                  CMP     H
0604 C2CD05              JNZ     GET2
0607 CD1908              CALL    TWOCR
060A CD4F08              CALL    SPMSG
060D 4F5554204F          DB      'OUT OF MEMORY',0
061B C3C305              JMP     ERREX

061E CD1C08     GETEX:   CALL    CCRLF
0621 CDDE07              CALL    CPDMA
0624 2AD408              LHLD    NEXT      ;SAVE THE LAST VALUE IN
0627 22D808              SHLD    NXTSAV    ;NEXT IN NXTSAVE
062A 21920A              LXI     H,BUFFER
062D 110902              LXI     D,521
0630 19                  DAD     D         ;OFFSET NEXT BY 521 POSITIONS
0631 22D408              SHLD    NEXT      ;STARTING ADDRESS IN NEXT
0634 2AD408     PRPMSTR:LHLD     NEXT
0637 112600              LXI     D,38
063A 19                  DAD     D
063B 22D608              SHLD    NEXT1
063E 111900              LXI     D,25
0641 19                  DAD     D         ;ADDRSS OF AV-LST-WK IN H REG.
0642 EB                  XCHG              ;ADDRSS MOVED TO D REG.
0643 2AD608              LHLD    NEXT1     ;ADDRSS OF QTY-AVLBL IN H REG.
0646 010A00              LXI     B,10
0649 CD8C05              CALL    MOVE      ;QTY-AVLBL MOVED TO AV-LST-WK
064C C35A06              JMP     ZFILL
064F 4E20202020NZERO:    DB      'N     0     0'
065A 2AD408     ZFILL:   LHLD    NEXT
065D 112000              LXI     D,32      ;BEGINNING ADDRS OF ORDRD
0660 19                  DAD     D
0661 EB                  XCHG              ;MOVED TO D REG.
```

```
0662 010B00            LXI     B,11
0665 214F06            LXI     H,NZERO
0668 CD8C05            CALL    MOVE     ;MOVE CONTAINTS OF NZERO
                                        ;TO ORDRD AND AV & ORDRD
                                        ;QUANTITIES
066B 2AD408            LHLD    NEXT
066E 114A00            LXI     D,74     ;PREPARE FOR NEXT RECORD
0671 19                DAD     D
0672 22D408            SHLD    NEXT
0675 7E                MOV     A,M
0676 EE1A              XRI     CTRLZ
0678 C23406            JNZ     PRPMSTR
067B 21920A            LXI     H,BUFFER
067E 110902            LXI     D,521
0681 19                DAD     D
0682 22D408            SHLD    NEXT
0685 C9                RET

                ;WRITE A FILE FROM "BUFFER" TO DISK

0686 21920A     PUT:   LXI     H,BUFFER
0689 22D408            SHLD    NEXT
068C 3AD208            LDA     RECCT
068F 32D308            STA     CTSAV
0692 3A5C00            LDA     TFCB
0695 B7                ORA     A        ;IS IT LEGAL?
0696 C29F06            JNZ     PUT1
0699 CDC007            CALL    WROPN
069C C33707            JMP     PUTEX    ;AND TRY AGAIN

069F 0E0D       PUT1:  MVI     C,INITF
06A1 CD0500            CALL    BDOS
06A4 AF                XRA     A
06A5 327C00            STA     FCBCR    ;CURRENT RECORD
06A8 210000            LXI     H,0
06AB 226800            SHLD    FCBEX
06AE 226A00            SHLD    FCBS2
06B1 115C00            LXI     D,TFCB
06B4 0E11              MVI     C,FINDF
06B6 CD0500            CALL    BDOS
06B9 FEFF              CPI     BDERR
06BB CAE306            JZ      PUT2
06BE CD1C08            CALL    CCRLF
06C1 CD4F08            CALL    SPMSG
06C4 4F4B20544F        DB      'OK TO ERASE  ',0
06D2 CD3E07            CALL    SHOFN
06D5 CD5B08            CALL    GETYN
06D8 C23707            JNZ     PUTEX
06DB 115C00            LXI     D,TFCB
06DE 0E13              MVI     C,DELEF
06E0 CD0500            CALL    BDOS
06E3 115C00     PUT2:  LXI     D,TFCB
06E6 0E16              MVI     C,MAKEF
06E8 CD0500            CALL    BDOS
06EB FEFF              CPI     BDERR
06ED C2F606            JNZ     PUT3
06F0 CDC007            CALL    WROPN
06F3 C33707            JMP     PUTEX

06F6 2AD408     PUT3:  LHLD    NEXT
06F9 EB                XCHG
06FA 0E1A              MVI     C,SDMAF
06FC CD0500            CALL    BDOS
06FF 2AD408            LHLD    NEXT
0702 118000            LXI     D,128
0705 19                DAD     D
0706 22D408            SHLD    NEXT
0709 115C00            LXI     D,TFCB
070C 0E15              MVI     C,WRITF
070E CD0500            CALL    BDOS
0711 FE00              CPI     BDAOK
0713 CA1C07            JZ      PUT4
```

```
0716 CDA207                 CALL     WEMSG
0719 C33707                 JMP      PUTEX

071C 3AD208       PUT4:     LDA      RECCT
071F 3D                     DCR      A
0720 32D208                 STA      RECCT
0723 C2F606                 JNZ      PUT3
0726 CDDE07                 CALL     CPDMA
0729 115C00                 LXI      D,TFCB
072C 0E10                   MVI      C,CLOSF
072E CD0500                 CALL     BDOS
0731 3AD308                 LDA      CTSAV
0734 32D208                 STA      RECCT
0737 CD1C08       PUTEX:    CALL     CCRLF
073A CDDE07                 CALL     CPDMA
073D C9                     RET
                  ; DISPLAY FILENAME.TYP FROM TRANSIENT FCB
073E C5           SHOFN:    PUSH     B
073F E5                     PUSH     H
0740 3A6500                 LDA      FCBTY
0743 4F                     MOV      C,A
0744 AF                     XRA      A
0745 326500                 STA      FCBTY
0748 326800                 STA      FCBEX
074B 215C00                 LXI      H,TFCB
074E 7E                     MOV      A,M
074F E60F                   ANI      0FH
0751 F640                   ORI      40H
0753 CD0C08                 CALL     CO
0756 3E3A                   MVI      A,':'
0758 CD0C08                 CALL     CO
075B 23                     INX      H
075C CD2608                 CALL     COMSG
075F 79                     MOV      A,C
0760 216500                 LXI      H,FCBTY
0763 77                     MOV      M,A
0764 3E2E                   MVI      A,'.'
0766 CD0C08                 CALL     CO
0769 CD2608                 CALL     COMSG
076C E1                     POP      H
076D C1                     POP      B
076E C9                     RET

                  ;DISPLAY TRANS FILENAME
076F 0608         SHOTFN:   MVI      B,8
0771 21DC08                 LXI      H,FCBSAV+1
0774 7E           TFN:      MOV      A,M
0775 CD0C08                 CALL     CO
0778 23                     INX      H
0779 05                     DCR      B
077A 78                     MOV      A,B
077B FE00                   CPI      0
077D C27407                 JNZ      TFN
0780 CD1C08                 CALL     CCRLF
0783 C9                     RET

                  ; DISPLAY READ ERROR MESSAGE

0784 CD1908       REMSG:    CALL     TWOCR
0787 CD4F08                 CALL     SPMSG
078A 5045524D41             DB       'PERMANENT READ ERROR',CR,LF,0
07A1 C9                     RET

                  ; DISPLAY WRITE ERROR MESSAGE
07A2 CD1908       WEMSG:    CALL     TWOCR
07A5 CD4F08                 CALL     SPMSG
07A8 50524D414E             DB       'PRMANENT WRITE ERROR',CR,LF,0
07BF C9                     RET

                  ;DISPLAY WRITE OPEN ERROR MESSAGE
07C0 CD1908       WROPN:    CALL     TWOCR
07C3 CD4F08                 CALL     SPMSG
```

```
07C6 43414E4E4F           DB      'CANNOT OPEN FOR WRIT',CR,LF,0
07DD C9                   RET

                ;RESTORE CP/M DMA ADDRESS TO THE TRANSIENT BUFFER
07DE 118000     CPDMA:    LXI     D,TBUFF
07E1 0E1A                 MVI     C,SDMAF
07E3 CD0500               CALL    BDOS
07E6 C9                   RET

                ; GET A VALID DRIVE SELECT DESIGNATOR
07E7 CD3008     DRSEL:    CALL    CIMSG
07EA 3A8008               LDA     INBUF+2
07ED E65F                 ANI     01011111B
07EF D640                 SUI     '@'
07F1 FAFC07               JM      DRERR
07F4 D611                 SUI     17
07F6 F2FC07               JP      DRERR
07F9 C611                 ADI     17
07FB C9                   RET
07FC AF         DRERR:    XRA     A
07FD C9                   RET
                ;CONSOLE CHARACTER INTO REGISTER A MASKED TO 7 BITS
07FE C5         CI:       PUSH    B         ;SAVE RIGISTERS
07FF D5                   PUSH    D
0800 E5                   PUSH    H
0801 0E01                 MVI     C,RCONF   ;READ FUNCTION
0803 CD0500               CALL    BDOS
0806 E67F                 ANI     7FH       ;MASK TO 7 BITS
0808 E1                   POP     H         ;RESTORE REGISTERS
0809 D1                   POP     D
080A C1                   POP     B
080B C9                   RET

                ;CHARACTER IN REGISTER A OUTPUT TO CONSOLE
080C C5         CO:       PUSH    B

080D D5                   PUSH    D
080E E5                   PUSH    H
080F 0E02                 MVI     C,WCONF   ;SELECT FUNCTION
0811 5F                   MOV     E,A       ;CHARACTER TO E
0812 CD0500               CALL    BDOS      ;OUTPUT BY CP/M
0815 E1                   POP     H
0816 D1                   POP     D
0817 C1                   POP     B
0818 C9                   RET
0819 CD1C08     TWOCR:    CALL    CCRLF

                ; CARRIAGE RETURN, LINE FEED TO CONSOLE
081C 3E0D       CCRLF:    MVI     A,CR
081E CD0C08               CALL    CO
0821 3E0A                 MVI     A,LF
0823 C30C08               JMP     CO

                ; MESSAGE POINTED TO BY HL OUT TO CONSOLE
0826 7E         COMSG:    MOV     A,M
0827 B7                   ORA     A
0828 C8                   RZ
0829 CD0C08               CALL    CO
082C 23                   INX     H
082D C32608               JMP     COMSG

                ; INPUT CONSOLE MESSAGE INTO BUFFER
0830 C5         CIMSG:    PUSH    B
0831 D5                   PUSH    D
0832 E5                   PUSH    H
0833 217F08               LXI     H,INBUF+1         ;ZERO CHARACTER COUNTER
0836 3600                 MVI     M,0
0838 2B                   DCX     H         ;SET MAXIMUM LINE LENGTH
0839 3650                 MVI     M,80
083B EB                   XCHG              ;INBUF POINTER TO DE REGISTERS
083C 0E0A                 MVI     C,RBUFF   ;SET UP READ BUFFER FUNCTION
083E CD0500               CALL    BDOS      ;INPUT A LINE
```

```
0841 217F08             LXI     H,INBUF+1           ;GET CHRACTER COUNTER
0844 5E                 MOV     E,M
0845 1600               MVI     D,0         ;ZERO MSB
0847 19                 DAD     D
0848 23                 INX     H
0849 3600               MVI     M,0
084B E1                 POP     H
084C D1                 POP     D
084D C1                 POP     B
084E C9                 RET

                ; MESSAGE POINTED TO BY STACK OUT TO CONSOLE
084F E3         SPMSG:  XTHL
0850 AF                 XRA     A
0851 86                 ADD     M
0852 23                 INX     H
0853 E3                 XTHL
0854 C8                 RZ
0855 CD0C08             CALL    CO
0858 C34F08             JMP     SPMSG

                ; GET YES OR NO FROM CONSOLE
085B CD4F08     GETYN:  CALL    SPMSG
085E 2028592F4E         DB      ' (Y/N)?: ',0
0868 CD3008             CALL    CIMSG
086B CD1C08             CALL    CCRLF
086E 3A8008             LDA     INBUF+2
0871 E65F               ANI     01011111B
0873 FE59               CPI     'Y'
0875 C8                 RZ
0876 FE4E               CPI     'N'
0878 C25B08             JNZ     GETYN
087B FE00               CPI     0
087D C9                 RET


                ; RAM VARIABLES AND BUFFERS
087E            INBUF:  DS      83          ;LINE INPUT BUFFER
08D1            DRSAV:  DS      1           ;CURRENT DRIVE AT ENTRY
08D2            RECCT:  DS      1           ;TOTAL RECORDS READ/TO WRITE
08D3            CTSAV:  DS      1           ;SAVE LOCATION FOR COUNTER
08D4            NEXT:   DS      2           ;NEXT DMA ADDRESS
08D6            NEXT1:  DS      2
08D8            NXTSAV: DS      2           ;SAVE NEXT: FOR OUTPUT
08DA            NUMFILS:DS      1           ;DENOTE NUMB. OF TRANS FILES PROCESSED
08DB            FCBSAV: DS      36
08FF            INBUFR1:DS      128         ;INPUT BUFFER FROM TRANS
097F 24                 DB      '$'
0980            INBUFR2:DS      128         ;ALTERNAT BUFFER FOR TRANS
0A00 24                 DB      '$'
0A01            RECORD: DS      74
0A4B            BSAVE:  DS      2
0A4D            FINFLG: DS      1
0A4E            DIVER:  DS      1
0A4F            FLAG:   DS      1
0A50            EOF:    DS      1           ;SET TO ONE WHEN LAST BUFFR READ IN

                ; SET UP STACK SPACE
0A51                    DS      64          ;40H LOCATIONS
0A91 00         STAK:   DB      0           ;TOP OF STACK

                ; FROM HERE THROUGH CCP IS BUFFER SPACE
                BUFFER:
0A92                    END
```

Appendix C

Demonstration run on one file for INVUPDT.CMD

```
USE INV-1111
. DISP STRU
STRUCTURE FOR FILE:  INV-1111.DBF
NUMBER OF RECORDS:   00026
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	NAME	C	020	
002	CODE	C	010	
003	AVAILABLE	C	001	
004	ORDRD	C	001	
005	QTY:ORDRD	N	005	
006	QTY:AVLBL	N	005	
007	QTY:USED	N	005	
008	QTY:PLAND	N	005	
009	UNITS	C	010	
010	AV:LST:WK	N	005	
011	USD:LST:WK	N	005	
012	FLAG	N	001	
** TOTAL **			00074	

00001	CAP	10004500	Y	N	0	80	19	480	NUMBER	90	9	
00002	HINGE	11021122	Y	N	0	20	9	370	NUMBER	25	4	
00003	BRICK	11280670	Y	N	0	9	10	129	LOADS	10	9	
00004	CABLE	11397278	Y	N	0	100	60	2000	FT	110	50	
00005	CABLE	11397279	Y	N	0	39	9	267	FT	43	5	
00006	GLUE	21113990	Y	N	0	10	6	120	POUND	13	3	
00007	GLUE	21113993	Y	N	0	15	10	50	POUND	20	5	
00008	ASPHALT	22261100	Y	N	0	1	3	20	TON	2	2	
00009	ASPHALT	22261103	Y	N	0	3	1	24	TON	2	2	
00010	CONCRETE	22278000	Y	N	0	2	1	30	TON	3	0	
00011	CEMENT	22299900	Y	N	0	6	2	30	TON	7	1	
00012	CEMENT	22299901	Y	N	0	1	4	10	TON	2	3	
00013	GLASS	22444500	Y	N	0	40	7	260	SQR FT	44	3	
00014	PIPE	62661100	Y	N	0	98	25	500	FT	103	20	
00015	WIRE	62661122	N	Y	1000	0	0	18000	FT	0	0	
00016	PLASTER	63334000	Y	N	0	20	6	100	POUND	23	3	
00017	PLASTER	63334001	Y	N	0	10	20	50	POUND	20	10	
00018	ASBESTOS	63335000	Y	N	0	30	10	40	SHEET	35	5	
00019	ASBESTOS	63335002	Y	N	0	10	5	20	SHEET	13	2	
00020	PAINT	63336400	N	N	0	0	0	190	CAN	0	0	
00021	CLAY	63337000	Y	N	0	30	10	100	CAN	35	5	
00022	BOLT	77001000	Y	N	0	100	20	300	NUMBER	110	10	
00023	CONNECTOR	77002000	Y	N	0	230	60	400	NUMBER	240	50	
00024	NAIL	77003000	Y	N	0	100	100	5000	NUMBER	150	50	
00025	SCREW	77004000	Y	N	0	100	20	5000	NUMBER	105	15	
00026	NUT	77005000	Y	N	0	100	70	5000	NUMBER	120	50	

.

```
A>DBASE

ENTER TODAYS DATE  AS MM/DD/YY        OR RETURN FOR NONE  : 11/24/85

***  dBASE II     Ver 2.3B   22 FEB 82
. DO INVUPDT
*****************************************
* THIS IS THE INVENTORY UPDATING PROGRAM *
* YOU CAN RUN IT ANY TIME AND ANY DAY    *
* IT WILL OPERATE ON THE INV- FILES      *
* YOU WILL HAVE TO SUPPLY THE NAME OF    *
* THE FILE AND THE INDEX KEY---THIS      *
* KEY IS THE FIELD YOU WILL USE TO       *
* SELECT THE RECORD TO BE UPDATED        *
*****************************************


ENTER FILE NAME:INV-1111

ENTER INDEX KEY NAME:CODE

===>===>
THIS PROGRAM WILL OPERATE ON INV-1111
TO PERFORM THE FOLLOWING:

         (1) UPDATE THE QUANTITY USED
         (2) ENTER ANY QUANTITY RECEIVED
         (3) REGISTER ANY QUANTITY ORDERED
         (4) REPORTING FACILITY


ENTER OPTION NUMBER:1
```

```
===>
This option will subtract the used quantity from QTY:AVLBL
Updating QTY:USED and checking if we have exceeded the Plan
IF the available quantity becomes zero
THEN the AVAILABLE field is set to (N)


ENTER KEY (NUMBER/NAME)....:211139900
NO FIND
** requested record does not exist **
TRY AGAIN...(Y/N)?:2

ANOTHER USED ITEM....(Y/N)? :Y
ENTER KEY (NUMBER/NAME)....:224445000
NO FIND
** requested record does not exist **
TRY AGAIN...(Y/N)?:Y
ENTER KEY (NUMBER/NAME)....:22444500

ENTER QUANTITY USED...:10



00013  GLASS                22444500   Y N     0     30     17     260 SQR F]

ANOTHER USED ITEM....(Y/N)? :Y
ENTER KEY (NUMBER/NAME)....:77005000

ENTER QUANTITY USED...:500



*** error in quantity: updating cannot proceed ***
Quantity Available Becomes Negative

TRY AGAIN...(Y/N)?:Y
ENTER KEY (NUMBER/NAME)....:77005000

ENTER QUANTITY USED...:50



00026  NUT                  77005000   Y N     0     50     120    5000 NUMBER

ANOTHER USED ITEM....(Y/N)? :Y
ENTER KEY (NUMBER/NAME)....:N
NO FIND
** requested record does not exist **
TRY AGAIN...(Y/N)?:N

ANOTHER USED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:Y

WRONG OPTION NUMBER

DO YOU WISH TO CONTINUE WITH INVUPDT
(Y/N)? :Y

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:1

```
===>
This option will subtract the used quantity from QTY:AVLBL
Updating QTY:USED and checking if we have exceeded the Plan
IF the available quantity becomes zero
THEN the AVAILABLE field is set to (N)



ENTER KEY (NUMBER/NAME)....:Y
NO FIND
** requested record does not exist **
TRY AGAIN...(Y/N)?:N

ANOTHER USED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:2

```
===>
This option will subtract the value of the
received quantity from the ordered one
and add it to the available quantity

ENTER KEY (CODE/NAME):62661122

ENTER QUANTITY RECEIVED:200

00015  WIRE                      62661122    Y Y    800     200       0  18000 FT

ANOTHER RECEIVED ITEM...(Y/N)? :62661100

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:2

```
===>
This option will subtract the value of the
received quantity from the ordered one
and add it to the available quantity

ENTER KEY (CODE/NAME):62661100

ENTER QUANTITY RECEIVED:100

*** error in quantity updating cannot proceed ***
Quantity Received Exceeded The Ordered Quantity

MORE INFORMATION (Y/N)? :Y
This may be wrong and therefore would not be updated
by this program. A manual procedure is used in these cases
to insure that the data is properly checked before proceeding
with the updating

There are two possible reasons why this data is wrong:
   (1) The Consignment is Meant for Another Contract But Landed
       Here by Mistake
   (2) Your Calculation is Wrong

MORE ON ANY OF THESE POSSIBILITIES..(Y/N)? :Y


ENTER POSSIBILITY NUMBER:1

The delivery papers may be correct but have accompanied the wrong
consignment which means that your consignment may be on its
way to another place.

ENTER (1) OR (2) TO REVIEW THE PREVIOUS POSSIBILITIES
ANY OTHER NO. WILL EXIT YOU FROM HERE:2

Your calculation may be wrong because different items are
actually packed together, so make sure first that all the
items counted are of the same type
But if you have been using the figures from the delivery
papers then the papers might be wrong, while the actual
quantity received is  correct, SO CHECK ACCORDINGLY

MORE ON HOW TO CHECK THE PAPERS..(Y/N)? :Y

......A PROCEDURE FOR CHECKING PAPERS.....

ENTER (1) OR (2) TO REVIEW THE PREVIOUS POSSIBILITIES
ANY OTHER NO. WILL EXIT YOU FROM HERE:7
```

IF the consignment is definitely for you
and your calculations are correct,THEN
You may use this item*** PROVIDED THAT:
    (1) You have ordered the same item before
    (2) Your starting date to use this item is DUE

MORE ON AUTHORIZED USE..(Y/N)? :Y

Send a Message to the Upper Rank Containing :
 1-The Quantity of the Received Item
 2-Your Ordered Quantity
 3-Name and Code of the Item

 And Wait for One Week
IF there will be NO OBJECTION
THEN and only then you can use it

MORE ON MANUAL UPDATE...(Y/N)? :Y

.....A MANUAL PROCEDURE.....

DO YOU WANT TO ADD ANY PROCEDURE:Y

YOU MAY ADD ANY USEFUL INFORMATION IN HERE
BY FOLLOWING THIS PROCEDURE:
......A DESCRIPTIVE DETAIL ON......
......HOW TO ADD THE DATA...........

DO YOU WISH TO REVIEW THIS INFORMATION AGAIN
...(Y/N)?:Y
   (1) The Consignment is Meant for Another Contract But Landed
       Here by Mistake
   (2) Your Calculation is Wrong

MORE ON ANY OF THESE POSSIBILITIES..(Y/N)? :N

You may use this item*** PROVIDED THAT:
    (1) You have ordered the same item before
    (2) Your starting date to use this item is DUE

MORE ON AUTHORIZED USE..(Y/N)? :Y

Send a Message to the Upper Rank Containing :
 1-The Quantity of the Received Item
 2-Your Ordered Quantity
 3-Name and Code of the Item

 And Wait for One Week
IF there will be NO OBJECTION
THEN and only then you can use it

MORE ON MANUAL UPDATE...(Y/N)? :N

```
DO YOU WANT TO ADD ANY PROCEDURE:N

DO YOU WISH TO REVIEW THIS INFORMATION AGAIN
...(Y/N)?:N
TRY THIS ITEM AGAIN...(Y/N)?:N

ANOTHER RECEIVED ITEM...(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:3

```
===>
This option will add the ordered quantity
to the QTY:ORDRD field and set
the ORDRD field to (Y)

ENTER KEY (CODE/NAME):22299900

ENTER QUANTITY ORDERED:29

00011   CEMENT                22299900   Y Y   29   6   2   30 TON

ANOTHER ORDERED ITEM....(Y/N)? :Y

ENTER KEY (CODE/NAME):22299901

ENTER QUANTITY ORDERED:20

00012   CEMENT                22299901   Y Y   20   1   4   10 TON

ANOTHER ORDERED ITEM....(Y/N)? :11280670

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:3

```
===>
This option will add the ordered quantity
to the QTY:ORDRD field and set
the ORDRD field to (Y)

ENTER KEY (CODE/NAME):22299900

ENTER QUANTITY ORDERED:0

00011   CEMENT                 22299900   Y Y    29     6     2     30 TON

ANOTHER ORDERED ITEM....(Y/N)? :Y

ENTER KEY (CODE/NAME):11280670

ENTER QUANTITY ORDERED:5

00003   BRICK                  11280670   Y Y     5     9    10    129 LOADS

ANOTHER ORDERED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:1

```
===>
This option will subtract the used quantity from QTY:AVLBL
Updating QTY:USED and checking if we have exceeded the Plan
IF the available quantity becomes zero
THEN the AVAILABLE field is set to (N)


ENTER KEY (NUMBER/NAME)....:211139900
NO FIND
** requested record does not exist **
TRY AGAIN...(Y/N)?:Y
ENTER KEY (NUMBER/NAME)....:21113993

ENTER QUANTITY USED...:15


00007   GLUE                  21113993   N N     0     0    25    50 POUND

ANOTHER USED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:3

```
===>
This option will add the ordered quantity
to the QTY:ORDRD field and set
the ORDRD field to (Y)

ENTER KEY (CODE/NAME):22278000

ENTER QUANTITY ORDERED:10

00010  CONCRETE              22278000   Y Y    10     2     1    30 TON

ANOTHER ORDERED ITEM....(Y/N)? :Y

ENTER KEY (CODE/NAME):633340000
NO FIND
** requested record does not exist **
TRY AGAIN (Y/N)?:Y

ENTER KEY (CODE/NAME):63334000

ENTER QUANTITY ORDERED:200

00016  PLASTER               63334000   Y Y    200    20    6    100 POUND

ANOTHER ORDERED ITEM....(Y/N)? :Y

ENTER KEY (CODE/NAME):22299901

ENTER QUANTITY ORDERED:10

00012  CEMENT                22299901   Y Y    30     1     4    10 TON

ANOTHER ORDERED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

```
            (1) UPDATE THE QUANTITY USED
            (2) ENTER ANY QUANTITY RECEIVED
            (3) REGISTER ANY QUANTITY ORDERED
            (4) REPORTING FACILITY

ENTER OPTION NUMBER:2
```

```
===>
This option will subtract the value of the
received quantity from the ordered one
and add it to the available quantity

ENTER KEY (CODE/NAME):63334000

ENTER QUANTITY RECEIVED:180

00016   PLASTER              63334000   Y Y    20    200     6    100 POUND

ANOTHER RECEIVED ITEM...(Y/N)? :Y

ENTER KEY (CODE/NAME):22299901

ENTER QUANTITY RECEIVED:20

00012   CEMENT               22299901   Y Y    10     21     4     10 TON

ANOTHER RECEIVED ITEM...(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UP ATE THE QUANTITY U ED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:1

```
===>
This option will subtract the used quantity from QTY:AVLBL
Updating QTY:USED and checking if we have exceeded the Plan
IF the available quantity becomes zero
THEN the AVAILABLE field is set to (N)


ENTER KEY (NUMBER/NAME)....:22299901

ENTER QUANTITY USED...:20


!!! quantity used exceeded Plan !!!

00012   CEMENT              22299901   Y Y    10    1    24    10 TON

ANOTHER USED ITEM....(Y/N)? :Y
ENTER KEY (NUMBER/NAME)....:63334000

ENTER QUANTITY USED...:200


!!! quantity used exceeded Plan !!!

00016   PLASTER             63334000   N Y    20    0   206   100 POUND

ANOTHER USED ITEM....(Y/N)? :N

DO YOU WISH TO CONTINUE WITH INVUPDT
   (Y/N)? :Y
```

(1) UPDATE THE QUANTITY USED
(2) ENTER ANY QUANTITY RECEIVED
(3) REGISTER ANY QUANTITY ORDERED
(4) REPORTING FACILITY

ENTER OPTION NUMBER:4

```
===>
THIS OPTION  PROVIDES A REPORTING FACILITY
ON THE FOLLOWING:

          (1) EVERY NON-AVAILABLE ITEM
          (2) EVERY ORDERED ITEM
          (3) EVERY ITEM EXCEEDED THE PLANNED LIMIT
          (4) EVERY ITEM WHICH RAN OUT OF STOCK
              AND HAS NOT BEEN ORDERED YET

ENTER OPTION NUMBER:1
```

```
THE NON-AVAILABLE ITEMS ARE:
..........ITEM NAME.........CODE........
00007   GLUE                21113993
00016   PLASTER             63334000
00020   PAINT               63336400

DO YOU WISH TO CONTINUE WITH REPORTER
   (Y/N)? :Y
```

```
(1) EVERY NON-AVAILABLE ITEM
(2) EVERY ORDERED ITEM
(3) EVERY ITEM EXCEEDED THE PLANNED LIMIT
(4) EVERY ITEM WHICH RAN OUT OF STOCK
    AND HAS NOT BEEN ORDERED YET

ENTER OPTION NUMBER:2
```

```
THE ORDERED ITEMS ARE:
..........ITEM NAME.........CODE......ORDRD....PLAND...
00003   BRICK               11280670        5     129
00010   CONCRETE            22278000       10      30
00011   CEMENT              22299900       29      30
00012   CEMENT              22299901       10      10
00015   WIRE                62661122      800   18000
00016   PLASTER             63334000       20     100

DO YOU WISH TO CONTINUE WITH REPORTER
   (Y/N)? :Y
```

```
(1) EVERY NON-AVAILABLE ITEM
(2) EVERY ORDERED ITEM
(3) EVERY ITEM EXCEEDED THE PLANNED LIMIT
(4) EVERY ITEM WHICH RAN OUT OF STOCK
    AND HAS NOT BEEN ORDERED YET

ENTER OPTION NUMBER:3
```

```
THE ITEMS THAT EXCEEDED THEIR PLANNED USE ARE:
..........ITEM NAME.........CODE......USED.....PLAND...
00012  CEMENT                22299901      24      10
00016  PLASTER               63334000     206     100

DO YOU WISH TO CONTINUE WITH REPORTER
   (Y/N)? :Y
```

```
          (1) EVERY NON-AVAILABLE ITEM
          (2) EVERY ORDERED ITEM
          (3) EVERY ITEM EXCEEDED THE PLANNED LIMIT
          (4) EVERY ITEM WHICH RAN OUT OF STOCK
              AND HAS NOT BEEN ORDERED YET

 ENTER OPTION NUMBER:4
```

```
THE ITEMS THAT ARE NOT AVAILABLE AND NOT ORDERED YET ARE:
..........ITEM NAME.........CODE....
00007  GLUE                 21113993
00020  PAINT                63336400

DO YOU WISH TO CONTINUE WITH REPORTER
   (Y/N)? :N
```

```
DO YOU WISH TO CONTINUE WITH INVUPDT
    (Y/N)? :N


. QUIT
*** END RUN       dBASE II       ***
```

**********INV-1111**********

00001	CAP	10004500	Y N	0	80	19	480	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	9	370	NUMBER	25	4
00003	BRICK	11280670	Y Y	5	9	10	129	LOADS	10	9
00004	CABLE	11397278	Y N	0	100	60	2000	FT	110	50
00005	CABLE	11397279	Y N	0	39	9	267	FT	43	5
00006	GLUE	21113990	Y N	0	10	6	120	POUND	13	3
00007	GLUE	21113993	N N	0	0	25	50	POUND	20	5
00008	ASPHALT	22261100	Y N	0	1	3	20	TON	2	2
00009	ASPHALT	22261103	Y N	0	3	1	24	TON	2	2
00010	CONCRETE	22278000	Y Y	10	2	1	30	TON	3	0
00011	CEMENT	22299900	Y Y	29	6	2	30	TON	7	1
00012	CEMENT	22299901	Y Y	10	1	24	10	TON	2	3
00013	GLASS	22444500	Y N	0	30	17	260	SQR FT	44	3
00014	PIPE	62661100	Y N	0	98	25	500	FT	103	20
00015	WIRE	62661122	Y Y	800	200	0	18000	FT	0	0
00016	PLASTER	63334000	N Y	20	0	206	100	POUND	23	3
00017	PLASTER	63334001	Y N	0	10	20	50	POUND	20	10
00018	ASBESTOS	63335000	Y N	0	30	10	40	SHEET	35	5
00019	ASBESTOS	63335002	Y N	0	10	5	20	SHEET	13	2
00020	PAINT	63336400	N N	0	0	0	190	CAN	0	0
00021	CLAY	63337000	Y N	0	30	10	100	CAN	35	5
00022	BOLT	77001000	Y N	0	100	20	300	NUMBER	110	10
00023	CONNECTOR	77002000	Y N	0	230	60	400	NUMBER	240	50
00024	NAIL	77003000	Y N	0	100	100	5000	NUMBER	150	50
00025	SCREW	77004000	Y N	0	100	20	5000	NUMBER	105	15
00026	NUT	77005000	Y N	0	50	120	5000	NUMBER	120	50

Appendix D

Demonstration run on 21 files for INVUPDT.COM

a) contents of all files.

b) updating run at Rank 3.

c) contents of R3 and R4 files (after the updating).

d) updating run at Rank 2.

e) contents of R2 and R3 files.

f) updating run at Rank 1.

g) contents of R1 and R2 files.

***********INV-1111*********

00001	CAP	10004500	Y N	0	80	19	480	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	9	370	NUMBER	25	4
00003	BRICK	11280670	Y Y	5	9	10	129	LOADS	10	9
00004	CABLE	11397278	Y N	0	100	60	2000	FT	110	50
00005	CABLE	11397279	Y N	0	39	9	267	FT	43	5
00006	GLUE	21113990	Y N	0	10	6	120	POUND	13	3
00007	GLUE	21113993	N N	0	0	25	50	POUND	20	5
00008	.'SPHALT	22261100	Y N	0	1	3	20	TON	2	2
00009	ASPHALT	22261103	Y N	0	3	1	24	TON	2	2
00010	CONCRETE	22278000	Y Y	10	2	1	30	TON	3	0
00011	CEMENT	22299900	Y Y	29	6	2	30	TON	7	1
00012	CEMENT	22299901	Y Y	10	1	24	10	TON	2	3
00013	GLASS	22444500	Y N	0	30	17	260	SQR FT	44	3
00014	PIPE	62661100	Y N	0	98	25	500	FT	103	20
00015	WIRE	62661122	Y Y	800	200	0	18000	FT	0	0
00016	PLASTER	63334000	N Y	20	0	206	100	POUND	23	3
00017	PLASTER	63334001	Y N	0	10	20	50	POUND	20	10
00018	ASBESTOS	63335000	Y N	0	30	10	40	SHEET	35	5
00019	ASBESTOS	63335002	Y N	0	10	5	20	SHEET	13	2
00020	PAINT	63336400	N N	0	0	0	190	CAN	0	0
00021	CLAY	63337000	Y N	0	30	10	100	CAN	35	5
00022	BOLT	77001000	Y N	0	100	20	300	NUMBER	110	10
00023	CONNECTOR	77002000	Y N	0	230	60	400	NUMBER	240	50
00024	NAIL	77003000	Y N	0	100	100	5000	NUMBER	150	50
00025	SCREW	77004000	Y N	0	100	20	5000	NUMBER	105	15
00026	NUT	77005000	Y N	0	50	120	5000	NUMBER	120	50

**********INV-1114**********

00001	CAP	10004500	Y N	0	70	200	400	NUMBER	100	170
00002	HINGE	11021122	Y N	0	20	90	200	NUMBER	50	60
00003	BRICK	11280670	Y N	0	9	40	120	LOADS	20	29
00004	CABLE	11397278	Y N	0	100	600	2500	FT	200	500
00005	CABLE	11397279	Y N	0	39	90	300	FT	45	84
00006	GLUE	21113990	N N	0	0	70	120	POUND	0	70
00007	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00008	CONCRETE	22278000	Y N	0	2	1	30	TON	2	1
00009	CEMENT	22299900	Y N	0	6	2	30	TON	7	1
00010	CEMENT	22299901	Y N	0	1	4	10	TON	2	3
00011	GLASS	22444500	Y N	0	40	70	300	SQR FT	50	60
00012	PIPE	62661100	Y N	0	98	250	500	FT	108	240
00013	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y N	0	20	60	200	POUND	30	50
00015	PLASTER	63334001	Y N	0	10	20	50	POUND	15	15
00016	ASBESTOS	63335000	Y N	0	30	10	40	SHEET	35	5
00017	ASBESTOS	63335002	Y N	0	10	5	20	SHEET	10	5
00018	PAINT	63336400	N N	0	0	0	190	CAN	0	0
00019	SCREW	77004000	Y N	0	100	200	4000	NUMBER	200	100
00020	NUT	77005000	Y N	0	100	700	2000	NUMBER	500	300

**********INV-1112**********

00001	CAP	10004500	Y N	0	80	19	400	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	9	310	NUMBER	25	4
00003	GLUE	21113990	Y N	0	10	6	150	POUND	13	3
00004	GLUE	21113993	Y N	0	15	2	50	POUND	17	0
00005	ASPHALT	22261100	Y N	0	1	3	20	TON	3	1
00006	ASPHALT	22261103	Y N	0	3	1	24	TON	3	1
00007	CONCRETE	22278000	Y Y	10	2	1	30	TON	3	0
00008	CEMENT	22299900	Y N	0	6	2	30	TON	7	1
00009	CEMENT	22299901	Y Y	5	1	4	10	TON	4	1
00010	GLASS	22444500	Y N	0	40	7	300	SQR FT	42	5
00011	PIPE	62661100	Y N	0	98	25	500	FT	108	15
00012	WIRE	62661122	Y Y	500	500	0	18000	FT	0	0
00013	PLASTER	63334000	Y N	0	20	60	100	POUND	25	55
00014	PLASTER	63334001	Y N	0	10	20	50	POUND	15	15
00015	ASBESTOS	63335000	Y N	0	30	10	100	SHEET	35	5
00016	ASBESTOS	63335002	Y N	0	10	5	20	SHEET	11	4
00017	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00018	CLAY	63337000	Y N	0	30	50	100	CAN	35	45
00019	BOLT	77001000	Y N	0	100	20	300	NUMBER	105	15
00020	CONNECTOR	77002000	Y N	0	230	60	400	NUMBER	240	50
00021	SCREW	77004000	Y N	0	100	200	4000	NUMBER	110	190
00022	NUT	77005000	Y N	0	100	700	7000	NUMBER	105	695

**********INV-1115**********

00001	CAP	10004500	Y N	0	80	19	400	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	9	350	NUMBER	25	4
00003	BRICK	11280670	Y N	0	9	10	120	LOADS	10	9
00004	CABLE	11397278	Y N	0	80	120	2100	FT	100	100
00005	CABLE	11397279	Y N	0	39	90	250	FT	49	80
00006	GLUE	21113990	Y N	0	10	60	120	POUND	40	30
00007	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00008	CONCRETE	22278000	N N	0	0	3	30	TON	1	2
00009	CEMENT	22299900	Y N	0	6	2	30	TON	7	1
00010	CEMENT	22299901	Y N	0	1	4	10	TON	2	3
00011	GLASS	22444500	Y N	0	40	70	200	SQR FT	70	40
00012	PIPE	62661100	Y N	0	98	25	500	FT	110	13
00013	WIRE	62661122	N Y	0	900	100	18000	FT	400	100
00014	PLASTER	63334000	Y N	0	20	60	200	POUND	20	60
00015	NUT	77005000	Y N	0	100	700	1000	NUMBER	100	700

**********INV-1119**********

00001	GLUE	21113990	Y N	0	10	60	120	POUND	60	10
00002	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00003	ASPHALT	22261100	Y N	0	1	3	20	TON	3	1
00004	ASPHALT	22261103	Y N	0	1	3	24	TON	3	1
00005	CEMENT	22299901	Y N	0	1	4	10	TON	4	1
00006	GLASS	22444500	Y N	0	30	80	200	SQR FT	80	30
00007	PIPE	62661100	Y N	0	80	26	500	FT	90	16
00008	CONNECTOR	77002000	Y N	0	230	60	400	NUMBER	250	40
00009	NAIL	77003000	Y N	0	100	1000	3000	NUMBER	150	950
00010	SCREW	77004000	Y N	0	100	200	3000	NUMBER	200	100
00011	NUT	77005000	Y N	0	100	700	5000	NUMBER	700	100

**********INV-1121**********

00001	CAP	10004500	Y N	0	80	19	400	NUMBER	89	10
00002	HINGE	11021122	Y N	0	20	9	310	NUMBER	25	4
00003	GLUE	21113990	Y N	0	10	6	150	POUND	12	4
00004	GLUE	21113993	Y N	0	15	2	50	POUND	17	0
00005	ASPHALT	22261100	Y N	0	1	3	20	TON	3	1
00006	ASPHALT	22261103	Y N	0	1	3	24	TON	3	1
00007	CONCRETE	22278000	Y Y	10	1	2	30	TON	2	1
00008	CEMENT	22299900	Y N	0	2	6	30	TON	6	2
00009	CEMENT	22299901	Y Y	5	1	4	10	TON	4	1
00010	GLASS	22444500	Y N	0	40	7	300	SQR FT	44	3
00011	PIPE	62661100	Y N	0	98	50	500	FT	108	40
00012	PLASTER	63334000	Y N	0	20	6	100	POUND	22	4
00013	PLASTER	63334001	Y N	0	10	10	50	POUND	14	6
00014	ASBESTOS	63335000	Y N	0	10	30	100	SHEET	30	10
00015	ASBESTOS	63335002	Y N	0	5	10	20	SHEET	10	5
00016	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00017	CLAY	63337000	Y N	0	50	30	100	CAN	30	50
00018	BOLT	77001000	Y N	0	10	20	300	NUMBER	20	10
00019	CONNECTOR	77002000	Y N	0	23	6	400	NUMBER	25	3
00020	SCREW	77004000	Y N	0	20	10	4000	NUMBER	10	20
00021	NUT	77005000	Y N	0	100	700	7000	NUMBER	700	100

**********INV-1122**********

00001	CAP	10004500	Y N	0	80	19	400	NUMBER	90	10
00002	HINGE	11021122	Y N	0	20	9	350	NUMBER	25	4
00003	BRICK	11280670	Y N	0	9	4	120	LOADS	11	2
00004	CABLE	11397278	Y N	0	80	20	2100	FT	90	10
00005	CABLE	11397279	Y N	0	39	9	250	FT	40	8
00006	GLUE	21113990	Y N	0	10	60	120	POUND	15	55
00007	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00008	CONCRETE	22278000	N N	0	0	3	30	TON	0	3
00009	CEMENT	22299900	Y N	0	2	6	30	TON	6	2
00010	CEMENT	22299901	Y N	0	1	4	10	TON	4	1
00011	GLASS	22444500	Y N	0	40	7	200	SQR FT	47	0
00012	PIPE	62661100	Y N	0	98	25	500	FT	110	13
00013	WIRE	62661122	N Y	3000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y N	0	20	60	200	POUND	60	20
00015	NUT	77005000	Y N	0	100	70	1000	NUMBER	140	30

**********INV-1181**********

00001	CAP	10004500	Y N	0	80	90	400	NUMBER	90	80
00002	HINGE	11021122	Y N	0	20	90	310	NUMBER	90	20
00003	GLUE	21113990	Y N	0	10	60	150	POUND	60	10
00004	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00005	ASPHALT	22261100	Y N	0	1	3	20	TON	3	1
00006	ASPHALT	22261103	Y N	0	1	3	24	TON	3	1
00007	CONCRETE	22278000	Y Y	10	1	2	30	TON	2	1
00008	CEMENT	22299900	Y N	0	6	2	30	TON	2	6
00009	CEMENT	22299901	Y Y	5	1	4	10	TON	4	1
00010	GLASS	22444500	Y N	0	40	70	300	SQR FT	70	40
00011	PIPE	62661100	Y N	0	98	25	500	FT	105	18
00012	PLASTER	63334000	Y N	0	20	60	100	POUND	60	20
00013	PLASTER	63334001	Y N	0	10	20	50	POUND	20	10
00014	ASBESTOS	63335000	Y N	0	10	30	100	SHEET	30	10
00015	ASBESTOS	63335002	Y N	0	5	10	20	SHEET	10	5
00016	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00017	CLAY	63337000	Y N	0	30	50	100	CAN	50	30
00018	BOLT	77001000	Y N	0	100	200	300	NUMBER	200	100
00019	CONNECTOR	77002000	Y N	0	23	60	400	NUMBER	53	30
00020	SCREW	77004000	Y N	0	100	200	4000	NUMBER	200	100
00021	NUT	77005000	Y N	0	100	700	7000	NUMBER	150	650

**********INV-1182*********

00001	CAP	10004500	Y N	0	70	200	400	NUMBER	80	190
00002	HINGE	11021122	Y N	0	20	90	200	NUMBER	30	80
00003	BRICK	11280670	Y N	0	9	40	120	LOADS	15	34
00004	CABLE	11397278	Y N	0	10	60	2500	FT	60	10
00005	CABLE	11397279	Y N	0	39	90	300	FT	50	79
00006	GLUE	21113990	N N	0	0	70	120	POUND	10	60
00007	GLUE	21113993	Y N	0	15	20	50	POUND	20	15
00008	CONCRETE	22278000	Y N	0	1	2	30	TON	2	1
00009	CEMENT	22299900	Y N	0	2	6	30	TON	6	2
00010	CEMENT	22299901	Y N	0	1	4	10	TON	4	1
00011	GLASS	22444500	Y N	0	40	70	300	SQR FT	50	60
00012	PIPE	62661100	Y N	0	110	30	500	FT	120	20
00013	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y N	0	20	60	200	POUND	25	55
00015	PLASTER	63334001	Y N	0	10	20	50	POUND	20	10
00016	ASBESTOS	63335000	Y N	0	30	10	40	SHEET	35	5
00017	ASBESTOS	63335002	Y N	0	5	10	20	SHEET	10	5
00018	PAINT	63336400	N N	0	0	0	190	CAN	0	0
00019	SCREW	77004000	Y N	0	100	200	4000	NUMBER	200	100
00020	NUT	77005000	Y N	0	10	70	2000	NUMBER	70	10

**********INV-1411*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	15	55
00002	GLUE	21113998	Y N	0	15	20	50	POUND	17	18
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	3	1
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	2	2
00005	CEMENT	22299901	Y N	0	1	4	10	TON	2	2
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	34	4
00007	PIPE	62661108	Y N	0	80	26	500	FT	86	20
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	240	50
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	110	900
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	150	150
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	150	650

**********INV-1412*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	15	55
00002	GLUE	21113998	Y N	0	15	20	50	POUND	17	18
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	3	1
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	2	2
00005	CEMENT	22299901	Y N	0	1	4	10	TON	2	2
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	34	4
00007	PIPE	62661108	Y N	0	80	26	500	FT	86	20
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	240	50
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	110	900
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	150	150
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	150	650

**********INV-1421*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	15	55
00002	GLUE	21113998	Y N	0	15	20	50	POUND	17	18
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	3	1
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	2	2
00005	CEMENT	22299901	Y N	0	1	4	10	TON	2	2
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	34	4
00007	PIPE	62661108	Y N	0	80	26	500	FT	86	20
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	240	50
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	110	900
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	150	150
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	150	650

**********INV-1422*********

00001	CAP	10004500	Y N	0	80	19	480	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	9	370	NUMBER	25	4
00003	CABLE	11397278	Y N	0	100	60	2000	FT	110	50
00004	CABLE	11397279	Y N	0	39	9	267	FT	43	5

**********INV-0111***********

00001	CAP	10004500	Y N	0	370	197	1680	NUMBER	0	0
00002	HINGE	11021122	Y N	0	125	72	1230	NUMBER	0	0
00003	BRICK	11280670	Y N	0	40	47	519	LOADS	0	0
00004	CABLE	11397278	Y N	0	410	650	6600	FT	0	0
00005	CABLE	11397279	Y N	0	137	169	817	FT	0	0
00006	GLUE	21113990	N N	0	126	116	480	POUND	0	0
00007	GLUE	21113993	Y N	0	97	50	150	POUND	0	0
00008	ASPHALT	22261100	Y N	0	8	4	60	TON	0	0
00009	ASPHALT	22261103	Y N	0	8	4	72	TON	0	0
00010	CONCRETE	22278000	N Y	10	9	3	120	TON	0	0
00011	CEMENT	22299900	Y N	0	28	4	120	TON	0	0
00012	CEMENT	22299901	Y Y	5	14	11	60	TON	0	0
00013	GLASS	22444500	Y N	0	286	138	1260	SQR FT	0	0
00014	PIPE	62661100	Y N	0	519	304	2500	FT	0	0
00015	WIRE	62661122	N Y	3500	400	100	54000	FT	0	0
00016	PLASTER	63334000	Y N	0	98	168	600	POUND	0	0
00017	PLASTER	63334001	Y N	0	50	40	150	POUND	0	0
00018	ASBESTOS	63335000	Y N	0	105	15	180	SHEET	0	0
00019	ASBESTOS	63335002	Y N	0	34	11	60	SHEET	0	0
00020	PAINT	63336400	N Y	50	0	0	580	CAN	0	0
00021	CLAY	63337000	Y N	0	70	50	200	CAN	0	0
00022	BOLT	77001000	Y N	0	215	25	600	NUMBER	0	0
00023	CONNECTOR	77002000	Y N	0	730	140	1200	NUMBER	0	0
00024	NAIL	77003000	Y N	0	300	1000	8000	NUMBER	0	0
00025	SCREW	77004000	Y N	0	615	405	16000	NUMBER	0	0
00026	NUT	77005000	Y N	0	1525	1845	20000	NUMBER	0	0

**********INV-0112**********

00001	CAP	10004500	Y N	0	179	20	800	NUMBER	0	0
00002	HINGE	11021122	Y N	0	50	8	660	NUMBER	0	0
00003	BRICK	11280670	Y N	0	11	2	120	LOADS	0	0
00004	CABLE	11397278	Y N	0	90	10	2100	FT	0	0
00005	CABLE	11397279	Y N	0	40	8	250	FT	0	0
00006	GLUE	21113990	Y N	0	27	59	270	POUND	0	0
00007	GLUE	21113993	Y N	0	37	15	100	POUND	0	0
00008	ASPHALT	22261100	Y N	0	3	1	20	TON	0	0
00009	ASPHALT	22261103	Y N	0	3	1	24	TON	0	0
00010	CONCRETE	22278000	N Y	10	2	4	60	TON	0	0
00011	CEMENT	22299900	Y N	0	12	4	60	TON	0	0
00012	CEMENT	22299901	Y Y	5	8	2	20	TON	0	0
00013	GLASS	22444500	Y N	0	91	3	500	SQR FT	0	0
00014	PIPE	62661100	Y N	0	218	53	1000	FT	0	0
00015	WIRE	62661122	N Y	3000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	82	24	300	POUND	0	0
00017	PLASTER	63334001	Y N	0	14	6	50	POUND	0	0
00018	ASBESTOS	63335000	Y N	0	30	10	100	SHEET	0	0
00019	ASBESTOS	63335002	Y N	0	10	5	20	SHEET	0	0
00020	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00021	CLAY	63337000	Y N	0	30	50	100	CAN	0	0
00022	BOLT	77001000	Y N	0	20	10	300	NUMBER	0	0
00023	CONNECTOR	77002000	Y N	0	25	3	400	NUMBER	0	0
00024	SCREW	77004000	Y N	0	10	20	4000	NUMBER	0	0
00025	NUT	77005000	Y N	0	840	130	7000	NUMBER	0	0

**********INV-0118**********

00001	CAP	10004500	Y N	0	170	270	800	NUMBER	0	0
00002	HINGE	11021122	Y N	0	120	100	510	NUMBER	0	0
00003	BRICK	11280670	Y N	0	15	34	120	LOADS	0	0
00004	CABLE	11397278	Y N	0	60	10	2500	FT	0	0
00005	CABLE	11397279	Y N	0	50	79	300	FT	0	0
00006	GLUE	21113990	N N	0	70	70	270	POUND	0	0
00007	GLUE	21113993	Y N	0	40	30	100	POUND	0	0
00008	ASPHALT	22261100	Y N	0	3	1	20	TON	0	0
00009	ASPHALT	22261103	Y N	0	3	1	24	TON	0	0
00010	CONCRETE	22278000	Y Y	10	4	2	60	TON	0	0
00011	CEMENT	22299900	Y N	0	8	8	60	TON	0	0
00012	CEMENT	22299901	Y Y	5	8	2	20	TON	0	0
00013	GLASS	22444500	Y N	0	120	100	600	SQR FT	0	0
00014	PIPE	62661100	Y N	0	225	38	1000	FT	0	0
00015	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	85	75	300	POUND	0	0
00017	PLASTER	63334001	Y N	0	40	20	100	POUND	0	0
00018	ASBESTOS	63335000	Y N	0	65	15	140	SHEET	0	0
00019	ASBESTOS	63335002	Y N	0	20	10	40	SHEET	0	0
00020	PAINT	63336400	N Y	50	0	0	390	CAN	0	0
00021	CLAY	63337000	Y N	0	50	30	100	CAN	0	0
00022	BOLT	77001000	Y N	0	200	100	300	NUMBER	0	0
00023	CONNECTOR	77002000	Y N	0	53	30	400	NUMBER	0	0
00024	SCREW	77004000	Y N	0	400	200	8000	NUMBER	0	0
00025	NUT	77005000	Y N	0	220	660	9000	NUMBER	0	0

**********INV-0141***********

00001	GLUE	21113995	Y N	0	30	110	240	POUND	0	0
00002	GLUE	21113998	Y N	0	34	36	100	POUND	0	0
00003	ASPHALT	22261103	Y N	0	6	2	48	TON	0	0
00004	ASPHALT	22261120	Y N	0	4	4	40	TON	0	0
00005	CEMENT	22299901	Y N	0	4	4	20	TON	0	0
00006	GLASS	22444505	Y N	0	68	8	400	SQR FT	0	0
00007	PIPE	62661108	Y N	0	172	40	1000	FT	0	0
00008	CONNECTOR	77002009	Y N	0	480	100	800	NUMBER	0	0
00009	NAIL	77003090	Y N	0	220	1800	6000	NUMBER	0	0
00010	SCREW	77004030	Y N	0	300	300	6000	NUMBER	0	0
00011	NUT	77005077	Y N	0	300	1300	10000	NUMBER	0	0

**********INV-0142************

00001	CAP	10004500	Y N	0	90	9	480	NUMBER	0	0
00002	HINGE	11021122	Y N	0	25	4	370	NUMBER	0	0
00003	CABLE	11397278	Y N	0	110	50	2000	FT	0	0
00004	CABLE	11397279	Y N	0	43	5	267	FT	0	0
00005	GLUE	21113995	Y N	0	15	55	120	POUND	0	0
00006	GLUE	21113998	Y N	0	17	18	50	POUND	0	0
00007	ASPHALT	22261103	Y N	0	3	1	24	TON	0	0
00008	ASPHALT	22261120	Y N	0	2	2	20	TON	0	0
00009	CEMENT	22299901	Y N	0	2	2	10	TON	0	0
00010	GLASS	22444505	Y N	0	34	4	200	SQR FT	0	0
00011	PIPE	62661108	Y N	0	86	20	500	FT	0	0
00012	CONNECTOR	77002009	Y N	0	240	50	400	NUMBER	0	0
00013	NAIL	77003090	Y N	0	110	900	3000	NUMBER	0	0
00014	SCREW	77004030	Y N	0	150	150	3000	NUMBER	0	0
00015	NUT	77005077	Y N	0	150	650	5000	NUMBER	0	0

**********INV-0014********

00001	CAP	10004500	Y N	0	90	9	480	NUMBER	0	0
00002	HINGE	11021122	Y N	0	25	4	370	NUMBER	0	0
00003	CABLE	11397278	Y N	0	110	50	2000	FT	0	0
00004	CABLE	11397279	Y N	0	43	5	267	FT	0	0
00005	GLUE	21113995	Y N	0	45	165	120	POUND	0	0
00006	GLUE	21113998	Y N	0	51	54	50	POUND	0	0
00007	ASPHALT	22261103	Y N	0	9	3	24	TON	0	0
00008	ASPHALT	22261120	Y N	0	6	6	20	TON	0	0
00009	CEMENT	22299901	Y N	0	6	6	10	TON	0	0
00010	GLASS	22444505	Y N	0	102	12	200	SQR FT	0	0
00011	PIPE	62661108	Y N	0	258	60	500	FT	0	0
00012	CONNECTOR	77002009	Y N	0	720	150	400	NUMBER	0	0
00013	NAIL	77003090	Y N	0	330	2700	3000	NUMBER	0	0
00014	SCREW	77004030	Y N	0	450	450	3000	NUMBER	0	0
00015	NUT	77005077	Y N	0	450	1950	5000	NUMBER	0	0

**********INV-0011**********

00001	CAP	10004500	Y N	0	719	487	3280	NUMBER	0	0
00002	HINGE	11021122	Y N	0	295	180	2400	NUMBER	0	0
00003	BRICK	11280670	Y N	0	66	83	759	LOADS	0	0
00004	CABLE	11397278	Y N	0	560	670	11200	FT	0	0
00005	CABLE	11397279	Y N	0	227	256	1367	FT	0	0
00006	GLUE	21113990	N N	0	223	245	1020	POUND	0	0
00007	GLUE	21113993	Y N	0	174	95	350	POUND	0	0
00008	ASPHALT	22261100	Y N	0	14	6	100	TON	0	0
00009	ASPHALT	22261103	Y N	0	14	6	120	TON	0	0
00010	CONCRETE	22278000	N Y	30	15	9	240	TON	0	0
00011	CEMENT	22299900	Y N	0	48	16	240	TON	0	0
00012	CEMENT	22299901	Y Y	15	30	15	100	TON	0	0
00013	GLASS	22444500	Y N	0	497	241	2360	SQR FT	0	0
00014	PIPE	62661100	Y N	0	962	395	4500	FT	0	0
00015	WIRE	62661122	N Y	8500	400	100	90000	FT	0	0
00016	PLASTER	63334000	Y N	0	265	267	1200	POUND	0	0
00017	PLASTER	63334001	Y N	0	104	66	300	POUND	0	0
00018	ASBESTOS	63335000	Y N	0	200	40	420	SHEET	0	0
00019	ASBESTOS	63335002	Y N	0	64	26	120	SHEET	0	0
00020	PAINT	63336400	N Y	150	0	0	1170	CAN	0	0
00021	CLAY	63337000	Y N	0	150	130	400	CAN	0	0
00022	BOLT	77001000	Y N	0	435	135	1200	NUMBER	0	0
00023	CONNECTOR	77002000	Y N	0	808	173	2000	NUMBER	0	0
00024	NAIL	77003000	Y N	0	300	1000	8000	NUMBER	0	0
00025	SCREW	77004000	Y N	0	1025	625	28000	NUMBER	0	0
00026	NUT	77005000	Y N	0	2585	2635	37000	NUMBER		

***********INV-0001*************

00001	CAP	10004500	Y N	0	809	496	3760	NUMBER	0	0
00002	HINGE	11021122	Y N	0	320	184	2400	NUMBER	0	0
00003	BRICK	11280670	Y N	0	66	83	759	LOADS	0	0
00004	CABLE	11397278	Y N	0	670	720	11200	FT	0	0
00005	CABLE	11397279	Y N	0	270	261	1367	FT	0	0
00006	GLUE	21113990	N N	0	223	245	1020	POUND	0	0
00007	GLUE	21113993	Y N	0	174	95	350	POUND	0	0
00008	GLUE	21113995	Y N	0	45	165	480	POUND	0	0
00009	GLUE	21113998	Y N	0	51	54	200	POUND	0	0
00010	ASPHALT	22261100	Y N	0	14	6	100	TON	0	0
00011	ASPHALT	22261103	Y N	0	23	9	216	TON	0	0
00012	ASPHALT	22261120	Y N	0	6	6	80	TON	0	0
00013	CONCRETE	22278000	N Y	30	15	9	240	TON	0	0
00014	CEMENT	22299900	Y N	0	48	16	240	TON	0	0
00015	CEMENT	22299901	Y Y	15	36	21	140	TON	0	0
00016	GLASS	22444500	Y N	0	497	241	2360	SQR FT	0	0
00017	GLASS	22444500	Y N	0	0	0	2360	SQR FT	0	0
00018	GLASS	22444505	Y N	0	102	12	800	SQR FT	0	0
00019	PIPE	62661100	Y N	0	962	395	4500	FT	0	0
00020	PIPE	62661108	Y N	0	258	60	1000	FT	0	0
00021	WIRE	62661122	N Y	8500	400	100	90000	FT	0	0
00022	PLASTER	63334000	Y N	0	265	267	1200	POUND	0	0
00023	PLASTER	63334001	Y N	0	104	66	300	POUND	0	0
00024	ASBESTOS	63335000	Y N	0	200	40	420	SHEET	0	0
00025	ASBESTOS	63335002	Y N	0	64	26	120	SHEET	0	0
00026	PAINT	63336400	N Y	150	0	0	1170	CAN	0	0
00027	CLAY	63337000	Y N	0	150	130	400	CAN	0	0
00028	BOLT	77001000	Y N	0	435	135	1200	NUMBER	0	0
00029	CONNECTOR	77002000	Y N	0	808	173	2000	NUMBER	0	0
00030	CONNECTOR	77002009	Y N	0	720	150	1600	NUMBER	0	0
00031	NAIL	77003000	Y N	0	300	1000	8000	NUMBER	0	0
00032	NAIL	77003090	Y N	0	330	2700	12000	NUMBER	0	0
00033	SCREW	77004000	Y N	0	1025	625	28000	NUMBER	0	0
00034	SCREW	77004030	Y N	0	450	450	12000	NUMBER	0	0
00035	NUT	77005000	Y N	0	2585	2635	37000	NUMBER	0	0

```
INVUPDT A:INV-0111.DBF A:INV-1111.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROCCESSING FILE : INV-1111

PROCCESSING FILE : INV-1112

*** CANNOT FIND  : INV-1113

PROCCESSING FILE : INV-1114

PROCCESSING FILE : INV-1115

*** CANNOT FIND  : INV-1116

*** CANNOT FIND  : INV-1117

*** CANNOT FIND  : INV-1118

PROCCESSING FILE : INV-1119

OK TO ERASE  A:INV-0111.DBF (Y/N)?: Y
```

```
  INVUPDT A:INV-0112.DBF A:INV-1121.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROCCESSING FILE : INV-1121

PROCCESSING FILE : INV-1122

*** CANNOT FIND  : INV-1123

*** CANNOT FIND  : INV-1124

*** CANNOT FIND  : INV-1125

*** CANNOT FIND  : INV-1126

*** CANNOT FIND  : INV-1127

*** CANNOT FIND  : INV-1128

*** CANNOT FIND  : INV-1129

OK TO ERASE  A:INV-0112.DBF (Y/N)?: Y
```

```
INVUPDT A:INV-0118.DBF A:INV-1181.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROOCESSING FILE : INV-1181

PROOCESSING FILE : INV-1182

*** CANNOT FIND  : INV-1183

*** CANNOT FIND  : INV-1184

*** CANNOT FIND  : INV-1185

*** CANNOT FIND  : INV-1186

*** CANNOT FIND  : INV-1187

*** CANNOT FIND  : INV-1188

*** CANNOT FIND  : INV-1189

OK TO ERASE  A:INV-0118.DBF (Y/N)?: Y
```

```
INVUPDT A:INV-0141.DBF A:INV-1411.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROCCESSING FILE : INV-1411


PROCCESSING FILE : INV-1412


*** CANNOT FIND  : INV-1413

*** CANNOT FIND  : INV-1414

*** CANNOT FIND  : INV-1415

*** CANNOT FIND  : INV-1416

*** CANNOT FIND  : INV-1417

*** CANNOT FIND  : INV-1418

*** CANNOT FIND  : INV-1419

OK TO ERASE  A:INV-0141.DBF (Y/N)?: Y
```

```
INVUPDT A:INV-0142.DBF A:INV-1421.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROCCESSING FILE : INV-1421


PROCCESSING FILE : INV-1422


*** CANNOT FIND  : INV-1423


*** CANNOT FIND  : INV-1424


*** CANNOT FIND  : INV-1425


*** CANNOT FIND  : INV-1426


*** CANNOT FIND  : INV-1427


*** CANNOT FIND  : INV-1428


*** CANNOT FIND  : INV-1429

OK TO ERASE  A:INV-0142.DBF (Y/N)?: Y
```

***********INV-1111*********

00001	CAP	10004500	Y	N	0	80	19	480	NUMBER	80	19
00002	HINGE	11021122	Y	N	0	20	9	370	NUMBER	20	9
00003	BRICK	11280670	Y	Y	5	9	10	129	LOADS	9	10
00004	CABLE	11397278	Y	N	0	100	60	2000	FT	100	60
00005	CABLE	11397279	Y	N	0	39	9	267	FT	39	9
00006	GLUE	21113990	Y	N	0	10	6	120	POUND	10	6
00007	GLUE	21113993	N	N	0	0	25	50	POUND	0	25
00008	ASPHALT	22261100	Y	N	0	1	3	20	TON	1	3
00009	ASPHALT	22261103	Y	N	0	3	1	24	TON	3	1
00010	CONCRETE	22278000	Y	Y	10	2	1	30	TON	2	1
00011	CEMENT	22299900	Y	Y	29	6	2	30	TON	6	2
00012	CEMENT	22299901	Y	Y	10	1	24	10	TON	1	24
00013	GLASS	22444500	Y	N	0	30	17	260	SQR FT	30	17
00014	PIPE	62661100	Y	N	0	98	25	500	FT	98	25
00015	WIRE	62661122	Y	Y	800	200	0	18000	FT	200	0
00016	PLASTER	63334000	N	Y	20	0	206	100	POUND	0	206
00017	PLASTER	63334001	Y	N	0	10	20	50	POUND	10	20
00018	ASBESTOS	63335000	Y	N	0	30	10	40	SHEET	30	10
00019	ASBESTOS	63335002	Y	N	0	10	5	20	SHEET	10	5
00020	PAINT	63336400	N	N	0	0	0	190	CAN	0	0
00021	CLAY	63337000	Y	N	0	30	10	100	CAN	30	10
00022	BOLT	77001000	Y	N	0	100	20	300	NUMBER	100	20
00023	CONNECTOR	77002000	Y	N	0	230	60	400	NUMBER	230	60
00024	NAIL	77003000	Y	N	0	100	100	5000	NUMBER	100	100
00025	SCREW	77004000	Y	N	0	100	20	5000	NUMBER	100	20
00026	NUT	77005000	Y	N	0	50	120	5000	NUMBER	50	120

**********INV-1112**********

00001	CAP	10004500	Y	N	0	80	19	400	NUMBER	80	19
00002	HINGE	11021122	Y	N	0	20	9	310	NUMBER	20	9
00003	GLUE	21113990	Y	N	0	10	6	150	POUND	10	6
00004	GLUE	21113993	Y	N	0	15	2	50	POUND	15	2
00005	ASPHALT	22261100	Y	N	0	1	3	20	TON	1	3
00006	ASPHALT	22261103	Y	N	0	3	1	24	TON	3	1
00007	CONCRETE	22278000	Y	Y	10	2	1	30	TON	2	1
00008	CEMENT	22299900	Y	N	0	6	2	30	TON	6	2
00009	CEMENT	22299901	Y	Y	5	1	4	10	TON	1	4
00010	GLASS	22444500	Y	N	0	40	7	300	SQR FT	40	7
00011	PIPE	62661100	Y	N	0	98	25	500	FT	98	25
00012	WIRE	62661122	Y	Y	500	500	0	18000	FT	500	0
00013	PLASTER	63334000	Y	N	0	20	60	100	POUND	20	60
00014	PLASTER	63334001	Y	N	0	10	20	50	POUND	10	20
00015	ASBESTOS	63335000	Y	N	0	30	10	100	SHEET	30	10
00016	ASBESTOS	63335002	Y	N	0	10	5	20	SHEET	10	5
00017	PAINT	63336400	N	Y	50	0	0	200	CAN	0	0
00018	CLAY	63337000	Y	N	0	30	50	100	CAN	30	50
00019	BOLT	77001000	Y	N	0	100	20	300	NUMBER	100	20
00020	CONNECTOR	77002000	Y	N	0	230	60	400	NUMBER	230	60
00021	SCREW	77004000	Y	N	0	100	200	4000	NUMBER	100	200
00022	NUT	77005000	Y	N	0	100	700	7000	NUMBER	100	700

**********INV-1114**********

00001	CAP	10004500	Y	N	0	70	200	400	NUMBER	70	200
00002	HINGE	11021122	Y	N	0	20	90	200	NUMBER	20	90
00003	BRICK	11280670	Y	N	0	9	40	120	LOADS	9	40
00004	CABLE	11397278	Y	N	0	100	600	2500	FT	100	600
00005	CABLE	11397279	Y	N	0	39	90	300	FT	39	90
00006	GLUE	21113990	N	N	0	0	70	120	POUND	0	70
00007	GLUE	21113993	Y	N	0	15	20	50	POUND	15	20
00008	CONCRETE	22278000	Y	N	0	2	1	30	TON	2	1
00009	CEMENT	22299900	Y	N	0	6	2	30	TON	6	2
00010	CEMENT	22299901	Y	N	0	1	4	10	TON	1	4
00011	GLASS	22444500	Y	N	0	40	70	300	SQR FT	40	70
00012	PIPE	62661100	Y	N	0	98	250	500	FT	98	250
00013	WIRE	62661122	N	Y	2000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y	N	0	20	60	200	POUND	20	60
00015	PLASTER	63334001	Y	N	0	10	20	50	POUND	10	20
00016	ASBESTOS	63335000	Y	N	0	30	10	40	SHEET	30	10
00017	ASBESTOS	63335002	Y	N	0	10	5	20	SHEET	10	5
00018	PAINT	63336400	N	N	0	0	0	190	CAN	0	0
00019	SCREW	77004000	Y	N	0	100	200	4000	NUMBER	100	200
00020	NUT	77005000	Y	N	0	100	700	2000	NUMBER	100	700

**********INV-1115**********

00001	CAP	10004500	Y	N	0	80	19	400	NUMBER	80	19
00002	HINGE	11021122	Y	N	0	20	9	350	NUMBER	20	9
00003	BRICK	11280670	Y	N	0	9	10	120	LOADS	9	10
00004	CABLE	11397278	Y	N	0	80	120	2100	FT	80	120
00005	CABLE	11397279	Y	N	0	39	90	250	FT	39	90
00006	GLUE	21113990	Y	N	0	10	60	120	POUND	10	60
00007	GLUE	21113993	Y	N	0	15	20	50	POUND	15	20
00008	CONCRETE	22278000	N	N	0	0	3	30	TON	0	3
00009	CEMENT	22299900	Y	N	0	6	2	30	TON	6	2
00010	CEMENT	22299901	Y	N	0	1	4	10	TON	1	4
00011	GLASS	22444500	Y	N	0	40	70	200	SQR FT	40	70
00012	PIPE	62661100	Y	N	0	98	25	500	FT	98	25
00013	WIRE	62661122	N	Y	0	900	100	18000	FT	900	100
00014	PLASTER	63334000	Y	N	0	20	60	200	POUND	20	60
00015	NUT	77005000	Y	N	0	100	700	1000	NUMBER	100	700

**********INV-1119**********

00001	GLUE	21113990	Y	N	0	10	60	120	POUND	10	60
00002	GLUE	21113993	Y	N	0	15	20	50	POUND	15	20
00003	ASPHALT	22261100	Y	N	0	1	3	20	TON	1	3
00004	ASPHALT	22261103	Y	N	0	1	3	24	TON	1	3
00005	CEMENT	22299901	Y	N	0	1	4	10	TON	1	4
00006	GLASS	22444500	Y	N	0	30	80	200	SQR FT	30	80
00007	PIPE	62661100	Y	N	0	80	26	500	FT	80	26
00008	CONNECTOR	77002000	Y	N	0	230	60	400	NUMBER	230	60
00009	NAIL	77003000	Y	N	0	100	1000	3000	NUMBER	100	1000
00010	SCREW	77004000	Y	N	0	100	200	3000	NUMBER	100	200
00011	NUT	77005000	Y	N	0	100	700	5000	NUMBER	100	700

**********INV-1121**********

00001	CAP	10004500	Y	N	0	80	19	400	NUMBER	80	19
00002	HINGE	11021122	Y	N	0	20	9	310	NUMBER	20	9
00003	GLUE	21113990	Y	N	0	10	6	150	POUND	10	6
00004	GLUE	21113993	Y	N	0	15	2	50	POUND	15	2
00005	ASPHALT	22261100	Y	N	0	1	3	20	TON	1	3
00006	ASPHALT	22261103	Y	N	0	1	3	24	TON	1	3
00007	CONCRETE	22278000	Y	Y	10	1	2	30	TON	1	2
00008	CEMENT	22299900	Y	N	0	2	6	30	TON	2	6
00009	CEMENT	22299901	Y	Y	5	1	4	10	TON	1	4
00010	GLASS	22444500	Y	N	0	40	7	300	SQR FT	40	7
00011	PIPE	62661100	Y	N	0	98	50	500	FT	98	50
00012	PLASTER	63334000	Y	N	0	20	6	100	POUND	20	6
00013	PLASTER	63334001	Y	N	0	10	10	50	POUND	10	10
00014	ASBESTOS	63335000	Y	N	0	10	30	100	SHEET	10	30
00015	ASBESTOS	63335002	Y	N	0	5	10	20	SHEET	5	10
00016	PAINT	63336400	N	Y	50	0	0	200	CAN	0	0
00017	CLAY	63337000	Y	N	0	50	30	100	CAN	50	30
00018	BOLT	77001000	Y	N	0	10	20	300	NUMBER	10	20
00019	CONNECTOR	77002000	Y	N	0	23	6	400	NUMBER	23	6
00020	SCREW	77004000	Y	N	0	20	10	4000	NUMBER	20	10
00021	NUT	77005000	Y	N	0	100	700	7000	NUMBER	100	700

**********INV-1122**********

00001	CAP	10004500	Y	N	0	80	19	400	NUMBER	80	19
00002	HINGE	11021122	Y	N	0	20	9	350	NUMBER	20	9
00003	BRICK	11280670	Y	N	0	9	4	120	LOADS	9	4
00004	CABLE	11397278	Y	N	0	80	20	2100	FT	80	20
00005	CABLE	11397279	Y	N	0	39	9	250	FT	39	9
00006	GLUE	21113990	Y	N	0	10	60	120	POUND	10	60
00007	GLUE	21113993	Y	N	0	15	20	50	POUND	15	20
00008	CONCRETE	22278000	N	N	0	0	3	30	TON	0	3
00009	CEMENT	22299900	Y	N	0	2	6	30	TON	2	6
00010	CEMENT	22299901	Y	N	0	1	4	10	TON	1	4
00011	GLASS	22444500	Y	N	0	40	7	200	SQR FT	40	7
00012	PIPE	62661100	Y	N	0	98	25	500	FT	98	25
00013	WIRE	62661122	N	Y	3000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y	N	0	20	60	200	POUND	20	60
00015	NUT	77005000	Y	N	0	100	70	1000	NUMBER	100	70

**********INV-1181**********

00001	CAP	10004500	Y N	0	80	90	400	NUMBER	80	90
00002	HINGE	11021122	Y N	0	20	90	310	NUMBER	20	90
00003	GLUE	21113990	Y N	0	10	60	150	POUND	10	60
00004	GLUE	21113993	Y N	0	15	20	50	POUND	15	20
00005	ASPHALT	22261100	Y N	0	1	3	20	TON	1	3
00006	ASPHALT	22261103	Y N	0	1	3	24	TON	1	3
00007	CONCRETE	22278000	Y Y	10	1	2	30	TON	1	2
00008	CEMENT	22299900	Y N	0	6	2	30	TON	6	2
00009	CEMENT	22299901	Y Y	5	1	4	10	TON	1	4
00010	GLASS	22444500	Y N	0	40	70	300	SQR FT	40	70
00011	PIPE	62661100	Y N	0	98	25	500	FT	98	25
00012	PLASTER	63334000	Y N	0	20	60	100	POUND	20	60
00013	PLASTER	63334001	Y N	0	10	20	50	POUND	10	20
00014	ASBESTOS	63335000	Y N	0	10	30	100	SHEET	10	30
00015	ASBESTOS	63335002	Y N	0	5	10	20	SHEET	5	10
00016	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00017	CLAY	63337000	Y N	0	30	50	100	CAN	30	50
00018	BOLT	77001000	Y N	0	100	200	300	NUMBER	100	200
00019	CONNECTOR	77002000	Y N	0	23	60	400	NUMBER	23	60
00020	SCREW	77004000	Y N	0	100	200	4000	NUMBER	100	200
00021	NUT	77005000	Y N	0	100	700	7000	NUMBER	100	700

**********INV-1182*********

00001	CAP	10004500	Y N	0	70	200	400	NUMBER	70	200
00002	HINGE	11021122	Y N	0	20	90	200	NUMBER	20	90
00003	BRICK	11280670	Y N	0	9	40	120	LOADS	9	40
00004	CABLE	11397278	Y N	0	10	60	2500	FT	10	60
00005	CABLE	11397279	Y N	0	39	90	300	FT	39	90
00006	GLUE	21113990	N N	0	0	70	120	POUND	0	70
00007	GLUE	21113993	Y N	0	15	20	50	POUND	15	20
00008	CONCRETE	22278000	Y N	0	1	2	30	TON	1	2
00009	CEMENT	22299900	Y N	0	2	6	30	TON	2	6
00010	CEMENT	22299901	Y N	0	1	4	10	TON	1	4
00011	GLASS	22444500	Y N	0	40	70	300	SQR FT	40	70
00012	PIPE	62661100	Y N	0	110	30	500	FT	110	30
00013	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00014	PLASTER	63334000	Y N	0	20	60	200	POUND	20	60
00015	PLASTER	63334001	Y N	0	10	20	50	POUND	10	20
00016	ASBESTOS	63335000	Y N	0	30	10	40	SHEET	30	10
00017	ASBESTOS	63335002	Y N	0	5	10	20	SHEET	5	10
00018	PAINT	63336400	N N	0	0	0	190	CAN	0	0
00019	SCREW	77004000	Y N	0	100	200	4000	NUMBER	100	200
00020	NUT	77005000	Y N	0	10	70	2000	NUMBER	10	70

**********INV-1411*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	10	60
00002	GLUE	21113998	Y N	0	15	20	50	POUND	15	20
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	2	2
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	1	3
00005	CEMENT	22299901	Y N	0	1	4	10	TON	1	4
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	30	8
00007	PIPE	62661108	Y N	0	80	26	500	FT	80	26
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	230	60
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	100	1000
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	100	200
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	100	700

**********INV-1412*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	10	60
00002	GLUE	21113998	Y N	0	15	20	50	POUND	15	20
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	2	2
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	1	3
00005	CEMENT	22299901	Y N	0	1	4	10	TON	1	4
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	30	8
00007	PIPE	62661108	Y N	0	80	26	500	FT	80	26
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	230	60
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	100	1000
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	100	200
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	100	700

**********INV-1421*********

00001	GLUE	21113995	Y N	0	10	60	120	POUND	10	60
00002	GLUE	21113998	Y N	0	15	20	50	POUND	15	20
00003	ASPHALT	22261103	Y N	0	2	2	24	TON	2	2
00004	ASPHALT	22261120	Y N	0	1	3	20	TON	1	3
00005	CEMENT	22299901	Y N	0	1	4	10	TON	1	4
00006	GLASS	22444505	Y N	0	30	8	200	SQR FT	30	8
00007	PIPE	62661108	Y N	0	80	26	500	FT	80	26
00008	CONNECTOR	77002009	Y N	0	230	60	400	NUMBER	230	60
00009	NAIL	77003090	Y N	0	100	1000	3000	NUMBER	100	1000
00010	SCREW	77004030	Y N	0	100	200	3000	NUMBER	100	200
00011	NUT	77005077	Y N	0	100	700	5000	NUMBER	100	700

**********INV-1422*********

00001	CAP	10004500	Y N	0	80	19	480	NUMBER	80	19
00002	HINGE	11021122	Y N	0	20	9	370	NUMBER	20	9
00003	CABLE	11397278	Y N	0	100	60	2000	FT	100	60
00004	CABLE	11397279	Y N	0	39	9	267	FT	39	9

**********INV-0111***********

00001	CAP	10004500	Y N	0	310	454	1680	NUMBER	370	197
00002	HINGE	11021122	Y N	0	80	189	1230	NUMBER	125	72
00003	BRICK	11280670	Y Y	5	27	107	519	LOADS	40	47
00004	CABLE	11397278	Y N	0	280	1430	6600	FT	410	650
00005	CABLE	11397279	Y N	0	117	358	817	FT	137	169
00006	GLUE	21113990	N N	0	40	318	480	POUND	126	116
00007	GLUE	21113993	N N	0	60	137	150	POUND	97	50
00008	ASPHALT	22261100	Y N	0	3	13	60	TON	8	4
00009	ASPHALT	22261103	Y N	0	7	9	72	TON	8	4
00010	CONCRETE	22278000	N Y	20	6	9	120	TON	9	3
00011	CEMENT	22299900	Y Y	29	24	12	120	TON	28	4
00012	CEMENT	22299901	Y Y	15	5	51	60	TON	14	11
00013	GLASS	22444500	Y N	0	180	382	1260	SQR FT	286	138
00014	PIPE	62661100	Y N	0	472	655	2500	FT	519	304
00015	WIRE	62661122	N Y	3300	1600	200	54000	FT	400	100
00016	PLASTER	63334000	N Y	20	60	554	600	POUND	98	168
00017	PLASTER	63334001	Y N	0	30	100	150	POUND	50	40
00018	ASBESTOS	63335000	Y N	0	90	45	180	SHEET	105	15
00019	ASBESTOS	63335002	Y N	0	30	26	60	SHEET	34	11
00020	PAINT	63336400	N Y	50	0	0	580	CAN	0	0
00021	CLAY	63337000	Y N	0	60	110	200	CAN	70	50
00022	BOLT	77001000	Y N	0	200	65	600	NUMBER	215	25
00023	CONNECTOR	77002000	Y N	0	690	320	1200	NUMBER	730	140
00024	NAIL	77003000	Y N	0	200	2100	8000	NUMBER	300	1000
00025	SCREW	77004000	Y N	0	400	1025	16000	NUMBER	615	405
00026	NUT	77005000	Y N	0	450	4765	20000	NUMBER	1525	1845

**********INV-0112**********

00001	CAP	10004500	Y N	0	160	58	800	NUMBER	179	20
00002	HINGE	11021122	Y N	0	40	26	660	NUMBER	50	8
00003	BRICK	11280670	Y N	0	9	6	120	LOADS	11	2
00004	CABLE	11397278	Y N	0	80	30	2100	FT	90	10
00005	CABLE	11397279	Y N	0	39	17	250	FT	40	8
00006	GLUE	21113990	Y N	0	20	125	270	POUND	27	59
00007	GLUE	21113993	Y N	0	30	37	100	POUND	37	15
00008	ASPHALT	22261100	Y N	0	1	4	20	TON	3	1
00009	ASPHALT	22261103	Y N	0	1	4	24	TON	3	1
00010	CONCRETE	22278000	N Y	10	1	9	60	TON	2	4
00011	CEMENT	22299900	Y N	0	4	16	60	TON	12	4
00012	CEMENT	22299901	Y Y	5	2	10	20	TON	8	2
00013	GLASS	22444500	Y N	0	80	17	500	SQR FT	91	3
00014	PIPE	62661100	Y N	0	196	128	1000	FT	218	53
00015	WIRE	62661122	N Y	3000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	40	90	300	POUND	82	24
00017	PLASTER	63334001	Y N	0	10	16	50	POUND	14	6
00018	ASBESTOS	63335000	Y N	0	10	40	100	SHEET	30	10
00019	ASBESTOS	63335002	Y N	0	5	15	20	SHEET	10	5
00020	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00021	CLAY	63337000	Y N	0	50	80	100	CAN	30	50
00022	BOLT	77001000	Y N	0	10	30	300	NUMBER	20	10
00023	CONNECTOR	77002000	Y N	0	23	9	400	NUMBER	25	3
00024	SCREW	77004000	Y N	0	20	30	4000	NUMBER	10	20
00025	NUT	77005000	Y N	0	200	900	7000	NUMBER	840	130

**********INV-0118**********

00001	CAP	10004500	Y N	0	150	560	800	NUMBER	170	270
00002	HINGE	11021122	Y N	0	40	280	510	NUMBER	120	100
00003	BRICK	11280670	Y N	0	9	74	120	LOADS	15	34
00004	CABLE	11397278	Y N	0	10	70	2500	FT	60	10
00005	CABLE	11397279	Y N	0	39	169	300	FT	50	79
00006	GLUE	21113990	N N	0	10	200	270	POUND	70	70
00007	GLUE	21113993	Y N	0	30	70	100	POUND	40	30
00008	ASPHALT	22261100	Y N	0	1	4	20	TON	3	1
00009	ASPHALT	22261103	Y N	0	1	4	24	TON	3	1
00010	CONCRETE	22278000	Y Y	10	2	6	60	TON	4	2
00011	CEMENT	22299900	Y N	0	8	16	60	TON	8	8
00012	CEMENT	22299901	Y Y	5	2	10	20	TON	8	2
00013	GLASS	22444500	Y N	0	80	240	600	SQR FT	120	100
00014	PIPE	62661100	Y N	0	208	93	1000	FT	225	38
00015	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	40	195	300	POUND	85	75
00017	PLASTER	63334001	Y N	0	20	60	100	POUND	40	20
00018	ASBESTOS	63335000	Y N	0	40	55	140	SHEET	65	15
00019	ASBESTOS	63335002	Y N	0	10	30	40	SHEET	20	10
00020	PAINT	63336400	N Y	50	0	0	390	CAN	0	0
00021	CLAY	63337000	Y N	0	30	80	100	CAN	50	30
00022	BOLT	77001000	Y N	0	100	300	300	NUMBER	200	100
00023	CONNECTOR	77002000	Y N	0	23	90	400	NUMBER	53	30
00024	SCREW	77004000	Y N	0	200	600	8000	NUMBER	400	200
00025	NUT	77005000	Y N	0	110	1430	9000	NUMBER	220	660

**********INV-0141***********

00001	GLUE	21113995	Y N	0	20	230	240	POUND	30	110
00002	GLUE	21113998	Y N	0	30	76	100	POUND	34	36
00003	ASPHALT	22261103	Y N	0	4	6	48	TON	6	2
00004	ASPHALT	22261120	Y N	0	2	10	40	TON	4	4
00005	CEMENT	22299901	Y N	0	2	12	20	TON	4	4
00006	GLASS	22444505	Y N	0	60	24	400	SQR FT	68	8
00007	PIPE	62661108	Y N	0	160	92	1000	FT	172	40
00008	CONNECTOR	77002009	Y N	0	460	220	800	NUMBER	480	100
00009	NAIL	77003090	Y N	0	200	3800	6000	NUMBER	220	1800
00010	SCREW	77004030	Y N	0	200	700	6000	NUMBER	300	300
00011	NUT	77005077	Y N	0	200	2700	10000	NUMBER	300	1300

**********INV-0142************

00001	CAP	10004500	Y N	0	80	28	480	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	13	370	NUMBER	25	4
00003	CABLE	11397278	Y N	0	100	110	2000	FT	110	50
00004	CABLE	11397279	Y N	0	39	14	267	FT	43	5
00005	GLUE	21113995	Y N	0	10	115	120	POUND	15	55
00006	GLUE	21113998	Y N	0	15	38	50	POUND	17	18
00007	ASPHALT	22261103	Y N	0	2	3	24	TON	3	1
00008	ASPHALT	22261120	Y N	0	1	5	20	TON	2	2
00009	CEMENT	22299901	Y N	0	1	6	10	TON	2	2
00010	GLASS	22444505	Y N	0	30	12	200	SQR FT	34	4
00011	PIPE	62661108	Y N	0	80	46	500	FT	86	20
00012	CONNECTOR	77002009	Y N	0	230	110	400	NUMBER	240	50
00013	NAIL	77003090	Y N	0	100	1900	3000	NUMBER	110	900
00014	SCREW	77004030	Y N	0	100	350	3000	NUMBER	150	150
00015	NUT	77005077	Y N	0	100	1350	5000	NUMBER	150	650

```
  INVUPDT A:INV-0011.DBF A:INV-0111.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y


PROCCESSING FILE : INV-0111


PROCCESSING FILE : INV-0112


*** CANNOT FIND  : INV-0113


*** CANNOT FIND  : INV-0114


*** CANNOT FIND  : INV-0115


*** CANNOT FIND  : INV-0116


*** CANNOT FIND  : INV-0117

PROCCESSING FILE : INV-0118


*** CANNOT FIND  : INV-0119

OK TO ERASE  A:INV-0011.DBF (Y/N)?: Y
```

```
INVUPDT A:INV-0014.DBF A:INV-0141.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y


PROCCESSING FILE : INV-0141


PROCCESSING FILE : INV-0142



*** CANNOT FIND  : INV-0143


*** CANNOT FIND  : INV-0144


*** CANNOT FIND  : INV-0145


*** CANNOT FIND  : INV-0146


*** CANNOT FIND  : INV-0147


*** CANNOT FIND  : INV-0148


*** CANNOT FIND  : INV-0149

OK TO ERASE  A:INV-0014.DBF (Y/N)?: Y
```

**********INV-0111***********

00001	CAP	10004500	Y N	0	310	454	1680	NUMBER	310	454
00002	HINGE	11021122	Y N	0	80	189	1230	NUMBER	80	189
00003	BRICK	11280670	Y Y	5	27	107	519	LOADS	27	107
00004	CABLE	11397278	Y N	0	280	1430	6600	FT	280	1430
00005	CABLE	11397279	Y N	0	117	358	817	FT	117	358
00006	GLUE	21113990	N N	0	40	318	480	POUND	40	318
00007	GLUE	21113993	N N	0	60	137	150	POUND	60	137
00008	ASPHALT	22261100	Y N	0	3	13	60	TON	3	13
00009	ASPHALT	22261103	Y N	0	7	9	72	TON	7	9
00010	CONCRETE	22278000	N Y	20	6	9	120	TON	6	9
00011	CEMENT	22299900	Y Y	29	24	12	120	TON	24	12
00012	CEMENT	22299901	Y Y	15	5	51	60	TON	5	51
00013	GLASS	22444500	Y N	0	180	382	1260	SQR FT	180	382
00014	PIPE	62661100	Y N	0	472	655	2500	FT	472	655
00015	WIRE	62661122	N Y	3300	1600	200	54000	FT	1600	200
00016	PLASTER	63334000	N Y	20	60	554	600	POUND	60	554
00017	PLASTER	63334001	Y N	0	30	100	150	POUND	30	100
00018	ASBESTOS	63335000	Y N	0	90	45	180	SHEET	90	45
00019	ASBESTOS	63335002	Y N	0	30	26	60	SHEET	30	26
00020	PAINT	63336400	N Y	50	0	0	580	CAN	0	0
00021	CLAY	63337000	Y N	0	60	110	200	CAN	60	110
00022	BOLT	77001000	Y N	0	200	65	600	NUMBER	200	65
00023	CONNECTOR	77002000	Y N	0	690	320	1200	NUMBER	690	320
00024	NAIL	77003000	Y N	0	200	2100	8000	NUMBER	200	2100
00025	SCREW	77004000	Y N	0	400	1025	16000	NUMBER	400	1025
00026	NUT	77005000	Y N	0	450	4765	20000	NUMBER	450	4765

**********INV-0112**********

00001	CAP	10004500	Y N	0	160	58	800	NUMBER	160	58
00002	HINGE	11021122	Y N	0	40	26	660	NUMBER	40	26
00003	BRICK	11280670	Y N	0	9	6	120	LOADS	9	6
00004	CABLE	11397278	Y N	0	80	30	2100	FT	80	30
00005	CABLE	11397279	Y N	0	39	17	250	FT	39	17
00006	GLUE	21113990	Y N	0	20	125	270	POUND	20	125
00007	GLUE	21113993	Y N	0	30	37	100	POUND	30	37
00008	ASPHALT	22261100	Y N	0	1	4	20	TON	1	4
00009	ASPHALT	22261103	Y N	0	1	4	24	TON	1	4
00010	CONCRETE	22278000	N Y	10	1	9	60	TON	1	9
00011	CEMENT	22299900	Y N	0	4	16	60	TON	4	16
00012	CEMENT	22299901	Y Y	5	2	10	20	TON	2	10
00013	GLASS	22444500	Y N	0	80	17	500	SQR FT	80	17
00014	PIPE	62661100	Y N	0	196	128	1000	FT	196	128
00015	WIRE	62661122	N Y	3000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	40	90	300	POUND	40	90
00017	PLASTER	63334001	Y N	0	10	16	50	POUND	10	16
00018	ASBESTOS	63335000	Y N	0	10	40	100	SHEET	10	40
00019	ASBESTOS	63335002	Y N	0	5	15	20	SHEET	5	15
00020	PAINT	63336400	N Y	50	0	0	200	CAN	0	0
00021	CLAY	63337000	Y N	0	50	80	100	CAN	50	80
00022	BOLT	77001000	Y N	0	10	30	300	NUMBER	10	30
00023	CONNECTOR	77002000	Y N	0	23	9	400	NUMBER	23	9
00024	SCREW	77004000	Y N	0	20	30	4000	NUMBER	20	30
00025	NUT	77005000	Y N	0	200	900	7000	NUMBER	200	900

**********INV-0118**********

00001	CAP	10004500	Y N	0	150	560	800	NUMBER	150	560
00002	HINGE	11021122	Y N	0	40	280	510	NUMBER	40	280
00003	BRICK	11280670	Y N	0	9	74	120	LOADS	9	74
00004	CABLE	11397278	Y N	0	10	70	2500	FT	10	70
00005	CABLE	11397279	Y N	0	39	169	300	FT	39	169
00006	GLUE	21113990	N N	0	10	200	270	POUND	10	200
00007	GLUE	21113993	Y N	0	30	70	100	POUND	30	70
00008	ASPHALT	22261100	Y N	0	1	4	20	TON	1	4
00009	ASPHALT	22261103	Y N	0	1	4	24	TON	1	4
00010	CONCRETE	22278000	Y Y	10	2	6	60	TON	2	6
00011	CEMENT	22299900	Y N	0	8	16	60	TON	8	16
00012	CEMENT	22299901	Y Y	5	2	10	20	TON	2	10
00013	GLASS	22444500	Y N	0	80	240	600	SQR FT	80	240
00014	PIPE	62661100	Y N	0	208	93	1000	FT	208	93
00015	WIRE	62661122	N Y	2000	0	0	18000	FT	0	0
00016	PLASTER	63334000	Y N	0	40	195	300	POUND	40	195
00017	PLASTER	63334001	Y N	0	20	60	100	POUND	20	60
00018	ASBESTOS	63335000	Y N	0	40	55	140	SHEET	40	55
00019	ASBESTOS	63335002	Y N	0	10	30	40	SHEET	10	30
00020	PAINT	63336400	N Y	50	0	0	390	CAN	0	0
00021	CLAY	63337000	Y N	0	30	80	100	CAN	30	80
00022	BOLT	77001000	Y N	0	100	300	300	NUMBER	100	300
00023	CONNECTOR	77002000	Y N	0	23	90	400	NUMBER	23	90
00024	SCREW	77004000	Y N	0	200	600	8000	NUMBER	200	600
00025	NUT	77005000	Y N	0	110	1430	9000	NUMBER	110	1430

**********INV-0141***********

00001	GLUE	21113995	Y N	0	20	230	240	POUND	20	230
00002	GLUE	21113998	Y N	0	30	76	100	POUND	30	76
00003	ASPHALT	22261103	Y N	0	4	6	48	TON	4	6
00004	ASPHALT	22261120	Y N	0	2	10	40	TON	2	10
00005	CEMENT	22299901	Y N	0	2	12	20	TON	2	12
00006	GLASS	22444505	Y N	0	60	24	400	SQR FT	60	24
00007	PIPE	62661108	Y N	0	160	92	1000	FT	160	92
00008	CONNECTOR	77002009	Y N	0	460	220	800	NUMBER	460	220
00009	NAIL	77003090	Y N	0	200	3800	6000	NUMBER	200	3800
00010	SCREW	77004030	Y N	0	200	700	6000	NUMBER	200	700
00011	NUT	77005077	Y N	0	200	2700	10000	NUMBER	200	2700

**********INV-0142************

00001	CAP	10004500	Y N	0	80	28	480	NUMBER	80	28
00002	HINGE	11021122	Y N	0	20	13	370	NUMBER	20	13
00003	CABLE	11397278	Y N	0	100	110	2000	FT	100	110
00004	CABLE	11397279	Y N	0	39	14	267	FT	39	14
00005	GLUE	21113995	Y N	0	10	115	120	POUND	10	115
00006	GLUE	21113998	Y N	0	15	38	50	POUND	15	38
00007	ASPHALT	22261103	Y N	0	2	3	24	TON	2	3
00008	ASPHALT	22261120	Y N	0	1	5	20	TON	1	5
00009	CEMENT	22299901	Y N	0	1	6	10	TON	1	6
00010	GLASS	22444505	Y N	0	30	12	200	SQR FT	30	12
00011	PIPE	62661108	Y N	0	80	46	500	FT	80	46
00012	CONNECTOR	77002009	Y N	0	230	110	400	NUMBER	230	110
00013	NAIL	77003090	Y N	0	100	1900	3000	NUMBER	100	1900
00014	SCREW	77004030	Y N	0	100	350	3000	NUMBER	100	350
00015	NUT	77005077	Y N	0	100	1350	5000	NUMBER	100	1350

**********INV-0014********

00001	CAP	10004500	Y N	0	80	37	480	NUMBER	90	9
00002	HINGE	11021122	Y N	0	20	17	370	NUMBER	25	4
00003	CABLE	11397278	Y N	0	100	160	2000	FT	110	50
00004	CABLE	11397279	Y N	0	39	19	267	FT	43	5
00005	GLUE	21113995	Y N	0	30	510	120	POUND	45	165
00006	GLUE	21113998	Y N	0	45	168	50	POUND	51	54
00007	ASPHALT	22261103	Y N	0	6	12	24	TON	9	3
00008	ASPHALT	22261120	Y N	0	3	21	20	TON	6	6
00009	CEMENT	22299901	Y N	0	3	24	10	TON	6	6
00010	GLASS	22444505	Y N	0	90	48	200	SQR FT	102	12
00011	PIPE	62661108	Y N	0	240	198	500	FT	258	60
00012	CONNECTOR	77002009	Y N	0	690	480	400	NUMBER	720	150
00013	NAIL	77003090	Y N	0	300	8400	3000	NUMBER	330	2700
00014	SCREW	77004030	Y N	0	300	1500	3000	NUMBER	450	450
00015	NUT	77005077	Y N	0	300	6000	5000	NUMBER	450	1950

**********INV-0011**********

00001	CAP	10004500	Y N	0	620	1559	3280	NUMBER	719	487
00002	HINGE	11021122	Y N	0	160	675	2400	NUMBER	295	180
00003	BRICK	11280670	Y Y	5	45	270	759	LOADS	66	83
00004	CABLE	11397278	Y N	0	370	2200	11200	FT	560	670
00005	CABLE	11397279	Y N	0	195	800	1367	FT	227	256
00006	GLUE	21113990	N N	0	70	888	1020	POUND	223	245
00007	GLUE	21113993	N N	0	120	339	350	POUND	174	95
00008	ASPHALT	22261100	Y N	0	5	27	100	TON	14	6
00009	ASPHALT	22261103	Y N	0	9	23	120	TON	14	6
00010	CONCRETE	22278000	N Y	40	9	33	240	TON	15	9
00011	CEMENT	22299900	Y Y	29	36	60	240	TON	48	16
00012	CEMENT	22299901	Y Y	25	9	86	100	TON	30	15
00013	GLASS	22444500	Y N	0	340	880	2360	SQR FT	497	241
00014	PIPE	62661100	Y N	0	876	1271	4500	FT	962	395
00015	WIRE	62661122	N Y	8300	1600	300	90000	FT	400	100
00016	PLASTER	63334000	N Y	20	140	1106	1200	POUND	265	267
00017	PLASTER	63334001	Y N	0	60	242	300	POUND	104	66
00018	ASBESTOS	63335000	Y N	0	140	180	420	SHEET	200	40
00019	ASBESTOS	63335002	Y N	0	45	97	120	SHEET	64	26
00020	PAINT	63336400	N Y	150	0	0	1170	CAN	0	0
00021	CLAY	63337000	Y N	0	140	400	400	CAN	150	130
00022	BOLT	77001000	Y N	0	310	530	1200	NUMBER	435	135
00023	CONNECTOR	77002000	Y N	0	736	592	2000	NUMBER	808	173
00024	NAIL	77003000	Y N	0	200	3100	8000	NUMBER	300	1000
00025	SCREW	77004000	Y N	0	620	2280	28000	NUMBER	1025	625
00026	NUT	77005000	Y N	0	760	9730	37000	NUMBER	2585	2635

```
INVUPDT A:INV-0001.DBF A:INV-0011.DBF

UPDATES INV FILES ONLY
 (Y/N)?: Y

PROCCESSING FILE : INV-0011


*** CANNOT FIND  : INV-0012

*** CANNOT FIND  : INV-0013

PROCCESSING FILE : INV-0014


*** CANNOT FIND  : INV-0015

*** CANNOT FIND  : INV-0016

*** CANNOT FIND  : INV-0017

*** CANNOT FIND  : INV-0018

*** CANNOT FIND  : INV-0019

OK TO ERASE  A:INV-0001.DBF (Y/N)?: Y
```

**********INV-0011**********

00001	CAP	10004500	Y N	0	620	1559	3280	NUMBER	620	1559
00002	HINGE	11021122	Y N	0	160	675	2400	NUMBER	160	675
00003	BRICK	11280670	Y Y	5	45	270	759	LOADS	45	270
00004	CABLE	11397278	Y N	0	370	2200	11200	FT	370	2200
00005	CABLE	11397279	Y N	0	195	800	1367	FT	195	800
00006	GLUE	21113990	N N	0	70	888	1020	POUND	70	888
00007	GLUE	21113993	N N	0	120	339	350	POUND	120	339
00008	ASPHALT	22261100	Y N	0	5	27	100	TON	5	27
00009	ASPHALT	22261103	Y N	0	9	23	120	TON	9	23
00010	CONCRETE	22278000	N Y	40	9	33	240	TON	9	33
00011	CEMENT	22299900	Y Y	29	36	60	240	TON	36	60
00012	CEMENT	22299901	Y Y	25	9	86	100	TON	9	86
00013	GLASS	22444500	Y N	0	340	880	2360	SQR FT	340	880
00014	PIPE	62661100	Y N	0	876	1271	4500	FT	876	1271
00015	WIRE	62661122	N Y	8300	1600	300	90000	FT	1600	300
00016	PLASTER	63334000	N Y	20	140	1106	1200	POUND	140	1106
00017	PLASTER	63334001	Y N	0	60	242	300	POUND	60	242
00018	ASBESTOS	63335000	Y N	0	140	180	420	SHEET	140	180
00019	ASBESTOS	63335002	Y N	0	45	97	120	SHEET	45	97
00020	PAINT	63336400	N Y	150	0	0	1170	CAN	0	0
00021	CLAY	63337000	Y N	0	140	400	400	CAN	140	400
00022	BOLT	77001000	Y N	0	310	530	1200	NUMBER	310	530
00023	CONNECTOR	77002000	Y N	0	736	592	2000	NUMBER	736	592
00024	NAIL	77003000	Y N	0	200	3100	8000	NUMBER	200	3100
00025	SCREW	77004000	Y N	0	620	2280	28000	NUMBER	620	2280
00026	NUT	77005000	Y N	0	760	9730	37000	NUMBER	760	9730

**********INV-0014********

00001	CAP	10004500	Y N	0	80	37	480	NUMBER	80	37
00002	HINGE	11021122	Y N	0	20	17	370	NUMBER	20	17
00003	CABLE	11397278	Y N	0	100	160	2000	FT	100	160
00004	CABLE	11397279	Y N	0	39	19	267	FT	39	19
00005	GLUE	21113995	Y N	0	30	510	120	POUND	30	510
00006	GLUE	21113998	Y N	0	45	168	50	POUND	45	168
00007	ASPHALT	22261103	Y N	0	6	12	24	TON	6	12
00008	ASPHALT	22261120	Y N	0	3	21	20	TON	3	21
00009	CEMENT	22299901	Y N	0	3	24	10	TON	3	24
00010	GLASS	22444505	Y N	0	90	48	200	SQR FT	90	48
00011	PIPE	62661108	Y N	0	240	198	500	FT	240	198
00012	CONNECTOR	77002009	Y N	0	690	480	400	NUMBER	690	480
00013	NAIL	77003090	Y N	0	300	8400	3000	NUMBER	300	8400
00014	SCREW	77004030	Y N	0	300	1500	3000	NUMBER	450	450
00015	NUT	77005077	Y N	0	300	6000	5000	NUMBER	450	1950

**********INV-0001********

00001	CAP	10004500	Y N	0	700	2092	3760	NUMBER	809	496
00002	HINGE	11021122	Y N	0	180	876	2400	NUMBER	320	184
00003	BRICK	11280670	Y Y	5	45	353	759	LOADS	66	83
00004	CABLE	11397278	Y N	0	470	3080	11200	FT	670	720
00005	CABLE	11397279	Y N	0	234	1080	1367	FT	270	261
00006	GLUE	21113990	N N	0	70	1133	1020	POUND	223	245
00007	GLUE	21113993	N N	0	120	434	350	POUND	174	95
00008	GLUE	21113995	Y N	0	30	675	480	POUND	45	165
00009	GLUE	21113998	Y N	0	45	222	200	POUND	51	54
00010	ASPHALT	22261100	Y N	0	5	33	100	TON	14	6
00011	ASPHALT	22261103	Y N	0	15	44	216	TON	23	9
00012	ASPHALT	22261120	Y N	0	3	27	80	TON	6	6
00013	CONCRETE	22278000	N Y	40	9	42	240	TON	15	9
00014	CEMENT	22299900	Y Y	29	36	76	240	TON	48	16
00015	CEMENT	22299901	Y Y	25	12	131	140	TON	36	21
00016	GLASS	22444500	Y N	0	340	1121	2360	SQR FT	497	241
00017	GLASS	22444500	Y N	0	0	0	2360	SQR FT	0	0
00018	GLASS	22444505	Y N	0	90	60	800	SQR FT	102	12
00019	PIPE	62661100	Y N	0	876	1666	4500	FT	962	395
00020	PIPE	62661108	Y N	0	240	258	1000	FT	258	60
00021	WIRE	62661122	N Y	8300	1600	400	90000	FT	400	100
00022	PLASTER	63334000	N Y	20	140	1373	1200	POUND	265	267
00023	PLASTER	63334001	Y N	0	60	308	300	POUND	104	66
00024	ASBESTOS	63335000	Y N	0	140	220	420	SHEET	200	40
00025	ASBESTOS	63335002	Y N	0	45	123	120	SHEET	64	26
00026	PAINT	63336400	N Y	150	0	0	1170	CAN	0	0
00027	CLAY	63337000	Y N	0	140	530	400	CAN	150	130
00028	BOLT	77001000	Y N	0	310	665	1200	NUMBER	435	135
00029	CONNECTOR	77002000	Y N	0	736	765	2000	NUMBER	808	173
00030	CONNECTOR	77002009	Y N	0	690	630	1600	NUMBER	720	150
00031	NAIL	77003000	Y N	0	200	4100	8000	NUMBER	300	1000
00032	NAIL	77003090	Y N	0	300	11100	12000	NUMBER	330	2700
00033	SCREW	77004000	Y N	0	620	2905	28000	NUMBER	1025	625
00034	SCREW	77004030	Y N	0	300	1950	12000	NUMBER	450	450
00035	NUT	77005000	Y N	0	760	12365	37000	NUMBER	2585	2635

References

Ashby, W. R. (1956). An Introduction to Cybernetics. Chapman and Hall.

Barbier, K. (1983). CP/M Assembly Language Programming. Prentice-Hall.

Beer, S. (1966). Decision and Control. John Wiley & Sons.

Beer, S. (1969). ´Prerogatives of System in Management Control´. Eighth Annual Lecture of the United Kingdom Automation Council.

Beer, S. (1972). Brain of the Firm. Allen Lane The Penguin Press.

Beer, S. (1974). Designing Freedom. John Wiley & Sons.

Beer, S. (1975). Platform for Change. John Wiley & Sons.

Beer, S. (1979). The Heart of Enterprise. John Wiley & Sons.

von Bertalanffy, L. (1956). ´General System Theory´. General Systems, volume I.

von Bertalanffy, L. (1962). ´General System Theory — A Critical Review´. General Systems, volume VII.

Ceri, S. and Pelagatti, G. (1984). Distributed Databases: Principles and Systems. McGraw Hill.

Date, C. J. (1981). An Introduction to Database Systems. Addison-Wesley.

Haas, E.B. and Ruggle, J. G. (1975). ´International Technology and International Action: who, how, and why´. International Organizations, number 29, volume 3.

Hoos, I. R. (1976). ´Engineers as analysts of social systems —a critical enquiry. Journal of Systems Engineering, number 4, volume 2.

Kremyanskiy, V. I. (1958). ´Certain peculiarities of organisms as a "system" from the point of view of physics, cybernetics, and biology´. General Systems, volume VI, 1960.

MacKay, D. M. (1957). ´Information Theory in the Study of Man´. In John Cohen (editor), Readings in Psychology. Allen & Unwin, 1964.

Maruyama, M. (1968). ´Mutual Causality in General Systems´. In Milsum, J. H. (editor), Positive Feedback, 1968.

Milsum, J. H. (1968). Positive Feedback. Pergamon Press.

Pask. G. (1961). An Approach to Cybernetics. Hutchinson & Co.

Rudall, B. H. (1981). Computers and Cybernetics. Abacus Press.

Shannon, C. E. and Weaver, W. (1962). The Mathematical Theory of Communication. University of Illinois Press.

Stewart, D. J. (1982). ´Seminars on the Physical Basis of Cybernetics´. Seminar series at Brunel University.

Wadington, C. H. (1977). Tools for Tought. Paladin.

Wiener, N. (1948). Cybernetics. Second Edition, M.I.T. and John Wiley & Sons.

DBASE II users manual. Ashton-Tate.

Digital Research CP/M Users Manual (1976).