Chapter 8
Political Information System Engineering (PISE)

8.1- Prologue Political Information System engineering (PISE) is defined as the application of interlocking set of formal techniques for the planning, analysis, design, and construction of political information systems on an CIPSE (political-computerise) basis or across a major sector of the CIPSE (political-computer centre) .

A political-organization-wide set of automated disciplines for getting the right information to political-users at the right time.

PISE applies structured techniques on a CIPSE basis, or to a larger sector of an CIPSE, rather than on a project-wide basis. PISE progresses in a top-down fashion through the following stages:

- Political-strategic systems planning
- Political-information planning
- Political-area analysis and design
- Construction.

As it progresses through these stages, PISE builds a steadily evolving repository (encyclopaedia) of knowledge about the CIPSE (political-computer-centre), its polit-data models, -process models, and -system designs.

PISE creates a framework for developing a integrated computerized political-centre. Separately developed polit-systems fit into this framework. The CIPSE approach makes it possible to achieve coordination among separately built polit-systems, and facilities the maximum use of reusable design and reusable code.

PISE helps to identify strategic-systems opportunities and achieve competitive advantage by building such systems before the competition.

PISE enables an CIPSE to get its act together. Different systems are coordinated the same political-data is represented in the same way in different polit-systems. There is integration among political-systems where needed.

PISE gives the capability to change computerized procedures quickly.

PISE facilities the building of systems of greater complexity, and the understanding and control of complex link between political systems.

The stages of a political-information engineering project list as:

Stage 1: Political-information Strategy Planning; Concerned with top management goals and critical success factors. Concerned with how technology can be used to create new opportunities or competitive advantages. A high-level overview is created of the, CIPSE its functions, political-data, and -information needs.

Stage 2: Political-Area Analysis; Concerned with processes are needed to run a selected political-area, how these processes interrelate, and what political-data is needed.

Stage 3: Political-system Design; Concerned with how selected in the polit-area are processes implemented in procedures and how these procedures work. Direct end-user involvement is needed in the design of procedures.

Stage 4: Construction; Implementation of the procedures using **/** where practical, code generations, fourth-generation languages, and end-user tools. Design is linked to construction by means of prototyping.

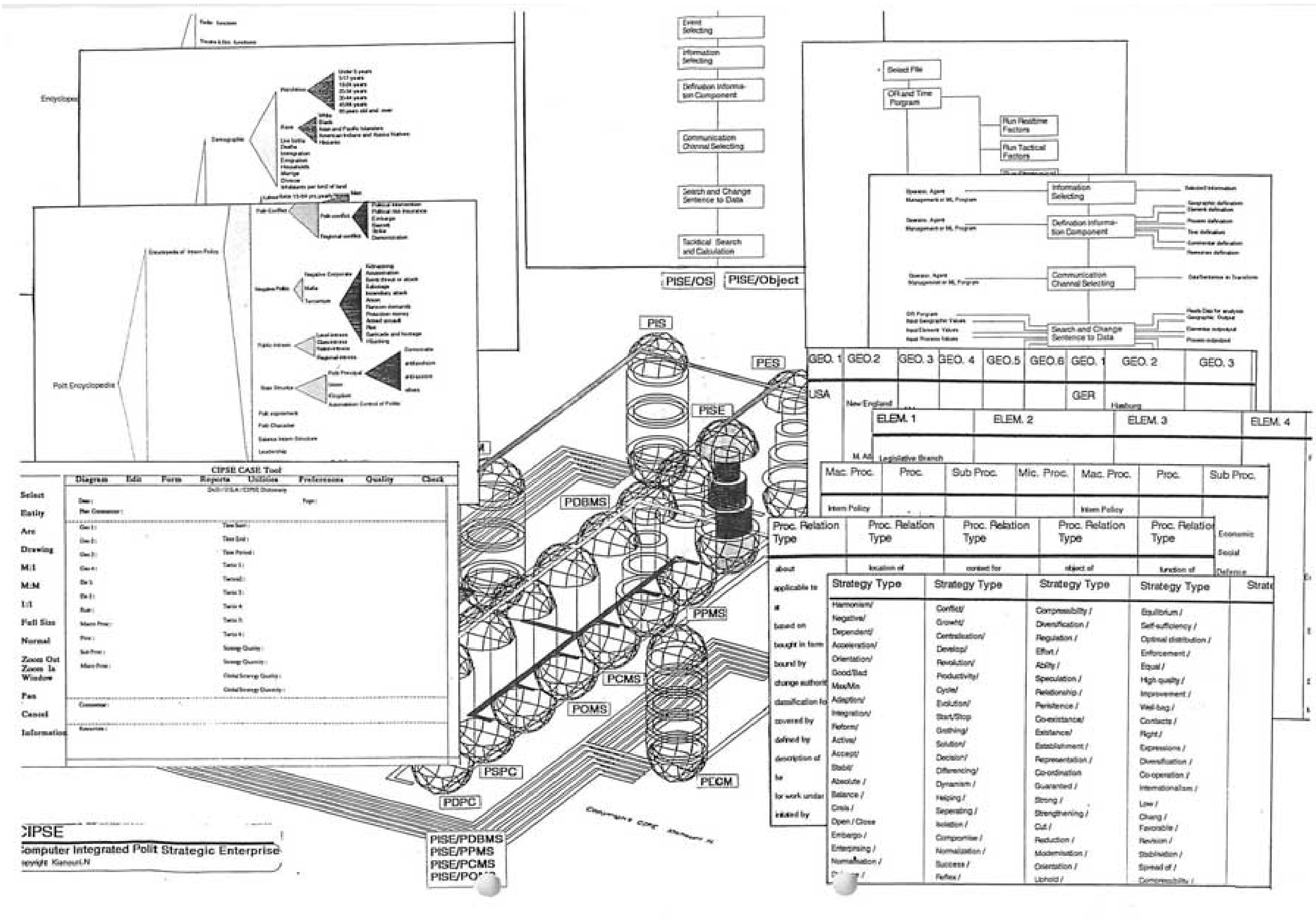

FIG.8.1. SHOWS AN INFORMATION SYSTEM ENGINEERING STATION AND INTERFACES BETWEEN CIPSE STATIONS. THE STATION HAS THE POSSIBILITY FOR:
1- USE OF CASE TOOL
2 - DIAGRAMMING
3 - USE OF POLITICAL DICTIONARY

8.2- List of Functions of Political-Information System Engineering (PISE) :

- Definition the ways for collection political-information
- Building the pyramid of polit-information
- Definition the stages and model of political-information
- Integration polit data process
- Make system measuring for political-system information
- Select software package for selecting and analysing political-data
- Select and make the special power tools for polit-information engineering
- Make software hierarchy for Programming, Planning, Design, Analysis and other cases of political software engineering
- Hardware integration aspect
- Make plan for computer hardware, receiver and broadcasting hardware, environment hardware and technical materials
- Make time protocols and Political Operation Research (POR) and Political Management (PM) protocols
- First design of intelligent polit-user interface model.

8.2.1- Elements of Political Strategy Planning:

- Organization chart
- Hierarchy of goals and critical success factors
- Function dependency diagram
- Matrices showing relations among planning information

- Data subjects
- Entity-relationship diagram.

8.2.2- Elements of political Analysis:

- Process decomposition diagram
- Process dependency model
- Process/entity matrix
- Process life-cycle diagram
- Fully normalized data model
- Data description.

8.2.3- Elements of Political Design:

- Data navigation diagram
- Dataflow diagram
- Action diagram
- Decision tree
- Dialog design
- Screen layout
- Report layout
- Prototypes
- Data structure diagram
- Data descriptions.

8.2.4- Elements of Political Construction (Build, implementation, replacement, and documentation):

- Action diagram with code
- Code generation
- Prototypes
- Program view of data
- Data description code
- Implementation, operation, polit-solution making, and documentation.

8.3- Basic Definition in PISE (Political-Information System Engineering):

Political-System - No political-organization could function without some form of polit-knowledge or -system by which means its objectives are met. The ideal polit system, however, are a combination of manual and computer processes that elegantly fit into the way the policy needs to run. Such systems will take into account the needs of many functional areas, especially where there would otherwise be duplication of data. They will also take into consideration the day-to-day and the management information needs of the polit-organization. Leading companies have such systems whose goal includes:

- Supporting the need,
- Supporting management decision making,
- Integration with other systems,
- Reacting to policy and technology change,
- Acceptance and exploitation by employees.

Polit-Data - A mass of undigested political facts. Political-data are facts or are believed to be or are said to be facts which results from the observation of political object. Political-data are objective measurement of the attributes (the characteristics) of entities (such as people, situations, conflicts, and events). These measurements are usually represented by symbols such as numbers and words, or by codes composed of a mixture of numerical, alphabetical, and other characters. However, data commonly takes a variety of forms, including numeric data, voice, and images.

Political-Information - Political-facts which are digested, analyzed, and summarized so as to be useful to decision makers. Political information is data which is used in decision-making. Political information is a relative quantity, relative to the situation, to the time at which a decision is made, and to the decision-maker and the decision-maker's background and history. The terms 'information' and 'data' are sometimes used

synonymously with 'information' supplanting 'data' in contexts where the emphasis is on the broad, grand, or useful aspects.
Polit-information is a basic resource that individuals and organizations must have to survive and succeed in today's policy.

Procedures - Procedures relate specifically to how a process is carried out. Functions and processes are concerned with what has to be done to operate an enterprise, not with how it is done. Procedures are concerned with how it is done.

Political-Information Model - An polit-information system uses the resources of hardware (machines and media), software (programs and procedures), and people (specialists and end users) to perform input, processing, output, storage, and control activities that convert data resources into political-information solution.
This information system model will help you tie together many of the political-facts and concepts involved in the study of computer based information system. It emphasizes the following major concepts:

- Hardware (machines and media), software (programs and procedures), and people (specialists and end users) are the primary resources needed to accomplish information .processing activities in information systems.
- Data resources are transformed into a variety of political information solution by the political-information processing activities of political-information systems.
- Political-information processing consists of the basic system activities of input, processing, output, storage, and control.
-

Political-System Development - The development of such systems is far from easy. There many problems to overcome in the policy, technology, tools, techniques, and the attitude of the users and existing computer development staff. Political-problems vary from region to region, but they tend to include:

- The lack of clear direction
- A rapid rate of change
- No commitment from political-senior management to systems
- Conflicting requirements
- Differences in management style.

The rapid evolution of technology has also caused problems with:

- The proliferation of micro-computer
- Hardware that has a life-span of only two or three years
- A vast range of non-integrated software
- Instant worldwide communication
- Tools emerging only recently to help control development.

Prototyping Microcomputer workstations and a variety of 4GL / CASE, and other software packages allow the rapid development and testing of working models, or prototypes, of new applications in an interactive, iterative process involving both systems analysts and political-end-users. Prototyping not only makes the development process faster and easier for systems analysts and computer programmers, but it has opened up the applications development process to end user.
Prototyping is a fast and interactive systems development methodology used for both large and small applications.

Political-View - A common term in database technology is view. A view of a database is a representation of data that is perceived by one person or program. The structure of database may be far more complex than the structure of the view. The **political-view** shows only those fields in which the political-user is interested at this time. The view is a subset of the overall database structure.

Political-Perspective - A political-perspective is a collection of political knowledge about a given activity or group of activities and the data which these activities use. It is built with diagramming techniques and tools.

Enterprise Functions - After establishing the polit-organization chart, the next step in modelling an political-enterprise is to create a chart decomposing the political-functions the enterprise performs. A function is a group

of activities that together support one aspect of furthering the mission of the CIPS-Enterprise.
Functions are determined during political-information strategy planning. An CIPS-Enterprise function is a group of activities which together support one aspect of furthering the mission of the CIPS-Enterprise.

Polit-Processes - CIPS-Enterprise functions can be further subdivided into processes. A process relates to a specific act that has a definable start and stop. A process has identifiable inputs and outputs. Processes are analyzed during political-area analysis.

- A polit-process is a specified activity that is repeatedly executed in an CIPS-Enterprise.
- A polit-process can be described in terms of inputs and outputs.
- A political-process has a definable start and stop.
- A political-process identifies what is done, not how.

8.4- Polit-System Development Life Cycle (PSDLC) - Regardless of whether analysts or key users are creating entirely new information systems or improving existing ones, they must recognize that all such systems pass through five basic stages known as the system life cycle:

- Political-Strategy (planning)
- Political-Analysis
- Political-Design
- Build stage
- Implementation
- Replacement.

The term life cycle use because, like living organisms, systems are born (planned and analyzed), grow (are designed implemented), and die (are replaced).

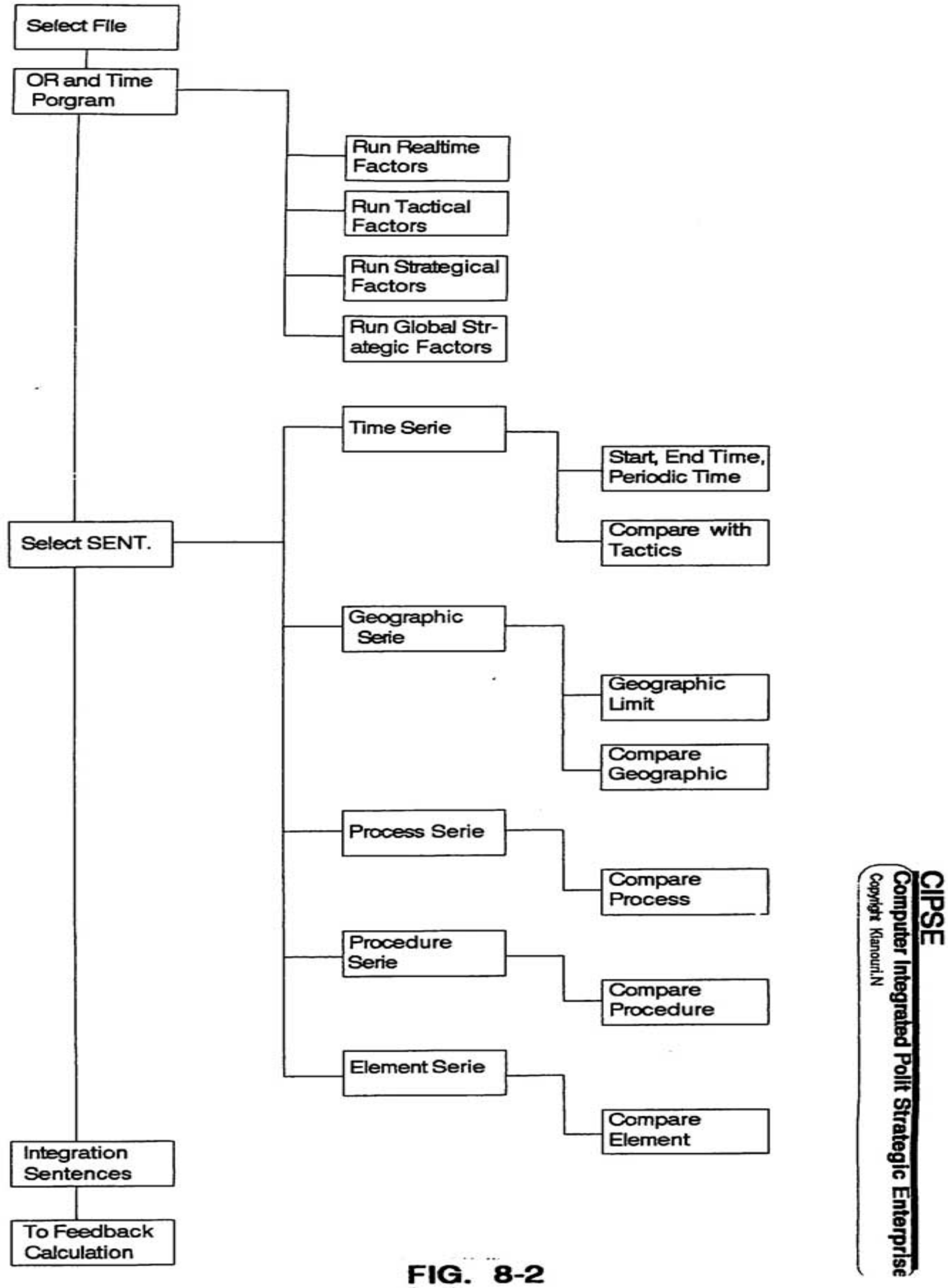

FIG. 8-2

FIG.8.2. IS A DEPENDENCY DIAGRAM OF **CIPSE** POLITICAL SOLUTION. THE DIAGRAM IS STARTED FROM SELECT INFORMATION AND FINISHED IN FEEDBACK AND TRANSFORM OF THE OUTPUTS TO POLITICAL OBJECT. THE DEPENDENCY DIAGRAM IS A METHOD FOR POLITICAL STRATEGY PLANNING.

8.5- Political-Strategy Planning - Centres on three objectives: what must be done, how to do it, and when to do it.

The success of political-systems design depends. on achieving a clear understanding of the needs of the polit-user organization, and the environment in which it operates. An understanding of the needs can only be acquired by a distinct separation of' analysis ('what is to be done') from design ('how it is to be done').

The objectives of the strategy stage is to produce, with CIPSE management, a set of policy models, a set of recommendations and an agreed plan for information systems development, which will serve the political-organization's current and future needs, while taking account of organizational, financial and technical constraints.

This objective cannot be achieved without first creating a good working relationship with all participants and subsequently gaining consensus on what forward direction is required.

A complete, detailed analysis of an organization would be an excellent basis for evolving a plan for information systems development, but would be uneconomic to prepare. Instead, a strategy is derived by doing complete, but not detailed, analysis from which a broadly-based policy model is built. Timescales are kept short, to maintain momentum and avoid results becoming out of date.

The approach to caring out a political-strategy study is to work very closely with key people to derive a common understanding of the polit-functions and -information needs of the polit organization. The approach is 'top down' and should start with the overall objectives of the political-organization. List of Tasks;

- Project administration and management
- Scope the study and agree Terms of Reference
- Plan a strategy study
- Briefings, interviews and other information gathering
- Model the policy
- Prepare for feedback session
- Conduct feedback session
- Consolidate results of feedback session
- Complete documentation of the political-model
- Evolve political-information system architecture and make other recommendations
- Determine forward system development plan
- Prepare verbal report
- Report to management
- Prepare and deliver written report.

Political-top management must work together with its political information systems executives to formulate an overall strategic vision that relates to future technology and how it could affect the policy, its solutions or services, and its goals and objectives. Formulating a strategic vision and performing strategic planning for the information needs of the political enterprise is the subject of this part.

Strategic planning of information systems trend seeks to relate the use of computers in an enterprise to top management needs and perspectives. It is concerned with formalizing management goals and critical success factors, modelling the enterprise, and strategic planning of information and its use. Particularly important, it should identify ways in which the enterprise can exchanges, or use technology differently, to achieve competitive thrusts.

A CIPS-Enterprise normally begins the information engineering process by developing an information strategy plan. The information strategy plan is concerned with the goals and targets of the polit-centre and with how technology can be used to create new opportunities or competitive advantages. Potential benefits of information strategy planning can list as:

1- to executive management:

- An assessment of the opportunities from new technology
- An assessment of how the Enterprise should be changed strategically to attack competition better
- An assessment of the competitive threats from new technology
- Adoption of the strategic polit-plan to accommodate five-year technology trends
- An assessment of the factors most critical for success
- Translation of the critical success factors into actions in building information system, decision-support systems and manager motivations, and control mechanisms
- A defined logical approach to aid in solving management control problems from a political perspective
- An evaluation of the effectiveness of current polit-Information system
- An assessment of future political-information system needs based on political related impacts and priorities
- A planned approach that will allow early return on the company's information systems investment
- Political-information systems that are relatively independent of political-organization structure.

2- To functional and operational management:

- An assessment of goals and problems, and identification of computing facilities that can help with these

- An assessment of the factors most critical for success
- Translation of these factors into actions in building appropriate
- A defined logical approach to solving management control and operational control problems
- Top management involvement in establishing organizational goals and direction.
- Increased probability of having the most valuable systems built
- Consist data to be used and shared by political-users
- Systems that are management and oriented rather than data-processing oriented.

3- To political-information systems management:
- Effective communication with top management
- Top-management support and interest in systems
- Better planning of systems that respond to political-needs
- A long-range planning base for data processing resources and funding
- Agreed-upon system priorities
- Higher probability of delivering systems those are really useful.

Political Strategic Information System is an information systems that provide a political-enterprise with competitive solution and services that give it a strategic advantage over its competitors in the marketplace. Also, information systems which promote policy innovation, improve operational efficiency, and build strategic information resources for CIPS-Enterprise.

8.6- Political System Analysis - Whether *you* want to develop a new application quickly or are involved in a long-term project, you will need to perform several basic activities of systems analysis. System analysis traditionally involves a detailed study of:
- The polit-information needs of the polit-organization and its end users.
- The activities, resources, and solution of any present political-information systems.
- The polit-information system capabilities required to meet the information needs of politicians.

System analysis canters on analyzing the present information system and include an introductory investigation, detailed investigation, and a concluding investigation. Also can define as analyzing the present system and comparing new system alternatives. Analysts begin designing new or better systems by first analyzing the basic problem areas in existing procedures. To write a problem definition, analysts must complete five steps:
- Collect data about the existing system (Reviewing policies, procedures, documents and reports- Preparing, distributing and analyzing employee questions)
- Describe and analyze the elements of the system (reviewing objectives, examining constraints, examining output, identifying input, understanding the processing, evaluating controls and feedback)
- Estimate current costs
- Device possible design alternatives
- Obtain management approval for a new design.

The analysis stage will take and verify the findings from the strategy stage and expand these into sufficient detail to ensure policy accuracy, feasibility and a sound foundation for design, within the scope of the organization and bearing in mind existing systems.
Analysts work side by side with users during this stage, establishing and checking detailed requirements. Between them, they must make certain that everything is questioned and shifted to determine the real needs and political benefits.

Lists of Tasks;
- Project administration and management
- Plan detailed analysis
- Review standards, constraints and potential design issues
- Investigate detailed requirement
- Review findings against Terms of Reference to confirm approach
- Provide detailed specification
- Provide initial transition strategy
- Define audit/control needs

- Define back-up/requirements
- Perform outline sizing and predict performance
- Review results of detailed analysis
- Obtain stage-end commitment

Analysis of data includes documenting all attributes. Analysis of the functions may involve further diagramming techniques to explore dependencies, data usage, conditions, data states or detailed logic.

Political-Goal and -Problem Analysis - An CIPS-Enterprise has certain goals. An important part of the political-information strategy planning processes is goal problem analysis, in which the analysts identify the goals of the enterprise and put them in writing. Goals are used in a control mechanism for the CIPS-Enterprise. They set targets, and the success in progressing toward those targets is measured.
Every CIPS-Enterprise has problems that make it more difficult to achieve the goals that have been set for the political organization. Sometimes a goal relates to the solution of a particular problem. Problems should be recorded along with goals.

Political-Area Analysis - A polit-area analysis study establishes a detailed framework for building a polit information-based enterprise. Political-area analysis takes one political area at a time and analyzes it in detail. It uses diagrams and matrices to record and analyze the data and activities in the CIPS-Enterprise and gives a clear understanding of the elaborate and suitable ways in which the information aspects of the CIPS-Enterprise interrelate. The diagrams and matrices are designed to be understood by management, end users, and political-information systems professionals and to greatly increase communication among these groups. Activities of polit-area analysis can define as:

- Provide a clear understanding of the policy and its activities.
- Provide an architectural framework for the building of systems in a political-information-based enterprise.
- Provide a framework such that separately built systems will work together. This framework consists of a fully normalized data model that becomes the foundation of application design and construction. A model of the polit-activities and their interdependencies.
- A linkage of the preceding models to show what processes use what data.
- Trigger the rethinking of procedures in the polit enterprise so that they are as efficient as possible for the area of desktop computers, information networks, and flexible databases.
- Identify requirements of highest priority for CIPSEnterprise activities and system design.
- Create an overview so that later design stages can proceed rapidly and coherently.

Polit-Critical Success Factor Analysis - We will use the following definitions:

- **Mission;** the mission of a CIPS-Enterprise is the highest-level statement of objectives. It gives a broad description of the purpose and policy of the political organization.
- **Strategy;** A strategy in a CIPS-Enterprise is a pattern of policies and plans that specifies how a political organization should function over a given period. A strategy may define areas for political solution development, techniques for responding to competition, means of financing, size of the political-organization, image that the CIPS-Enterprise will project, and so on.
- **Objectives:** An objective is a general statement about the direction a firm intends to take in a particular area without stating a specific target to be reached by particular points in time.
- **Political Goal;** A goal is a specific target that is intended to be reached by a given point in time. A goal is thus an operational transform of one or more objectives.
- **Critical success factors;** Critical success factors are the limited number of areas in which satisfactory results will ensure competitive performance for the individual, department, or organization. Critical success factors are the few key areas where "things must go right" for the policy to flourish and the manager's goals to be attained.

The Structured Analysis Methodologies - Many of the general-purpose analysis and design CASE tools

emphasize the structured analysis methodologies. Structured techniques in general emphasize diagrammatic and schematic designs, enhancing communications between the organization writing the requirements specification and the development team commissioned with building the software. Just as a photograph or blueprint of a building is infinitely more expressive than a written description, a structured diagram is a more easily understood mechanism for partitioning complex problems. All project members can then visualize the abstract ideas that exist in the minds of the designers.
Because of its graphical orientation, a structured specification is one of the best methods of bridging the gap between data processing professionals, who are focusing on what is being built. It is easy to give a definition of structured analysis because it is really a simple set of tools for creating a structured specification.

Sequence of Analysis - The steps in information strategy planning are ordinarily carried out in the following sequence, although the sequence of analysis can be varied according to the needs of the enterprise:

- Perform linkage analysis planning to formulate a strategic policy vision.
- Create an overview entity-relationship model of the enterprise.
- Perform technology impact analysis.
- Perform critical factor analysis.
- Perform goals and problem analysis.
- Perform policy area analysis.

Linkage Analysis Planning - For the information strategy planning process to be fully effective, it is important that top management form and clearly articulate a strategic policy vision, which must be communicated to all the executives in the CIPS-Enterprise. This strategic vision provides a framework within the strategic information planning process can operate. A methodology called linkage analysis planning (LAP) is particularly effective in helping the senior executives of an enterprise to formulate this vision.

8.7- Political System Design- Systems analysis describes what a polit-system should do to meet the information needs of polit-users. System design specifies how the system will accomplish this objective. System design consists of both logical design and physical design activities, which both produce system specifications satisfying the system requirements developed in the systems analysis stage.
System design centres on developing appropriate outputs, database, methods, procedures, data communications, inputs, and systems controls. Also can define as, Reviewing appropriate data compiled to date, determining the requirements for the new system, and designing it in detail.

The design stage will take detailed requirements from the analysis stage and find the best way of fulfilling them and achieving agreed service levels, given the technical environment and previous decisions on required levels of automation.
System Analysts is a member of PMIS department who performs systems analysis and system design work that is an integral part of a new system development process.
The design stage will take detailed requirements from the analysis stage and find the best way of fulfilling them and achieving agreed services levels, given the technical environment and previous decisions on required levels of automation.
The process of design is predominantly an iterative one, where the requirements and ideas are taken and design alternatives tried until an acceptable compromise solution is found. What is acceptable needs to be defined ahead of time, to act as objective criteria. Users, managers, operations, support and other staff may need to be presented with options for decision, to validate the designers' work and approve the final result. List of Tasks;

- Political project administration and management
- Design application
- Design and build political-database
- Produce network/communication design
- Produce audit/control needs
- Design back-up/recovery needs
- Review outline design and produce specifications program
- Complete system test plan
- Complete transition strategy
- Review results of design stage

- Obtain stage-end commitment.

After designing the problem completely, discussing possible alternatives, and providing management with enough information to make a decision, a new design will be selected. Analysts then prepare a model of the new system by using systems flowcharts and/or other design tools. Recall that there are seven components of a system-objectives, constraints, output, input, processing, control, and feedback. When analysts design a new system, they evaluate each of these components and make modifications as necessary.

- Selecting and Developing Software
- Selecting Hardware.

Tools Need - The tools needed in the workbench for design and construction are as follows;

- **Decomposition Diagrammer;** Decomposition diagrams enable a higher-level overview statement about a design to be successively decomposed into finer and finer detail.

- **Action Diagrammer;** Action diagrams facilities the building of structured procedures and structured and, again, the decomposition of a high-level overview into finer and finer detail.

- **Structure Chart Tool;** A structure chart is a form of decomposition diagram charting how program modules call submodules and what data and control information they pass to them.

- **Data Flow Diagrammer;** Data flow diagrams showing the flow data among modules of procedures or programs.

- **Data Model Diagrammer;** The data modelling tool should allow portions of the overall data model developed in BAA to be extracted for use in the design stage.

- **Data Structure Diagrammer;** The data structure diagramming tool allows appropriate parts of the data model to be represented as the structure used by a particular database management system, for example, IMS (information Management system) structures, IDMS (information database management system) structures, and relational structures.

- **Database Code Generator;** Database code, for example, IMS database descriptions, should be generated directly from the data structure diagrams.

- **Screen Painter;** A screen painter allows the screens of a computer-user dialog to be created quickly.

- **Dialog Generator;** The screens may be linked together with a dialog generator. This may be done with action diagrammer tool. The dialog generator may generate dialog that confirm to specified standards, for example, those of IBM's SAA (system Application Architecture).

- **Report Generator;** A report generator should allow the structure and layout of a report to be created quickly, along with calculations of derived fields that are in the report.

- **Code Generator;** A code generator should create executable code from the highest level specifications possible.

- **Ability to Run the Code;** If the generator has an interpreter, it should be possible to run the quickly after it has been generated. If the generator uses a compiler, it should generate the control language (such as JCL) and again provide the facilities for running the quickly, controlling the run from the workstation.

- **Test Data Generator;** Program-testing aids are needed which generate test data and facilitate a sequence of testing steps.

8.8- Build Stage - The build stage will code and test programs, using appropriate tools. These depend on the technical environment and the types of programs involved, but may range from conventional development to a 'quick build' approach using incremental development.

Staffs develop the programs in relative isolation from the user, so it is important that any alterations of the specification are recognized and checked by the user/analyst.

In addition programmer needs to make many decisions. Most of these will be minor, but collectively the wrong choice can often accidentally waste a lot of effort reworking the system. The programmers should be encouraged to really understand what they are building. They need to overview of the whole system and an indication of how their bit fits into total picture. Lists of Tasks;

- Project administration and management
- Prepare for build stage
- Review designs and estimates with programmer
- Produce programs
- Prepare, perform and review system test 6- Review test results
- Obtain stage-end commitment.

The task of building political task has a complex life-cycle, analogous to building a new office block. It is about creating and maintaining an architectural a plan and life-cycle for the understanding, requirement implementation, operation and this is coming to terms with detail what is needed for alternatives and technology.

Coding - Once the design is complete, most of the major decisions about the system have been made. However, many of the details about coding the designs, which often depend on the programming language chosen, are not specified during design. The goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim in this phase is to implement the design in the best possible manner.

The coding phase affects both testing maintenance profoundly. A well written code can reduce the testing and maintenance effort. An important concept that helps the understand ability of programs is structured programming. The goal of structured programming is to linearize the control flow in the program. That is, the text should be organized as a executed in the sequence given in the program. For structured programming, a few single-entry-exit constructs should be used. These constructs include selection (if-then-else), and iteration (while-do, repeat-until etc.). With these constructs it is possible to construct a program as a sequence of single-entry-single-exit constructs.

Testing - Testing is the major quality control measure employed during software development. Its basic function is to detect errors in the software. During requirements analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of testing is to uncover requirements, design or coding errors in the programs.

The starting point of testing is unit testing. In this a module is tested separately and is often performed by the coder himself simultaneously with the coding of the module. The purpose is to exercise the different parts of the module code to detect coding errors. After this modules are gradually integrated into subsystems, which are then integrated themselves to eventually form the entire system. During integration of modules, integration testing is performed. The goal of this testing is to detect design errors, while focussing on testing the interconnection between modules.

Unit Test and Standards - Once the major modules have been completed and debugged, stress testing can begin on the system as a whole. During this development phase, each module should be thoroughly tested on its own, generally using special testing harnesses designed to supply inputs and validate outputs to and from the module. Testing harnesses are programs that submit data to the module being tested and then examine the returned data. Often, printout, reports, and database records must be validated as well.

Once the individual modules have been tested, they are integrated into the final, whole executable system. Then the system is tested further with a test suite of cases designed to test the boundaries of the system.

Beta test Now that the system has been fully integrated, debugged, and stress tested for any performance problems, the system is ready for release to a small group of end-users in a pre-introduction field test called a beta test. The number of end-users selected for a beta test varies, depending on the type of system being deployed, and can range from a few, for very specialized applications, up to several hundred, for consumer software applications.

Beta test is a software development cycle phase where pre-release copies of the software are sent to actual end-users. Beta test releases identify trouble spots, performance problems, and bugs. Active end-user polling is conducted to provide maximum feedback to the development team.

8.9- System Transition and Implementation - The actual Installation of a new system, including its hardware and software.

The transition stage performs all tasks necessary for implementation and provides an initial period of support for the system. Transition must be accomplished with minimum disruption to exploit the new system. Finally the software writing begins. Coding should strictly follow the software design blueprint developed in the design specification phase. Depending in the design's structure and organization, many of modules may be built in parallel. The component parts have been segregated by function so that there are few module interdependencies. Each module should be designed to be treated like a black box with a well-articulated set of inputs and outputs. For testing and debugging purposes, other modules can be "stubbed out", or simulated, by testing harnesses. Lists of Tasks;

- Project administration and management
- Trains users
- Prepare for acceptance testing
- Support acceptance test
- Perform data take-on
- Carry out installation of hardware, system software, and other components of the production configuration
- Perform any other pre-implementation trails
- Prepare for cut-over
- Perform cut-over
- Support system during the critical period
- Perform post-implementation review.

8.10- Operation and Production - Operation is the last phase of the systems development cycle. At some point the project manager of the new or revised system must decide that the new system is ready for actual operations. Lists of Tasks:

- Provide operational service
- Respond to user requests
- Monitor/review performance
- Assess the future of the system.

8.11- Documentation Stage - Documentation of any system has be an art more than a science if it is to be useful and appropriate. The user documentation stage will deliver user manuals and operations hand-over documentation. These must be sufficient to support the system testing tasks in the concurrent build stage, and documentation must be completed before acceptance testing in the transition stage.

During the strategy and analysis stages, all the functions were identified and documented, including those that are wholly manual, and those that involve an interaction of automation and user procedure. This interaction was investigated further in the design stage. Outline operations hand-over documentation was also produced during the design stage and service levels agreed. All this information is now brought together to ensure that the users and operations staff are fully prepared. List of Tasks;

- Project administration and management
- Complete user documentation
- Provide operations hand-over documentation
- Obtain stage-end commitment.

Maintenance- This is an often overlooked phase of the software development cycle, for any successful software application will always be under development. If the application is eminently useful, there will be frequent suggestions for improvements; a lack of suggestions indicates the application probably isn't being used.

8.12- Diagrams for Activities - There are eight types of diagram that are important on the activities definition.

Decomposition Diagrams - A decomposition diagram is a basic tool for structured analysis and design. With decomposition diagrams, high-level activities are decomposed into lower-level activities showing more detail. This top-down structuring makes complex organizations or processes easier to comprehend. Most decomposition diagrams are simple tree structures.

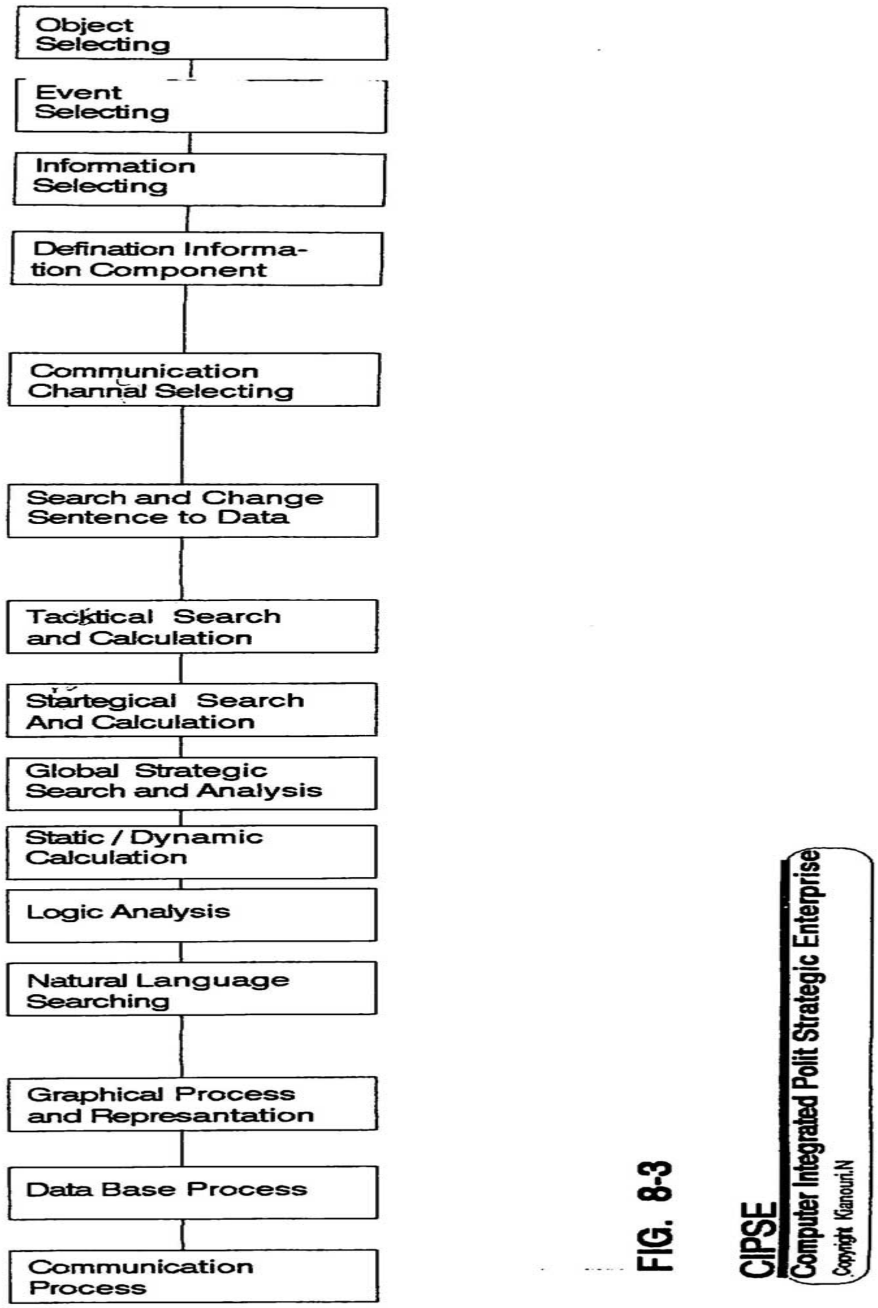

FIG.8.3. A DECOMPOSITION DIAGRAM.

Dependency Diagrams - A dependency diagram has blocks showing activities and arrows between blocks showing that one activity is dependent on another. A dependency diagram might be used for analyzing the processes that are needed in a political area.

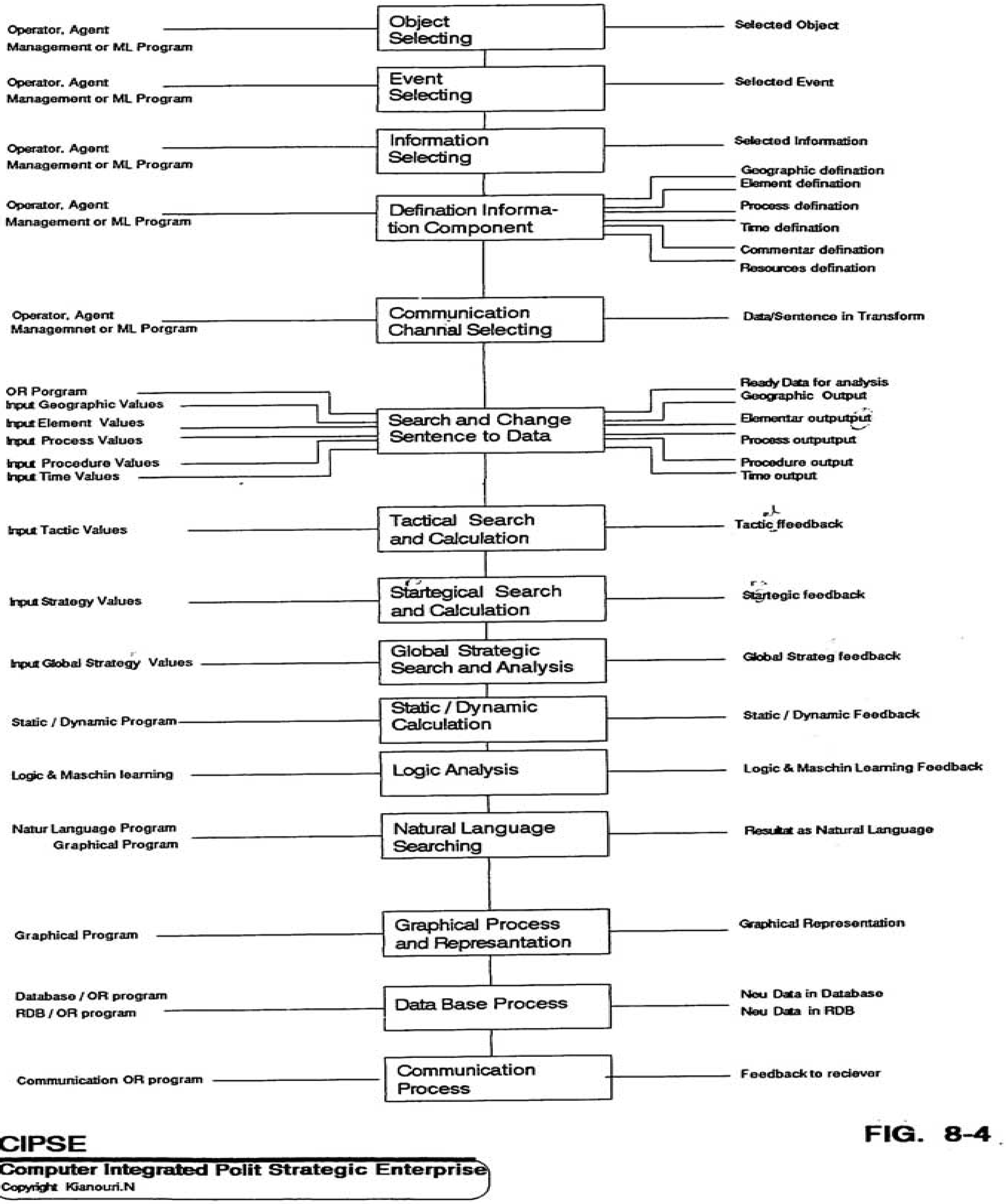

FIG.8.4. IS AN EXAMPLE FOR DATA DEPENDENCY DIAGRAMS IN A POLITICAL SOLUTION. THE PROCESS STARTED FROM SELECTION OBJECT AND ENDED IN COMMUNICATION OF RESULTS.

Data Flow Diagram - A data flow diagram shows procedures and flows of data among procedures. It is high-level design tool for mapping out the procedures required to automate a given area.
Data flow diagrams is a graphic depiction of the different data items in a system and their movement

from process to process. Data flow diagrams depict a system from the data's viewpoint rather than the control flow's viewpoint. Data flow diagrams, however, reveal only data flow, not control flow.

FIG.8.4. IS AN EXAMPLE FOR DATA FLOW DIAGRAM. FIGURE SHOWS THE NEXT STEP OF DIAGRAMMING OF POLITICAL SOLUTION. THE DATAFLOW DIAGRAM IN SYSTEM ENGINEERING IS THE MAJOR TYPE OF DIAGRAMMING AND IS THE RELATIVE DIAGRAM BETWEEN ALL FORMS OF DIAGRAMING IN INFORMATION SYSTEM ENGINEERING.
EVERY BOX OF DATAFLOW DIAGRAM HAS ONE OR MORE RELATION WITH ONE OR MORE BOXES, AND HAS ONE OR MORE INPUT AND OUTPUT. THE MAINTAIN OF BOXES SHORES THE PROCESSES. THE OUTPUT IS VALUE IS THE RESULT OF INPUT AFTER PROCESSING.

Action Diagram - Action diagrams provide a simple technique for drawing high-level decomposition diagrams, overview program structures, and detailed program control structures. Most diagramming techniques cannot draw both the overview structure of programs and the detailed control structures; action diagrams can. Action diagrams can also be used to document management procedures, we will use action diagrams to illustrate the steps involved in the information strategy planning methodologies we describe.

Data Navigation Diagrams - Data navigation diagrams show the access paths through a database structure that are used by a process or procedure. Data navigation diagrams provide a first step in charting procedures that use databases or multiple files. Appropriately drawn data navigation diagrams can be automatically converted to action diagrams using appropriate CASE tools.

Decision Trees and Decision Tables - Decision trees and decision tables provide a technique for drawing logic that involves multiple choices or complex sets of conditions.

State Transition Diagrams - State transition diagrams provide a technique for drawing complex logic that involves many possible transitions among states. Based on finite-state machine notation, neither decision trees nor transition diagrams are useful with every type of system or program; both relate to situations with certain types of complex logic.
Many type of systems are characterized by their dependence on certain activities happening at specific times. These real-time systems interact directly with a changing physical environment.

Dialog Design Diagrams - A dialog design diagram is similar to a state transition diagram. The states represent the screens or panels of a dialog, and the transitions among states represent various operator actions.

8.13- Diagrams for Data - The diagramming technique that we will use most in this chapter are those that are used for documenting data. There are three types of diagrams that are useful for documenting the data side of the information systems pyramid.

Entity-Relationship Diagrams - Entities are the people, things, or concepts about which we store data. An example of an entity is employee. Entity-relationship diagrams are used to document entities and the relationships that exist between them. Entity relationship diagrams document the high-level data models that are of interest to top management.
Data structure design, database design, and file specification form an integral part of the application design process. Consequently, it is important to graphically model data in the database and its various uses by the individual application programs. Structured analysis enlists the help of entity relationship diagrams and data dictionaries to organize the various data elements in an application's design.

Logical Data Model Diagrams - Data items are the individual elements of data that we store for a particular entity, for example employee name and department number. Logical data model diagrams, also sometimes called data analysis diagrams, consist of bubble and record diagrams that show individual data items and the dependencies that exist between them. They are used in the logical data modelling process.

Data structure Diagrams - Data structure diagrams are similar to the diagrams used in documenting the logical data model, but show detailed data structures that are used in implementing a logical data model in a specific database management system. The specific types of diagram used to document this type of structure may be different for different types of database management systems.

8.14- Changing Methods - Diagramming techniques in computing are still evolving. This is necessary because when we examine techniques in common use today, many of them have serious deficiencies. Flowcharts are falling out of use because they do not give a structured view of a program. Some of the early structured diagramming techniques need replacing because they cannot be used to represent some important types of structures.

Boxes - All the diagramming techniques we will discuss here use blocks to represent both activities and data. To distinguish between these, we will represent activities using boxes that have rounded corners, and we will show data using boxes with square corners.

Arrows - Many types of diagrams use lines that connect the various boxes that represent or data. We can use arrow on a connecting line to mean flow or sequence.

Cardinality - The term cardinality refers to how many of one item is associated with how many of another. There can be one-with-one, one-with-many, and many-with-many cardinality. We sometimes also include numbers to place upper or lower limits on cardinality.

Labelling of Lines - On some types of diagrams, the lines connecting boxes should be labelled. Lines between activity boxes are unidirectional. There may be lines in both directions between activity boxes, but these should be separate lines each with its own particular meaning.

Large Arrows - A large arrow on a diagram is used to show that an event occurs.

Diagram Connectors - A pentagon arrow is used as a connector to connect lines to a distant part of a diagram

Brackets - In an action diagram, we use a simple bracket to draw a program module that is executed a single time each time it is invoked. Brackets are the basic building blocks of action diagrams. Inside a bracket we can include a sequence of operations. A bracket can be of any length, so there is space in it for as much text or detail as is needed.

Repetition - Some brackets are used to represent repetition. A bracket that represents a module that is executed multiple times each time it is invoked has a double line at its top and bottom. A repetition structure can be used to document a program loop.

Selection - Often a program module or subroutine is executed only IF a certain condition applies. To show this, we include a description of the condition at the head of the bracket.

Case Structure - When only one of several processes is to be executed, we draw a bracket that has several divisions.
This type of bracket describe the structured programming CASE structure. One, and only one, of the divisions in the bracket is executed.

Sets of Data - Sometimes a procedure needs to be executed on all of the items in a set of items. It might be applied to all transactions or all records in a file.

Sub procedures - Sometimes a user needs to add an item to an action diagram that is itself a procedure containing actions. We call this a sub procedure, or subroutine, and draw it with a round cornered box.

Escapes - Certain conditions may cause a procedure to be terminated. They may cause the termination of the bracket in which the condition occurs, or they may cause the termination of multiple brackets. This is called an escape. Escapes are drawn with an arrow to the left through one or more brackets.

GO TO - When a language has a well-implemented escape mechanism, there is no need for GO TO instructions. Using good structured design, The GO TO can be employed to emulate an escape.

Next Iteration - In a repetition bracket a next-iteration construct is useful. With this, control skips the remaining instructions in a repetition bracket and goes to the next iteration of the loop.

Simple Database Actions - Most of the action diagrams drawn for commercial data processing systems relate to database or files. A common action of these diagrams is the database or file operation. We will

distinguish between simple and compound database actions.
A simple database action is an operation applied to one instance of one record type. There are four of simple actions:

- CREATE
- READ
- UPDATE
- DELETE

A compound database action also takes a single action against a database, but the action may affect multiple records of the same type and sometimes of more than one type. An operation may search or sort a logical file. It may be relational operation that operates on an entire table or set of tables.

Input and Output Data - The brackets of action diagrams are quick and easy to draw. If the user wants to show the data that enters and leaves a process, the bracket can be expanded into a rectangle. The square brackets may be thought of as a shorthand way of drawing rectangles. The data entering the process is written at the top right corner of the block. The data leaving is written at the bottom right corner. This type of functional decomposition is designed for computerized checking to ensure that all the inputs and outputs balance.

8.15- Entity-Relationship Modelling - An important step in information strategy planning process is to create an overview entity-relationship model of the enterprise. We begin this process by identifying organizational units, locations, functions, and entities. In performing this step, we identify where functions are carried out, what types of entities they use, how they relate to the organization chart, what organizational units are in what locations, and so on. We then create an overview entity relationship diagram that models the enterprise.

8.16- Flowcharting - Flowcharting is used in two broad areas: developing new systems and evaluating existing systems. In new systems, flowcharting serves as a map of what is to be created. Understanding a current system is a time-consuming and often difficult problem of systems design and analysis. For any system old or new, flowcharts provide pictorial information that is often difficult to communicate through language narratives alone.
Flowchart makes it easier to describe any type of system that has sequential processes, especially large, and weakness, complex one. An experienced analyst can readily see the flows, strengths, and weaknesses are frequently hidden in narrative its flowcharts. Flowcharts facilitate making changes during the planning and design stages because sections, decisions, or other elements can be substituted and redrawn more easily than a narrative description can be rewritten. In addition, they are a convenient tool for the systems analyst, and they are very useful in training new employees.

Flowchart Tools - Analysts use a system flowchart to depict system elements. The systems flowchart, like its more detailed counterpart, the program flowchart, shows the relationships between input, processing, and output, in terms of the system as a whole. It is a general representation of the information flow within the total system.

Flowchart pictorial representation of data flow; It aids in understanding what is happening, is an effective communicator to those using them, and serves as a permanent record. There are three major categories of standard flowchart symbols.

1- Basic symbols:

Input/output	annotation
Process	connector
Flow line	terminal

2- Input/output and file symbols:

Punched card	display
Punched tape	communication link
Document	on-line storage
Magnetic tape	on-line
Manual	input

3- Processing symbol

Predefined process	manual operation
Auxiliary operation	decision

Program flowchart; Use the basic symbols and the following processing symbols-predefined process and decision.

System flowchart; Uses the input/output and file symbols and the following processing symbols-auxiliary operation and manual operation. Flowcharts types can divide in three groups:

1- System flowchart or Procedural flowchart; Depicts the flow of data through the major parts of a system with a minimum of detail. For the most part, it shows where input enters the system, how it is processed and controlled, and how it leaves the system to storage and/or output.

2- Flow process chart; Shows the sequence of operations by functions and individuals, detailed operations can be analyzed to determine ways of improving them.

3- Document flowchart; Traces input data through each phase of processing and communication into files and, finally, out of files as output to satisfy managerial and operational personnel needs.

The systems analyst uses a document flowchart to trace the flow of documents and reports for that function through the system from organization to destination.

8.17- PISE/CASE (Computer Aided Systems Engineering) - A software technology that provides an automated, engineering discipline for software development, maintenance, and project management; includes automated structured methodologies and automated tools. A computer-aided systems engineering (CASE) process has emerged due to the availability of fourth-generation languages (4GL) and a variety of software packages for system and software development. CASE software (tools) should perform the following functions:

- Enable the polit-user to draw diagrams for planning, analysis, or design on a workstation screen.
- Information about the political-objects in the diagram and relationships among the objects so that a complete set of information is built up
- Store the meaning of the diagram, rather than the diagram itself, in a repository
- Check the diagram for accuracy, integrity, and completeness. The diagram types used should be chosen to facilities this
- Enable the user to draw programs with diagrams, showing conditions, loops, CASE structure, and other constructs of structural programming
- Enforce structured modelling and design of a type that enables accuracy and consistency checks to be as complete as possible
- Coordinate the information on multiple diagrams, checking that they are consistent, and together have accuracy, integrity, and completeness
- Store the information built up at workstations in a central repository shared by all analysts and designers
- Coordinate the information in the central repository, ensuring consistency among the work of all analysts and designers.

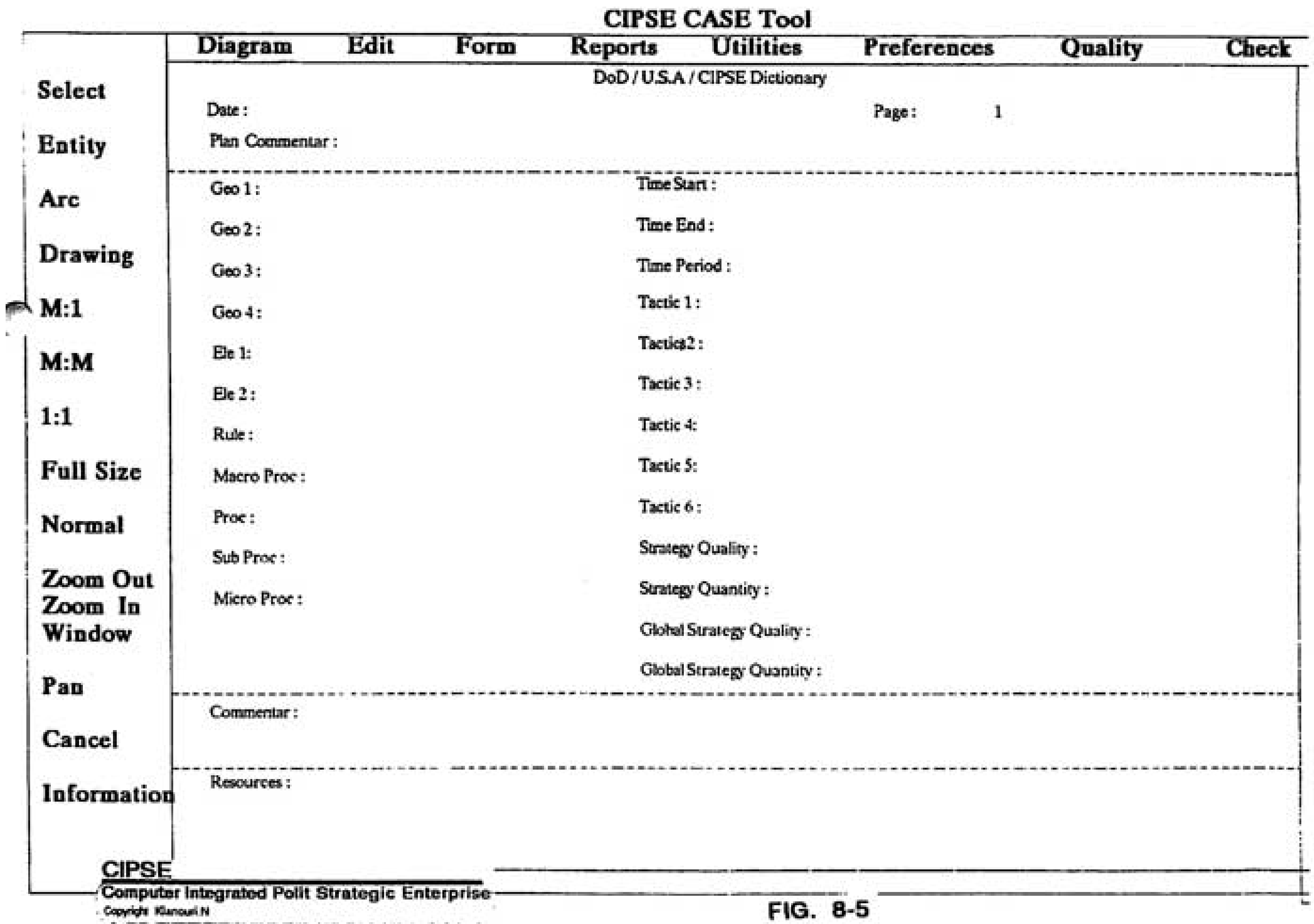

FIG.8.5. IS A REPRESENTATION OF CASE TOOLS THAT SPECIAL HAS CONSTRUCT FOR USE OF POLITICAL SYSTEM ENGINEERING. THE CIPE-CASE METHOD HAS THE FOLLOWING COMPONENTS:
1- POLITICAL-PROCESS DEFINATION
2 - POLITICAL-PROCEDURE DEFINATION
3 - DIAGRAMMING METHODS (DATAFLOW, DEPENDENCY, DECOMPOSITION, E - R AND ETC.)
4- CREATING AUTOMATIC PROCEDURE FOR INTERFACING TO OTHER CIPSE STATIONS, FOR EXAMPLE DATABASE COMMUNICATION, STATICAL, DYNAMICAL AND LOGICAL ANALYSIS, AND PROGRAMMING.
5 - MANY OTHER COMMON CASE TOOLS FUNCTIONS (EDIT, MOVE, ZOOM, FILE, COPY, AND ETC.).

8.18- User Interface Design Methodologies - The user interface is the most important and critical portion of a modern application program. It is the only part end-users can see. Users cannot peer inside the software package to marvel at the elegance and efficiency of the internal design; they can see only what is in front of them on the screen. If that on-screen view is clumsy, slow, unresponsive, or hard to learn, the application will be judged to be of flow quality. The user interface is usually the last component to be designed.

User interface are a difficult and time-consuming part of a software application to design and implement. Getting a user interface design right means constant end-user involvement from the beginning of the requirements analysis phase.

Most modern software applications emphasize the direct manipulation of data. Direct manipulation not only displays the data in a convenient and logical format; it also allows the user to place the cursor on the object and alter its value simply by entering a new one. There are at least as many ways to display information visually as there are application programs. The personal computer software industry offers a gold mine of different visual display metaphors.

The following guidelines, adapted from the Apple Human Interface Design Guidelines, are widely accepted:

- Real world metaphors
- Selection rather than remembering
- Consistency

- User control
- Feedback
- Forgiveness
- Perceived stability.

Real world metaphors means basing the user view of the system on things in the real world rather than on things internal to the computer system. For example, input screens should be designed to reflect the paper forms that use in real life rather than to reflect the structure of the database or files inside the computer.
Selection rather than remembering means that users should have to remember the names of commands, files, and the like. This is usually dealt with by using menus. The user should never be prompted for information that he or she must remember about the system. For example, a module to maintain entities that gives the users a choice of add, change, and delete should not ask for the name of the file to be used: it should instead provide a list of valid files.
Consistency means that all screens and menus should look work as similarly as possible. If you use functions keys, the same key should mean the same thing for each menu.
User control means that users should be able to choose what they want to do in the order in which they want to do it. For example, a system should not require that all new records be added before any existing records can be changed.
Feedback means that users should be kept informed of what the system is doing. For example, screens should have titles so that users know that the correct screen has selected.

Perceived stability means that users should not notice changes to the system unless those changes impact the user's tasks. For example, adding new transactions or new entities should not impact the structures of existing menus and absolutely should not change the meaning of user inputs to carry out other tasks.

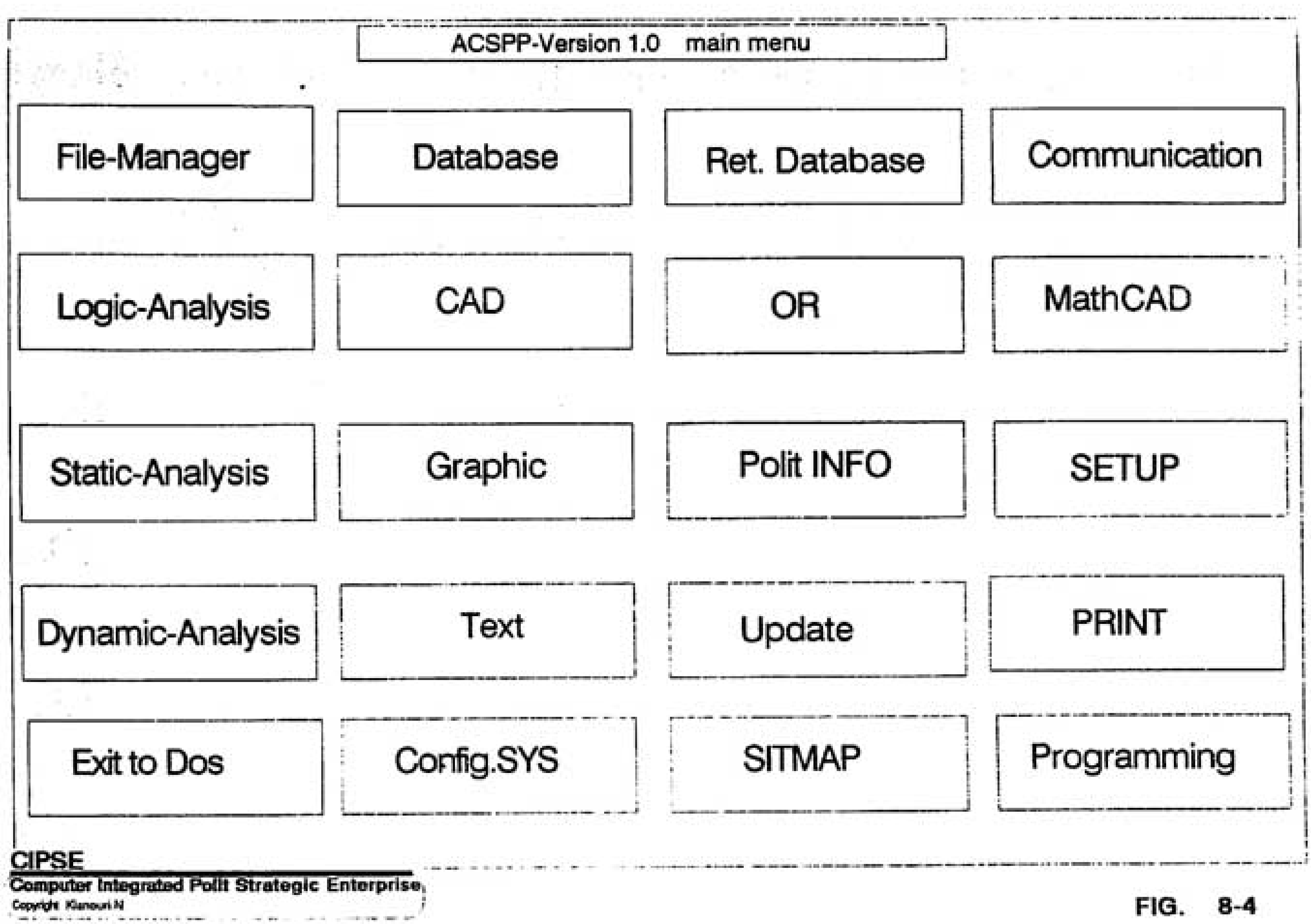

FIG.8.8. IS AN EXAMPLE FOR A MENU OF USER INTERFACE FOR ACSPP PACKAGE. ACSPP MENU IS A MULTIFUNCTIONAL PROGRAM WITH POSSIBILITY OF INTERFACING AND TRANSFORM OF DATA BETWEEN DIFFERENT MODUL OF

Representation Capability - The application of graphics, both for visualisation and interaction, requires a wide variety of features in the user interface, e.g.

- Icons
- Unstructured figures
- Structured pictures
- Miniatures
- Windows, sub windows
- Fixed and arbitrary multi-windows
- Transient sub windows
- pop-up menu
- Intelligent fields.

Political-solution enterprise management systems themselves give rise to some application-oriented features in representation capability, such as:

- Forms
- Intelligent forms
- Networks (e.g. CPM and Gantt charts)
- Rules, regulators, and associated yardsticks, etc. (in knowledge-based systems).

8.19- Interfaces of PISE with other CIPSE Stations:

8.19.1- PISE/CIPSE (Integration of Political Information System Engineering in CIPSE, and Developing of CIPSE):

This stage will be the daily work of CIPSE, also in a development centre intern the CIPSE, after establishing management and engineering stages.

Developing information system solutions to political problems is a major responsibility of managerial end users. Developing CIPSE means managing activities such as systems analysis and design, prototyping, applications programming, project management, quality assurance, and systems maintenance for all major systems development projects. Planning, organizing, and controlling the systems development function of an information services department is a major managerial responsibility.

In addition, many systems development groups have established development centres staffed with consultants to the professional programmers and systems analysts in their organizations.

The task of engineering of CIPSE as system has a complex lifecycle, analogous to engineering of political-system. It is about creating and maintaining an architectural model of what is required; following a plan and life-cycle for the system, which goes through stages of understanding, requirement, definition, design, construction, implementation, operation and review. The key to success in all of this is coming to terms with the political real world, and understanding in detail what is needed for the policy before selecting design alternatives and technology.

8.19.1.1- Engineering Policy (Political-Object, -Area) as System:

A hierarchical political system or political-knowledge-base define as a group of interrelated or interacting political elements forming a unified whole.

The most important components for political knowledge of CIPSE list as:

- Political-system(international-system and intern-system),
- political-economic-system,
- political-social-system,
- political-justice-system,
- political-military-system,
- political-geographic and environment-system,
- political-science-system,
- political-communication-system,
- Political-organization-system.

The Political-system through four basic stages known as the **system life cycle** will be engineered:

- Planning,
- Analysis,
- Design,
- Construction and implementation.

8.19.1.1.1- Political-information Strategy Planning; Concerned with top management goals and critical success factors. Concerned with identification of political problems, opportunities, symptoms, summary and evaluation of alternative solutions. List as:

- Separate major political problems and opportunities from their symptoms. Use the following system contexts to help reveal the problems or opportunities that are present:
  a) Describe the organizational system and its environmental systems.
  b) Identify affected organizational subsystems and their components.
  c) Define the objectives, standards, and constraints of the organization and affected subsystems.
- Briefly state the major problems or opportunities facing the CIPSE organization.
- Briefly identify several alternative solutions using evaluation criteria that reveal their advantages and disadvantages.
- Determine whether a political problem or opportunity exists.
- To determine whether a new or improved information system is needed.

8.19.1.1.2- Polit-system (Area or Object) Analysis; Concerned with processes are needed to run a selected political-area, how these processes interrelate, and what polit-data is needed. Concerned with how system engineering technology can be used to create new opportunities or competitive advantages. A high-level overview is created of the, political-computer-centre its functions, political-data, and -information needs. The tasks of polit-system analysis list as:

- Identify the main characteristic of political-systems, subsystems and processes.
- Determine the main output requirements, including response times.
- Analyse the organization chart, geographical distribution, etc., of the CIPSE department(s) involved.
- Consider possible alternative ways of meeting the requirements of user(s).
- Document the feasibility study in report for user and CIPSE management.

Determine that the requirement of the political-system is consistent with the CIPSE objectives.

8.19.1.1.3- Political-System Design; Concerned with how selected processes in the polit-area are implemented in procedures and how these procedures work. Direct end-user involvement is needed in the design of procedures.

- Propose designs for any new or improved information systems required by the selected solution. Use one or more tools of analysis proposal.
- Develop specifications for the hardware (machines and media), software (programs and procedures), people (specialist and end users), data resources, and information products that will satisfy the information needs of end users.

8.19.1.1.4- Construction and implementation; Implementation of the procedures using, where practical, code generators, fourth generation languages, and end-user tools. Design is linked to construction by means of prototyping.

- Propose an implementation plan for the selected political solution.
- Acquire (or develop) and install hardware and software.
- Test and document the system.
- Train people to operate and use the system.
- Convert to the new system.
- Use a post implementation review process to monitor, evaluate, and modify the political-system as needed.

8.19.1.2- Engineering CIPSE (Computer Integrated Political strategic Enterprise) as System:
CIPSE involves the use of the computer to tie together or "integrate" all the movements of data and solution in

the political enterprise under the control of one complete enterprising-system.
Computer integrated political strategic enterprise (CIPSE) encompasses the overall enterprise controls, organization, methodology, computers, and computer networks for the complete political-enterprise control and integration.
In the context of CIPSE, the time allowed for the creation of an overall strategy plan should be at least one year.
CIPSE from the **system approach** is a defined and interacting collection of political real world facts, procedures and processes, along with the organized deployment of people, machines and other resources that carry out those procedures and processes.
The task of engineering of CIPSE as system has a complex lifecycle, analogous to engineering of political-system. It is about creating and maintaining an architectural model of what is required; following a plan and life-cycle for the system, which goes through stages of understanding, requirement, definition, design, construction, implementation, operation and review. The key to success in all of this is coming to terms with the political real world, and understanding in detail what is needed for the policy before selecting design alternatives and technology. Part of the goals for engineering of CIPSE as a system list as:

- Definition the ways for collection political-information
- Building the pyramid of polit-information
- Definition the stages and model of political-information
- Integration political data process
- Make system measuring for political-system information
- Select software package for selecting and analysing polit-data
- Select and make the special power tools for polit-information engineering
- Make software hierarchy for Programming, Planning, Design, Analysis and other cases of political software engineering
- Hardware integration aspect
- Make plan for computer hardware, receiver and broadcasting hardware, environment hardware and technical materials
- Make time protocols and Political Operation Research (POR) and political management (PM) protocols
- First design of intelligent polit-user interface model.

8.19.2- CAPx / PISE - The design process can be broken down into the following phases:

- **Conception** (specification analysis, compilation of political-solution variants, assessment of the political solution),
- **Development** (specification of the political-solution concept, scale design, model construction, assessment of the political-solutions),
- **Detail** (representation of individual parts, assessment of the political-solutions). Large political-design systems also incorporate libraries of standard symbols, dimensioning systems, political-geographic Atlas formats, and data bases for the storage and retrieval of designs.

8.19.3- CAPSIM / PISE (System Engineering of CAPSIM) - There are four stages in the construction of a simulation model. The process involved has been explained in detail elsewhere. The stages are:

- Problem definition and data collection. This consists of the analysis to determine the scope the model and to establish the assumptions to be made. The data need is then established and collected. This can cause major difficulties. Especially where proposed systems are concerned. For example, realistic reliability statistics for modern machine tools are notably difficult to obtain.
- The construction and refinement of the model. This is probably the easiest and most rapid part of the process. Major problems at stage are almost always caused by inadequate attention to problem definition and data collection.
- Verification and validation of the model. Verification is the process of checking to see that the model functionally meets its specification. Validation is the process of ensuring that the model does what is intended. This is usually achieved by comparing the results of several runs of the model with each other and with real data where this is available. It is at this stage that any assumptions not previously considered often reveal themselves. The objectives of this stage are to generate confidence in the model and to assess its level of meeting reality.
- Model experimentation and results analysis.

Political Enterprise Job Scheduling - A CIPSE-job is a political consisting of a set of CIPSE workplaces operating independently. This set of workplaces may be divided into subgroups of workplaces which have the same characteristics. A job is a unit of output produced by a series of operations performed on workplaces in a specified sequence. This sequence is called the routing or the technical sequence of the CIPSE-job.

When scheduling jobs through the CIPSE, the objective is to ensure that the operations are done in the proper sequence. There are two classes of CIPSE-job problem:

1- The statistically case, in which all jobs are on hand at time zero.

2- The dynamically case, in which job arrivals vary with time.

Many scheduling rules have been suggested, including:

1- **First in-first out (FIFO):** the CIPSE-job with the earliest arrival time in the queue is selected.

2- **Highest priority:** each CIPSE-job is assigned a priority on arrival. The Job with highest priority gets the workstations first.

3- **Shortest processing time:** the job that requires minimum workplace time is chosen from the queue of waiting jobs.

4- **Smallest remaining job slack:** the job that has minimum job slack is selected. The job slack, or remaining free time, is defined as the due date, minus the current time, or remaining free time, is defined as the due date, minus the current time, minus the sum of the remaining processing times for that job.

5- **Earliest job due date:** the job with the earliest due date is chosen. This is similar to (4) above but does not include processing time in the scheduling rule.

8.19.4- ACSPP / MPIS (Managing Information system of ACSPP) - Petri Nets and Event Graphs - Petri nets are very good tools for representing, analyzing, and simulating political systems because they have the following features:

1- They provide a graphical way of visually representing a enterprising system.

2- They capture precedence relations and structural couplings among concurrent or asynchronous unpredictable discrete events.

3- They can easily model existing deadlocks, conflicts, and storage sizes.

There exist several extended PN models (PNs, colored PNs, stochastic PNs, fuzzy PNs, etc.) that cover a large variety of enterprising situations and problems.

Petri nets theory aims at an understanding of systems whose structure and behaviour are determined by the combinatorial nature of their states and changes. It studies such systems at different conceptual levels, in various degrees of detail, and in many areas of application. One important branch of research in net theory is concerned with the conceptual and mathematical foundation of an update notion of dynamic system and its different ways of presentation (its model). The basic proposed by Petri as the common reference model of net theory. Other models are considered theoretical in a strict sense if they are derived from or, can be translated into the basic model.

If a dynamic system has an adequate representation in a net theoretical model, we call its Petri system. In this paper we give "an example of introducing a net theoretical system model which is called predicate/transition nets (Petri-nets).

8.19.5- SCPP / PISE - The methods of statistically process control can provide significant payback to those companies that can successfully implement them. While SCPP seems to be a collection of statistically based problem-solving tools, there is more to the successful use of SCPP than learning and using these tools. Management involvement and commitment to the quality-improvement process is the most vital component of SCPP's potential success.

Statistical process control methods and experimental design, two very powerful tools for the improvement and optimization of processes, are closely interrelated. Experimental design is a active statistical method: we will actually perform a series of tests on the process making changes in the inputs and observing the corresponding changes in the outputs, and this will produce information that can lead to process improvement.

Experimental design methods can also be very useful in establishing statistical control of a process. Experimental design is a critically important engineering tool for improving a enterprising process. It also has extensive application in the development new processes. Application of these techniques early in process development can result in:

- Improved yield.
- Reduced variability and closer conformance to nominal.
- Reduced development time.
- Reduced overall costs.

Experimental design methods can also play a major role in engineering design activities, where new political solution are developed and existing ones improved. some applications of statistical experimental design in engineering design include:

- Evaluation and comparison of basic design configuration.
- Evaluation of material alternatives.
- Determination of key political solution design parameters that impact performance.

Use of experimental design in these areas can result in improved enterprise ability of the political solution, enhanced field performance and reliability, lower polit-solution cost, and shorter political solution development time.

Guidelines for Designing Experiments - Experimental design methods are a powerful approach to improving a process. In order to use this approach, it is necessary that everyone involved in the experiment have a clear idea in advance of the objective of the experiment, exactly what factors are to be studied, how the experiment is to conducted, and at least a qualitative understanding of how the data will be analyzed.

- Recognition of and statement of the problem
- Choice of factors and levels
- Selection of the response variable
- Choice of experimental design
- Performing the experiment
- Data analysis
- Conclusions and recommendations.

Quality improvement tools for SCPP can list as:

1- Process Flow Diagram;

- Expresses detailed knowledge of the process Identifies process flow and interaction among the process steps
- Identifies potential control points.

2- Cause and Effect (Fishbone) Diagram;

- All contributing factors and their relationship are displayed
- Identifies problem areas where data can be controlled and analyzed.

3- Control Chart;

- Helps reduce variability

- Monitors performance over time
- Allows process corrections to prevent rejections

Trends and out-of-control conditions are immediately detected.

4- Check sheet;

- Simplifies data collection and analysis
- Spots problem areas by frequency of location, type, or cause.

5- Pareto Diagram;

- Identifies most significant problems to be worked first
- Historically 80% of the problems are due to 20% of the factors
- Shows the vital few.

6- Scatter Plot;

- Identifies the relationship between two variables
- A positive, negative, or no relationship can be easily detected.

7- Design of Experiment (DOE);

- Useful in process development and troubleshooting
- Identifies magnitude and direction of important process variable effects
- Greatly reduces the number of runs required to perform an experiment
- Identified interaction among process variables
- Useful in engineering design and development
- Focuses on optimizing process performance.

8- Histogram;

- The shape shows the nature of the distribution of the data

The central tendency (average) and variability are easily seen specification limits can be used to display the capability of the process.

8.19.6- CDPP / PISE (Engineering User Interface for CDPP) - The four key stages involved in the implementations of dynamic control are:

- Understanding of dynamic behaviour of the political object; this includes an assessment of the nonlinearities and the determination of time constants.
- Specification of the political problem; It is necessary to consider dynamic control and supervisory control in conjunction with the need for additional tasks.
- Selection of the programming or integrated package; this is governed by the choice of sampling are major concerns.
- Choice of control strategy and tuning to achieve the desired performance.

The CDPP package must provide the user interface with the opportunity to customise the appearance of the GUI in such factors as colours and fonts, and also control functions such as keyboard focus policy and mouse button functions. The user is allowed to select the best mechanism for operating a function according to:

8.19.7- PISE / PEDBKS (Engineering User Interface ;or PEDBMS) - More and more organizations are looking to computer-aided software engineering (CASE) tools to improve the effectiveness of analysts and designers, increase the role of end users in systems designs, reduce programming and maintained time, and manage data more effectively.

CASE tools have become very important tools for managers to use in their efforts to manage data more effectively and prudently exploit the data resource.

Many application programs access database. These are several tools available to help you interface your program to various existing databases. Some also help you design the databases themselves, if necessary.

8.19.8- PISE / PEPMS (Engineering User Interface for PEPMS) Designing user interfaces for a Political enterprise project and planning management.

Political enterprise project management is a social process of working with people through the application of the key management functions of planning, organizing, motivating, directing, and controlling.
Project management is an approach for responding to the dynamic nature of the flow of projects in an political organization. Since complex projects are part of strategies to deal with complex problems of organizations, there is a real need to develop management techniques and devices which address themselves to the dynamic nature of projects. Political project life cycle consisting of these five phases:

- Level of experience; A totally new user may wish to interact through menus; an expert may prefer command 'short-cuts'
- Personal preference; A left-handed user prefers a different assignment of mouse button functions
- Special requirements; With a non-standard keyboard one has to modify default function keys for copy, undo, help and so on
- Familiarity; One may have strong background of previous experience with another GUI, and would like the new working environment to be familiar as possible.

Political project life cycle also can consisting of these four phases:

1- **The conceptual phase;** during this phase, the technical, military, and economic bases are established, and the management approach is formulated. The tasks list as:

- Determine that a project is needed
- Establish goals
- Estimate the resources the organization is willing to commit "Sell" the organization on the need for a project organization
- Make key personnel appointments.
-

2- **The validation phase;** during this phase, major program characteristics are validated and refined, and program risks and costs are assessed, resolved, or minimized. An affirmative decision concerning further work is sough when the success and cost realism become sufficient to warrant progression to the next phase. The tasks list as:

- Define the project organization approach
- Define project targets
- Prepare schedule for execution phase
- Define and allocate tasks and resources
- Build the project team.

3- **The full-scale development phase;** in the third phase, the design, fabrication, and testing are completed. Tasks list as:

Perform the work of the project, (i.e., design, construction, production, site activation, testing, delivery, etc.)

4- **The production phase;** in this period, the system is produced and delivered as an effective, economical, and supportable system. Tasks list as:

- Assist in transfer of project product
- Transfer human and nonhuman resources to other organizations
- Transfer or complete commitments
- Terminate project
- Reward personnel.

8.19. 9- PISE/PECKS (Engineering User Interface for PECKS) - The qulaity polit information system is a set of tools to trend and analyze network availability, queuing/delays, call setup times, bit error rates, and other measures associated with "world class" telecommunications service delivery. The set of applications envisioned will provide their basis for measurement of quality of service for both internal and external consumption.
The ability to monitor selected parameters within the network was one of the capabilities found in all of the network management systems. Such monitoring will result in a very large amount of data as the network management system periodically reads values from the many agents in a large network. The fundamental point of this monitoring is to analyze the collected data to gain insight about the network's internal activity.

8.19.10- PISE/PEOMS (Organization of PISE, and the Political Organization Chart) - The first step of enterprise modelling is to create a date-to-date organization chart. For a large political enterprise, the organization chart has many boxes on it. The boxes of the organization chart are called objects about which we may store data. If we are using information engineering techniques for the creation of systems, an object is an entity about which data is stored in the encyclopaedia. In this chapter, we refer to four types of objects:

- **Organizational Unit;** A division, department, or functional area of the enterprise that is individually managed and distinct from other organizational units.
- **Function;** A group of activities that together support one aspect of furthering the mission of the enterprise.
- **Location;** A place at which functions are performed.
- **Entity;** A person, thing, or abstract concept about which we store data.

Throughout the political-information strategy planning process, we collect political-information about each object that we identify during enterprise modelling. For example, where the object is organizational unit, we may collect polit-information about who manages a particular organizational unit. We may also want to collect related information, such as location, goals, or critical success factors for that organizational unit.
For political system engineering we use the jobs as:

- **System analyst -** Works with users to define information requirements
- **System designer -** Designs and chooses alternative systems to perform tasks specified by analysts
- **Application programmer -** Designs, codes, and tests computer programs based on the systems analyst's specifications
- **Maintenance programmer -** Enhances and makes changes in existing programs based on the system analyst's specifications
- **Systems programmer -** Maintains operating system software that controls the schedule and flow of application programs
- **Operator -** Operates the mainframe computer
- **End users -** Skilled users of information systems who help information systems specialists define the information requirements

Engineering User Interface for PEOMS - People are the most important resources within a political centre. Political personnel affected by the CIPSE are presented with three primary tools: Organizational Models, the "Everyone's Information System", and a political character Information System.

As a result, the initial focus area is to provide organizational modelling tools which assist in clarification of roles and responsibilities, promote ownership of assigned tasks, and create structures that foster teamwork. The types of tools that exist or are in development include concepts such as a "living" Organizational Chart, an Organizational Capacity Model, a skills inventory, and a quarterly achievement database. These tools provide CIPSE with an organizational architecture which begins to define clear roles and responsibilities, core competencies and skills required to perform an assigned role; tools which match existing skills to prioritized projects and a constant feedback of individual and team performance.

Information is key to the success of an individual as well as an organization. The CIPSE embraces the concept of the Executive Information System and extends it to everyone wi thin the organization, thus the "Everyone's Information System“. It provides an information system that allows appropriate access enterprise wide and provides appropriate levels of detail and summarization at the various levels of management.

CIPSE is a provider of telecommunications services throughout the world. Consequently, CIPSE’s knowledge database of its polit characters and the various key data about political character psychology is its most empowering feature.

8.19.11- PISE/PIS Artificial Intelligence (Intelligence PISE) Research and development of artificial intelligence (AI) systems for human machine interfaces have focused on natural language (NL) text, speech, and graphics primarily in isolation, rather than in integrated interfaces. With the increased functionality and reliability provided by new developments in interface devices such as speech recognition and production systems, high-resolution colour and monochrome graphic displays, and pointing devices, as well as the availability of increasingly powerful workstations environments, it is a natural and timely step in the evolution of human-computer interfaces that the media be integrated to meet the information processing needs of the user community. Lists of characteristics of an intelligent interface are:

- Adaptability
- Ability to Learn
- Smartness about use of resources
- Ability to Anticipate action
- Ability to assist the user
- Ability to teach
- Ability to explain action
- Symbolic of applications.
-

Each type of intelligent behaviour can be supported by the application and the interface. The job of developing software for enterprising applications is often more time-consuming and expensive than the hardware involved. Automating either the handling of political real objects in political solution making or the data controls the operation requires a great deal of software. To minimize the time and effort involved in programming, more efficient software development tools must be provided. As we have already seen, expert-system shells can be used in a variety of applications, avoiding the need to develop system software. Automated programming techniques using very high level definition languages and intelligent compilers can also reduce and simplify the programming effort significantly. In addition, AI-based compilers can generate more efficient assembly code than standard compilers, resulting in greater productivity from the computer resources being used.

8.19.12. PISE / PES (Expert PISE) - In this paragraph we concern ourselves with the process of building knowledge-based system and we focus on the issues of how such a system is implemented or built. Among the implementation questions we address are the following:

- How is a political-knowledge-based system developed?
- What are the stages of the development process?
- What tasks are suitable for encapsulation within an expert system?
- How is knowledge acquired? How is the expertise elicited?
- Can more than one expert be used?
- How is the expertise of several experts combined?
- How are KBESs tested, verified, and validated?
- How are KBESs maintained and supported?
- What are the issues for the organizations that undertake the building of expert systems?
- Are expert systems expensive? How is access to the experts guaranteed?
- How important is the commitment of management? What are the advantages (i.e., community memory, standardization, training, and education) to others?

How do we identify the beginning of a political expert project? In a model that is now almost traditional, these five stages are characterized as follow:

- Identification; In which we characterize the most important aspects of the problem and set goals for the entire project,
- Conceptualization; During which we make explicit the key attributes of the task nd its domain and give some thought to knowledge representation issues,
- Formalization; In which we develop a formal model of the task and outline representation scheme (s) for the task, its attributes, and their relationships,
- Implementation; During which time we program the representation scheme using whatever tool (programming language, shell, etc.) we have chosen for the project; and
- Testing; In which we exercise the prototype system just implemented against some of the case studies from a library of solutions. After the testing phase, or rather, within the testing phase, we implement feedback loops which will lead to further refinement the prototype or even a complete reformulation.

8.19.13- PISE/PESM (Management Software of PISE) - The role of Standards in the Political Information Engineering - CASE in the Software Development Process - Software engineering

methodology is a set of integrated approaches, rules, procedures, methods, tools, and environment. A good methodology provides a series of steps that directs the engineering effort from start to finish. There are many methodologies available on the market. Selecting the best methodology to suit the requirements of a system is not an easy job. A good methodology results in well structured, reliable, and maintainable software engineering.
The software development process must be fully understood before computer-aided software engineering technology can be truly appreciated.

IEEB standards - The IEEE Computer society, through the software engineering sub committee, has been actively developing software engineering standards. The most pertinent one from a life cycle model perspective is the Standard for Software Life Cycle Processes. The standard defines life cycle management processes, activities within each process, and the interrelationship of processes and activities.
The life cycle project model is based on four Life cycle processes: Project management, predevelopment, development, and post development. These are called processes instead of phases to facilitate mapping the individual processes and activities into specific life cycle models on a project-specific basis.

- Project management processes; These processes prepare the project for implementation and help to management throughout the life cycle. Included are activities such as defining standards, methodologies, and tools/tool sets; allocating resources; establishing the project environment and developing risk analyses and contingency plans.
- Predevelopment processes; These processes are normally performance before the seller activity begins, namely, before contract award. Predevelopment activities include identifying needs, formulating potential solutions, conducting feasibility studies, planning system transition, analyzing and allocating requirements, and developing system architecture.
- Development processes; These processes include defining and developing software requirements, defining software interface requirements, performing detailed design, and implementing the detailed design. These processes most closely resemble the waterfall software development model.
- Post development processes; these processes include installing, operating, supporting, maintaining, and retiring the software product.

Included in the methodology are the so-called integral processes such as software configuration management. These processes augment software development with management-oriented support activities such as reviews and audits. The integral processes defined in the IEEE standard are: verification and validation, software configuration management, documentation development, and training.
Standards (STD-2167A) The developed standards to meet the requirements of costly, complex system's quality, performance, and reliability. These systems are long lived and sufficiently flexible to undergo continuous changes, and they need continuous maintenance for reliability and quality. Standards are developed to increase the compatibility and reliability of computer systems that are mission critical to our nation's defence. These systems use automatic data processing equipment or services. Their function and operation -involve intelligence, command and control of forces, and others.
Each phase of DOD-STD-2176A's Software and Hardware Development Life Cycle concludes with the successful completion of a requirements, design, or test review. At each review point, documentation is produced according to a specified format.
These activities can overlap and can be applied iteratively or recursively:

- System requirements analysis/design
- Software requirements analysis
- Preliminary design
- Detailed design
- Coding and computer software unit (CSU) testing
- Computer software component (CSC) integration and testing
- Computer software configuration item (CSCI) testing
- System integration

-STD-2167A; Although the standard provides standardization for many aspects of software engineering, applying the standards within the framework of the acquisition process might have certain risks. The following list contains some suggestions for working around or solutions to minimize risks.

Data Item Dascreption; The software associated with STD-2167A fall four categories: management, software engineering. test, and operational and support, as write in following paragraph:

Management	Software Development Plan (SDP)
Software Engineering	System/Segment Specification (SSS) System Design Document (SDD) Version Description Document (VDD) Software Requirements Specification (SRS) Interface Requirements Specification (IRS) Interface Design Document (IDD) Software Product Specifications (SPS)
Test	Software Test Plan (STP) Software Test Description (STD) Software Test Report (STR) Operational and Support Computer Systems Operator's Manual (CSOM) Software User's Manual (SUM) Software Programmer's Manual (SPM) Firmware Support Manual (FSM) Computer Resources Integrated Support Document (CRISU)

The system/segment specification (SSS) specifies the requirements for a system or a segment of a system. After government approval and authentication, the SSS becomes the functional baseline for the system or segment. The SSS provides a general overview of the system or segment that can be used by training personnel, support personnel, or users of the system.

The system/segment design document (SSD) describes the design of a system/segment and its operational and support environments. It describes the organization of a system or segment composed, and manual operations. The SDD contains the highest level design information for the system/segment. This information is produced by the system/segment contractor. It is used to document the system or segment design for review at the system design review (SDR). The SSDD requires three types of information:

- Information that is appropriately defined by a contractor, and the definition that is consistent with the schedule (system design).
- Information that must originate with the government but is consistent with the schedule (operational concepts).
- Information that is not appropriate for the SDD.

The SDD describes the complete design of a CSCI that is composed of CSCs and CSUs. This document evolves throughout the life cycle and is to be submitted at:

- PDR containing the preliminary design
- CDR with the detailed design
- FCA/PCA when incorporated into SPS.

The SDD also provides requirements traceability for tracing the requirements to the design and the design to the requirements.
The version description document (VDD) identifies and describes a version of a CSCI.
The software test plan (STP) describes the formal qualifications test plans for one or more CSCI. The STP identifies the software test environment resources required for formal qualification testing (FQT). The standard requires through planning of the CSCI test including documentation, installation, and testing of the test environment.
The software test description (STD) contains the test cases and procedures needed to perform formal qualification testing of a CSCI identified in the plan. The STD is to be prepared and delivered inclemently with the test cases being submitted at CDR and test procedures at TRR.
The software test report (STR) is a record of the formal qualification testing performed on a CSCI. This DID require the reporting of the FQT test results for each test case defined in the STD.

The computer system operator's manual (CSOM) provides detailed procedures for initiating, operating, monitoring, and shutting down a computer system. The CSOM is also used for identifying a malfunctioning component in a computer system. This DID will not apply if the required information is provided in a commercially available document. The CSOM describes the general flow and operation of the computer system. This DID also contain all the information necessary for the day-to-day operation of the system. The CSOM references other commercially available documents where applicable and is organized to facilitate easy access of operational information.

The software user's manual (SUM) provides user personnel with instructions to execute one or more related.CSCI There might be some duplication of material if CSOM and SUM are delivered.

The software programmer's manual (SPM) provides information needed by a programmer to understand the instruction set architecture of the specified host and target computers. The SPM provides information that can be used to interpret, check out, troubleshoot, or modify existing software.

The firmware support manual (FSM) provides the information necessary to load software or data into firmware components of a system. The FSM applies to read-only memory (ROM), programmable ROM (PROM), erasable PROM (EPROM), and other firmware devices.

The computer resources integrated support document (CRISD) provides the information for planning for life-cycle support of deliverable software. The CRISD documents the contractor's plans for transitioning support of deliverable software to the support agency.

The software requirements specification (SRS) specifies the engineering and qualification requirements for a CSCI. The SRS identifies and describes the CSCI external interfaces.

The interface requirement specification (IRS) specifies the requirements for the one or more interfaces among CSCI's and other configuration items. These interfaces will be documented in the SRS.

The interface design document (IDD) specifies the detailed design for one or more interfaces among CSIC's and configuration items.

The software product specification (SPS) consists of the SDD and source code listings, as well as specification of the compiler assembler and measured resource utilization for a CSCI. The SPS become the product baseline when authenticated. The DID does not call for inclusion of the object listing.

The software development plan (SDP) contains information on software development management, software engineering, FQT, software product evaluation, and software configuration management. The DID describes a contractor's plans for conducting software development. Each item of no development software must be described in the SDP along with the rationale for its use.

The 2167A standard *is* supported by a number of other closely related standards:

1- System Software Quality Program (STD2168) i the purpose of this standard is to ensure the quality of deliverable and its documentation, the processes used to produced deliverable software, and no deliverable software. It sets requirements for the contractor quality assurance program and provides for a standard planning document (software Quality Program Plan), conduct of software quality evaluations, and maintenance of software quality records.

2- Configuration management (STD-483); these standard details requirements for the seller configuration management program.

3- Specifications practices (STD-490A); This standard describes the specifications used in the development process and procedures for their preparation and control.

4- Reviews and audits (STD-1521B) this standard prescribes practices for the conduct of formal reviews and audits conducted to see development projects.

One alternative management and engineering development standard. used within software development is STD-1703. Like STD-2167A, it defies the process and its activities, reviews, and products. Unlike STD-2167A, it contains development guidance, including for example, Ada design guidelines. In addition, STD-1703 employs a more flexible tailoring process that allows the buyer and seller to add, delete, and combine requirements and deli verbalise (documentation) as appropriate for the project.

RASA Standards - traditionally the U.S. National Aeronautics and Space Administration (NASA) development centres have operated autonomously, allowing each centre and project to use its own versions of development and documentation standards. There have been recent efforts, however, to provide a NASA-wide standard for software management and development. This work has its origin in the NASA Software Management and Assurance Program, commonly called SMAP.

The NASA SMAP software standards (Release 4.3) are defined in a five-volume set. The parent document, Information System Lifecycle and Documentation Standard, provides terminology, life cycle definitions, and guidelines for adoption to various life cycles. Four Documentation Standards and Data Item Descriptions volumes describe the format and contents for each of four fundamental types and guidelines for tailoring the documentation set. These four volumes together establish a complete framework for all relevant software documentation: specifications, plans, assurance, and reports.

1- Management Plan Documentation Standard and volume; this volume contains the planning information for the project. The outline of a project's planning document list as:

- Introduction
- Application documents
- Resources, budgets, schedules
- Acquisition plan
- Independent verification & validation
- Certification
- Development plan (risk management, engineering and integration, configuration management, assurance, training, delivery and operational transition)
- Sustaining engineering and operations
- Evolutionary acquisition
- Abbreviations and acronyms
- Glossary
- Notes
- Appendices

The software provided in this volume contain structure and description for each section of the planning document.

2- Product Specification Documentation Standard and volume; this volume contains all the engineering and technical information. Thus a product specification would consist of the following topical areas:

- Concept
- Requirements
- Design
- Version description
- User's guide
- Maintained manual.

3- Assurance Specification Documentation Standards and volume; this volume contains all technical assurance information for both the acquirer (buyer) and (provider) (seller). The taxonomy provides for quality assurance, safety and security, testing, verification and validation, and certification.

4- Management Control and Status Documentation Standard and volume; This volume contains all the reports, change papers, and the like that are specified in the management plan.
One major difference between the NASA standards, in general, is that standards are mandated, whereas NASA standards are available for any projects wishing to use them. Selection and enforcement of the SMAP standards is a project responsibility.

8.19.14- PISE/PEHM (Technology Impact Analysis) - Technology impact analysis employees a hierarchical list of technological changes and relates these to polit-management opportunities and competitive threats.

- Defining the requirements for new systems architectures;
- Select and evaluate the technology for development and implementation of the new computer system;
- Plan, manage, and control its development and production; and
- Determine the distribution and the control technology for its maintained and support. Also included in this section are information, image, and speech processing; and decision systems technology.

Information systems encompass the integration and engineering required to:

Engineering Political Enterprise Technology User Interface - Within the scope of he technology component of the CIPSE are: Technology platforms, operations information, and quality.

The technology platform system provides a set of models for examining new technologies and their potential impacts on the CIPSE network. This tool set offers the desirable architectures and technologies for selection, and optimizes their deployment and implementation. The telecommunication service sector has just transformed itself from an analogue to a digital environment. New technologies need to be carefully examined to determine their proper deployment within the infrastructure. Specific applications include architectural validation, analysis tools for technology selection, and performance analysis tools.