

INFORMATION SYSTEMS DESIGN FOR DEVELOPMENT PROJECTS IN CENTRAL AMERICA¹

*J. Ramiro Montealegre*²

**Institute of Nutrition of Central America and Panama
(INCAP),
Guatemala, Guatemala, C. A.**

SUMMARY

Present development projects in Central America, particularly food aid and food and nutrition education programs, are operating under severe management constraints which limit their potential to produce positive and measurable impacts on target communities and families. Thus, operational analysis and information systems are basic managerial tools to improve the efficiency of the projects.

This paper presents an information system development methodology which has been used by the Institute of Nutrition of Central America and Panama (INCAP) to assist different development projects of Central America. The methodology is geared on the one hand, towards systematizing the definition of the demand and of applications and, on the other, of the types of response required or deemed convenient, taking into account the technologies that are appropriate to the users' various capacities.

The complete development cycle is based on good understanding of the social, organizational and human aspects of the project. Through the whole process, not only is the information system being developed and documented, but personnel is being trained and assisted in solving their problems, contributing thereby, to true transfer of technology.

No new set of techniques of systems development is presented. What has been done is to carefully select elements of current technical practices, modifying tools and techniques to generate a unique methodology through which the information systems' requirements, specifications, and details may be expressed.

Manuscrito original recibido: 6-1-89.

1 This research is being sponsored by the Institute of Nutrition of Central America and Panama (INCAP), and The United States Agency for International Development (USAID/ROCAP) through the Project "Technical Support for Food Assistance Programs" (No. 596-0116).

2 Information System Specialist of the Project of Technical Support for Food Aid Programs (PROPAG), INCAP.

INTRODUCTION

This paper not merely presents a design technique, but a way of thinking about information systems which should be understood by users and information specialists in order to prevent development of software which has little use and application. The methodology presented here is based on the need to concentrate not only on the software system that must be built, but also on the human, social and organizational systems of which the software system will be part of.

No new set of techniques of systems development is presented. What has been done is to carefully select elements of current technical practices, modifying tools and techniques to generate a unique methodology. This methodology applies to high levels of system conceptualization all the way down to the lowest level of program design. Its main importance, however, is that the specifications given for complex systems are engineered with a precise progression of successive refinements to meet specific productivity requirements. But productivity is not measured in "lines of code per week" or some other sterile statistic, but as the ability to get software working, as quickly as possible, into the hands of the users, taking into consideration the particular management style of the organization.

1. *Background*

The present economic and social crisis affecting large segments of the populations of the Central American countries has generated the need for development programs in these nations. These social and economic development programs are geared towards upgrading educational, health, housing, industrialization and agricultural status, as well as towards achieving greater productivity in all sectors.

These development programs can be seen as tools to subsidize the incomes of beneficiary families and, in certain countries, they have been an important governmental instrument for income redistribution. On the other hand, many communities and families are so economically deprived that they need assistance programs to subsist and avoid further rapid deterioration of their health and nutritional status while governments identify, fund and implement efficient and effective policies and programs aimed at reducing poverty.

Present development programs, particularly food aid, and food and nutrition education programs are operating under severe management constraints which limit their potential to produce, at reasonable costs, positive and measurable impacts on target communities and families. Thus, operational analysis which identifies and corrects potential management problems, together with information systems to support decisions which target resources to more needy populations, monitor the delivery of services and assess program impacts, are basic processes to improve the efficiency of development programs.

However, given the dynamic environment and its susceptibility to crisis, the existing flexibility in defining public institutions' functions, the assignment of scarce resources and the changes in administrative personnel cause decision-making and evaluation criteria to rapidly lose validity. Therefore,

the usually greater effort involved in setting up information systems in traditional ways in these types of institutions, means that such systems soon become inefficient instruments or problems, rather than tools for better management (1,2).

The Institute of Nutrition of Central America and Panama (INCAP) in response to the needs expressed by the Governments of its Member Countries, mainly in relation to feeding programs, has initiated a process of generating an information systems development methodology (2). This methodology is geared towards systematizing the definition of the demand and of applications, and, secondly, towards systematizing the definition of the types of response required or that are more convenient, taking into account the technologies that are appropriate for the users' various capacities.

2. Methodology Description

Successful software systems are not developed in a vacuum. A major reason for complications with systems is inadequate communication between users and developers, or between requirement definers and programmers (3). Implementers and users should work and understand the requirements together. Each group can trigger creativity in the other so that the combination produces better results than could be produced by either group alone.

Taking into consideration the rapid changes that the demand (information needs), the supply structures (organizational and flow), and the information technology produces, participatory research is used with the information specialists and diverse users to implement an information strategy. The former have knowledge and abilities in information systems development, and the latter are interested in solving problems of the system in which they are directly involved. The concept of user-specialist tries to shorten the communications gap (potentially dangerous when considering problem analysis within a dynamic environment), and contributes to the systematic identification of institutional or sectoral problems while identifying users conduct, knowledge, capacity and idiosyncrasies. Both the information supply and demand are treated with flexibility, influencing each other.

The methodology involves the successful completion of three stages:

Stage 1: Problem identification

Stage 2: Analysis and design of the information system

Stage 3: Detailed design and implementation plan

Different languages and techniques are traditionally used for different aspects of the developmental phases of the information system life cycle. It has been shown that it is difficult to make each new phase correspond to the previous one (4). This is partly because each is defined in different languages. In this methodology, the techniques used to present requirements, specification and design are complementaries, allowing changes made during the last phases of development to be easily incorporated to the product of the initial ones, while preserving integrity and facilitating verification through the complete development process.

3. *Stages of the Methodology*

3.1 *Problem identification*

At the beginning of this stage, the Basic Technical Group (BTG) is organized. The BTG is composed of five or six key officers of the institution being studied. This Group would represent both the information users and personnel involved in the various stages of data processing.

The members of the BTG are likely to start off speaking different languages. Each will be using terminology based on his own view of the world. A good specification language should build a bridge of understanding among BTG members. A common language should be used to allow the specific members of the BTG to choose the level of detail they want to use in specifying the system.

During the first contacts with the situation to be analyzed, the BTG should define the following:

- The "objects" (institutions, sectors, programs, projects), "actors" (groups and individuals involved), and problem and solution categories that will be taken into consideration or excluded from the analysis.
- The analysis objectives and the immediate impact on the analyzed situation, actors and clients.

Supposedly, these definitions will have to suffer modifications, sometimes considerable, during the construction of the production and efficiency model by the analyst and participants.

3.2 *Analysis and design of the information systems*

For this stage, 25 to 30 institutional officers are selected to constitute the Basic Users Group (BUG). The members of this group should be selected according to their position or capacity to contribute to the process of identifying the information "problem set" and its possible solutions.

This stage is divided into three work phases: 1) the situation analysis, 2) verification of the analysis and identification of the solution capacity, and 3) general design.

3.2.1 *Analysis of the situation*

In the first phase it is important to make available to the members of the BTG and the BUG the techniques that would permit them to easily communicate their knowledge of the system. By means of guided working meetings, their information needs are identified (demand) and the existing capacity to meet these needs is evaluated.

These techniques are based on two basic concepts: graphic representation and a Top-Down approach. The use of graphic material during the analytical phase is effective while working with the users, since they can react to the graphic representation of the proposed system and pinpoint specific areas where understanding is faulty. From the standpoint of time, this is likely to be much more effective than simply asking questions. Furthermore, since the user is reacting to something tangible, it is easier for him to feel that he is an integral part of the development of the new system.

A top-down approach has proven (5) to be an effective way to quickly get underway and to manage the development once the system starts coming together. It helps reduce the risk of false beginnings and provides the ability to judge the effect of developmental problems of one area on the rest of the system. Nevertheless, pure top-down techniques are impractical, because a change made in a lower level module may affect the other modules, and it is desirable that this ripple effect be immediately traceable. The techniques employed must allow requirements addition from the top or bottom.

The specification process presented in this phase produces a succession of formal representations of the system which can steadily be broken down into more detail. Different people are likely to be involved at different levels of detail. The techniques employed are data flow, hierarchical and Warnier-Orr diagrams.

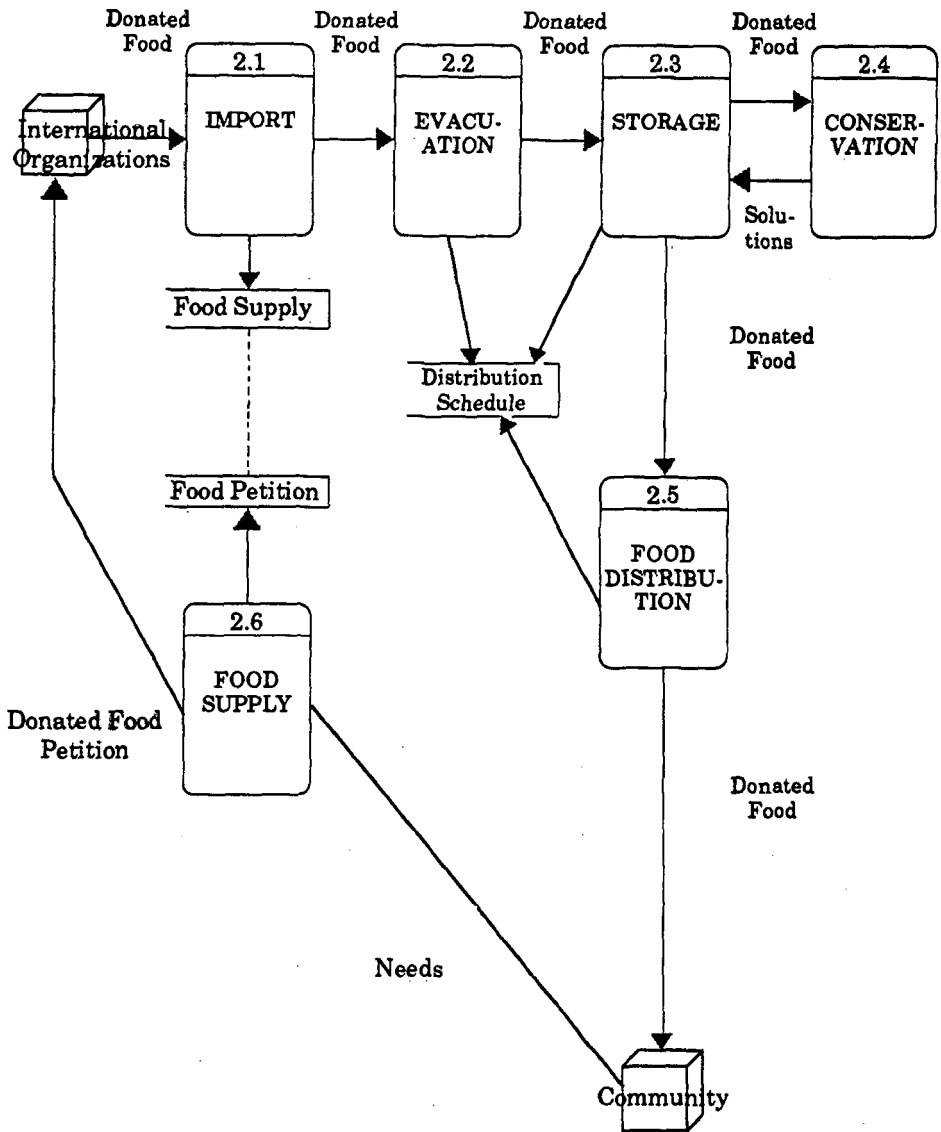
A data flow diagram is used as the first step in a structured design form. At a high level, it is used to show institutional events and the transactions resulting from these events. It is primarily a tool used to draw the basic procedural components and data that pass among them (6). The advantage of applying this technique as a first option is to permit the user to rapidly become familiar with the technique by indicating which procedures and routines they accomplish, which is what the users know best.

Figure 1 shows the highest level data flow chart of the logistic subsystem for donated food distribution in a Central American public institution. To transfer donated food by international organizations to the community, the users easily identified the data flow through six basic processes. The Import process (Process 2.1) includes all the activities required to accept the food in the country. Through the Evacuation process (Process 2.2) the food is transported from the point of disembarkation to the warehouses. The Storage process (Process 2.3) includes arrival, control and outflow of food in the warehouses. Thru the Conservation process (Process 2.4), problems in food conditions are detected and solutions are sought. Food arrives at the community through the Distribution process (Process 2.5). Finally, the communities' requests generate the Food Supply process (Process 2.6) that culminates in the formal requests made by the diverse international organizations. Further breakdown of this graph into more detailed data flow diagrams enhances understanding of the operation and structure of the model subsystem.

When a data flow diagram is developed during structured analysis, a process specification and a data dictionary are also developed to give additional system information.

Based on the data flow diagrams, a tree structure is generated which relates to functions and not to the data that those functions use. The chart needs to be drawn vertically, rather than horizontally (7). This technique is used at this point for two main purposes: 1) to review if the model generated by the data flow diagram is complete when the same system is viewed from another angle; and 2) to begin the abstraction process required to identify the basic elements of the model system.

Figure 2 presents the six processes shown in Figure 1, with the breakdown of the subprocesses and the main activities they are formed by. For example, the Import Process consists of the following subprocesses: The "RISI" Document Reception (subprocess 2.1.1), the "shipping Instruction (SI)", Document Reception (Subprocess 2.1.2), the "Bill of Lading (BL)"



SYMBOLOLOGY:





-  External Entity
-  Data Storage
-  Process
-  Data Flow

FIGURE 1

Data flow diagram of the logistic subsystem for donated food distribution of a Central America Institution.

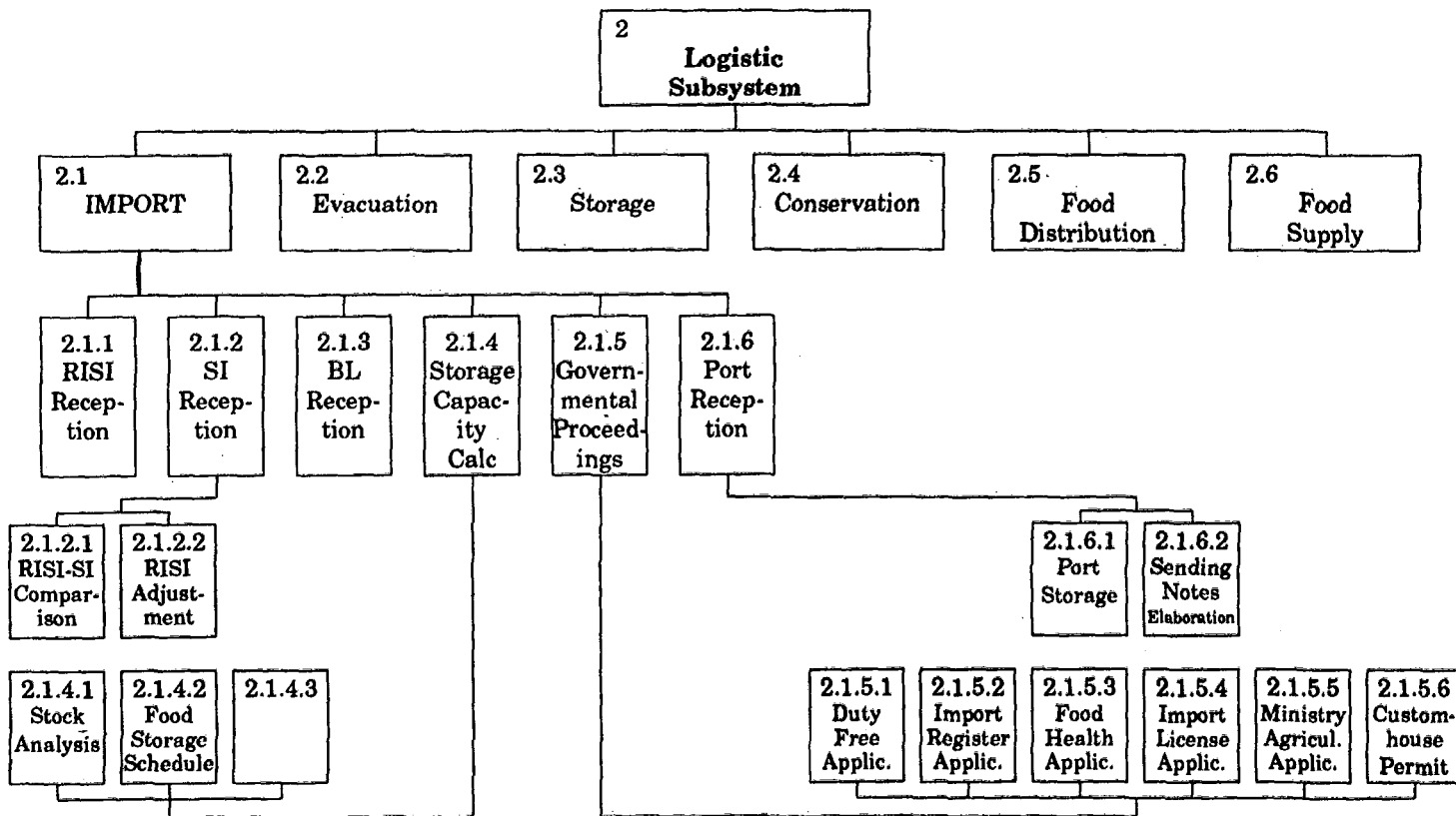


FIGURE 2

Hierarchical diagram of the import process of the logistic subsystem for donated food distribution

Document Reception (Subprocess 2.1.3), Storage Availability Calculation (Subprocess 2.1.4), procedures of governmental permissions (Subprocess 2.1.5) and, finally, food reception at the port (Subprocess 2.1.6).

The identification of the activities required for the objectives achievement should be done separately from the current organization chart (4). The organization may change, but still has to carry out the same functions and processes. Some institutions reorganize dramatically, like the Government of Bolivia. The identification of functions and processes should represent fundamental concern for how the institution operates apart from its current organization chart (which is often misleading). There needs to be basic questioning about whether the functions and processes perceived are sound.

In most public institutions of Latin American activities have never been charted. When they are listed and related to the data they use, it is usually clear that much duplication exists. Each area of the institutions tends to expand its activities without knowledge of similar activities taking place in other areas. Each department tends to create its own paperwork.

When the user and specialist groups have matured beyond this stage, then a functional breakdown should progress to show the data types that are input and output for each function. In this phase, Warnier-Orr diagrams are useful.

The Warnier-Orr technique has been selected for use during the third step of this phase because it is easy to learn and use, being composed of only four basic diagramming techniques (8). Furthermore, it can show the basic program control constructs of sequence, selection and repetition. Providing good data structure documentation is another major benefit of Warnier-Orr diagrams. This technique allows the user to easily understand which data are required for the execution of the model processes.

Figure 3 depicts the Warnier-Orr diagram of the Import process of the Food distribution subsystem exemplified in the above figures. For example, the "shipping Instructions (SI) Reception (Subprocess 2.1.2), is formed by the comparison of the "RISI" and "SI" Documents (Activity 2.1.2.1) and the readjustment of the "RISI" Document (Activity 2.1.2.2). To be able to carry out these activities, the following data of the "RISI" Document are required: RISI Number, Project Number, RISI Date, Food Type, Food Quantity and Arrival Date. Meanwhile, from the "SI" Document, the following data are required: SI Number, RISI Number, SI Date, Ship Name, Food Type Shipped, Food Quantity Shipped, Arrival Date, Arrival Port and Package Type.

3.2.2 *Verification of the analysis and system capacity assessment*

During the second phase, the specification charts are handed over to a different team. This team can give feedback on refinements to the chart, which are then checked by the members of the BUG.

Also, a presentation of the first phase should be given to the authorities of the institution being studied to evaluate the results, incorporate the authorities' contributions and give priority to areas needing solutions.

3.2.3 *General design*

Considering that human inventiveness and creativity are the main

	2.1.5.2 Import Register Application	2.1.5.2.1 Form "10"	Date and Place Importer Data: - Name and Address - NIT Exporter Data: - Name and Address - Origin Country - Name and Address representative Food Data: from the BL
	2.1.5.3. Food Health Application	2.1.5.3.1 Food Health Form	Date and Place Food Type Food Quantity
	2.1.5.4 Import License Application	2.1.5.4.1 Import License	Licence Number Importer Name Original Country Food Type Food Quantity Date and Place Expiration Date
	2.1.5.5 Ministry of Agriculture Application	2.1.5.5.1. Import Permit	Permit Number Importer Name and Address Food Type Food Quantity Origin Country Package Type Arrival Port
	2.1.5.6. Customhouse Permit Application	2.1.5.6.1. Customhouse Permit	Permit Number Permit Date Customhouse Name Importer Name and Address NIT Duty Free Permit Number Form "10" Number BL Number Health Form Number Import Permit Number Arrival Date Food Value
2.1.6. Port Reception	2.1.6.1. Port Storage	2.1.6.1.1. Superintendencia Report	Report number Date and Place Ship Name Arrival Date Arrival Port BL Number
	2.1.6.2. Elaboration of Sending Notes	2.1.6.2.1. Food Storage Schedule 2.1.6.2.2. Superintendencia Report	

aspects of the design, once again, techniques used should provide a way to simplify the drawing of concepts.

The same graphic breakdowns generated during the first phase, and validated in the second phase, can be done in greater detail until they represent the program modules that need to be computerized.

Based on the Warnier-Orr diagrams, the system is designed as follows:

1. Define the process outputs, representing each program's output as a hierarchical data structure.
2. Define the Logical Data Base, defining all the data elements needed to produce the program output.
3. Perform Event Analysis, defining all the events that can affect (change) the data elements in the logical data base.
4. Develop the Physical Data Base, defining the physical files for the input data.
5. Design the Logical Process, designing the program processing logic that is needed to produce the desired output from the input.
6. Design the Physical Process, adding the control logic and file handling procedure to complete the program design.

Figure 4 demonstrates the software system design generated for the logistic subsystem for the donated food distribution presented above. Note that based on the study of the processes, appropriate points could be identified to obtain the data for the different data banks designed.

This is also very practical for the documentation of the system, because the entire structure, top to bottom, is reflected. Subsequent system changes and maintenance could be done using this documentation to have a clear and detailed description of the design.

3.3. Detailed design and implementation plan

As mentioned by John Boddie: "If the project met its milestones and the system has come together smoothly, there should only be a few thousand little details to attend to in order to finish the job..." (9). The results of the previous stage should be taken into consideration by the information specialists, reviewed and optimized according to the particular environment in which the system will be used.

Until now, this stage is very time-consuming because the information specialist needs to go through every step of the design in a pencil-and-paper approach. The power of this methodology for the future lies on Computer Aided Software Engineering (CASE) products, which could allow the development team to define the system's specifications at a work station screen and let the machine generate the executable code needed. The machine could apply all sorts of cross-checks validation and automate many of the tasks which are now tedious and time-consuming.

The members of the BTG and BUG should be held primarily responsible. However, it is not necessary for the entire development group to be present at the installation. In fact, by the time installation begins, some members of these groups will normally be busy doing something else. The people necessary for the installation are those who during the development process have shown abilities in isolating problems and fixing them fast.

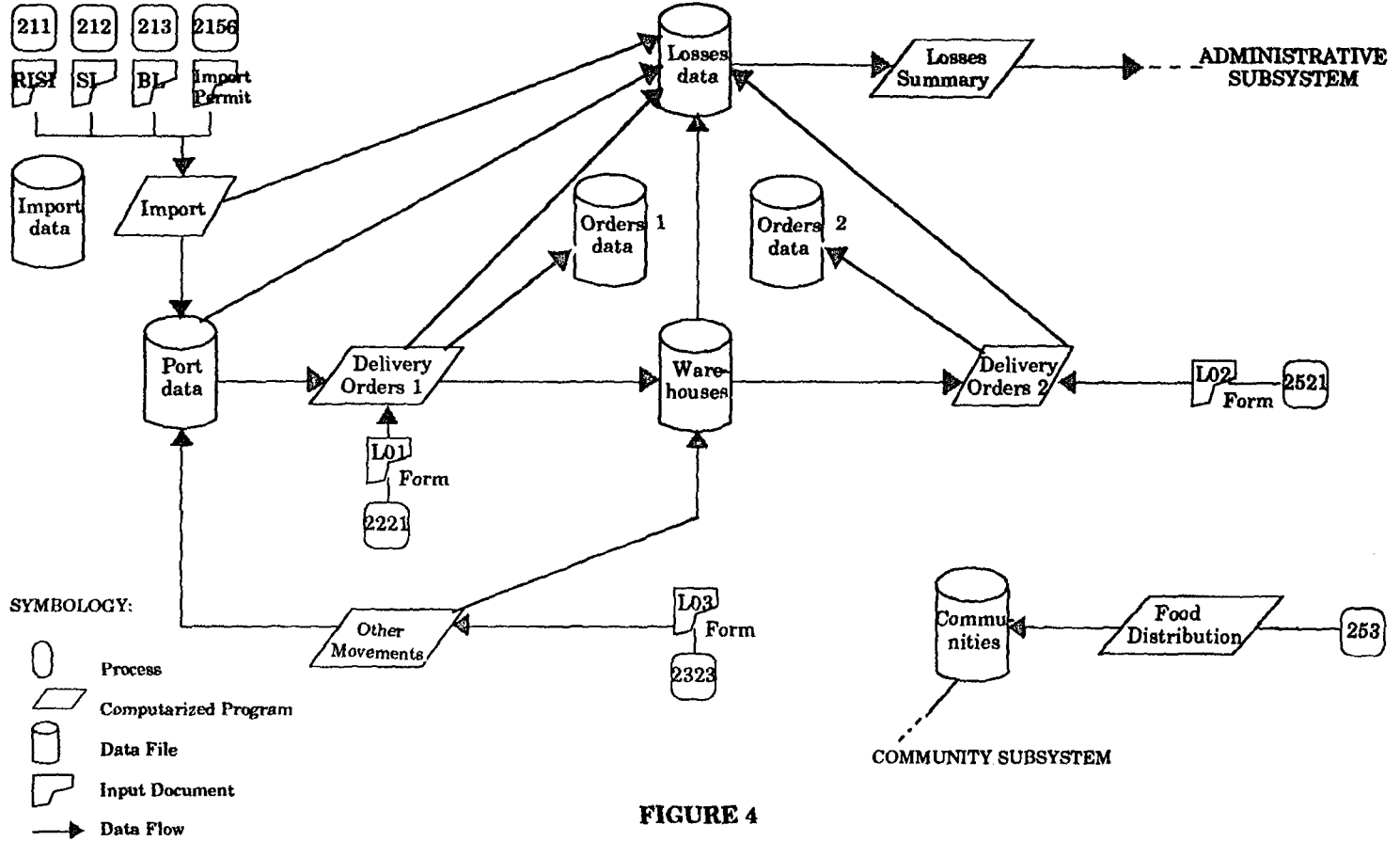


FIGURE 4

Physical design of the logistic subsystem for donated food distribution

4. *Documentation, Training and Schedule*

4.1 *Documentation*

A system needs two kinds of documentation: development documentation, and user documentation. The development documentation, as mentioned before, consists mostly of the drawings used during the previous stages. It has the advantage that it is produced as the system is being developed.

User documentation is different. Preparing good documentation is not a trivial job. It requires someone who understands the users, understands the system, understands the way they will work best together, and is able to organize all this understanding so it is useful. The job is beyond the ability of most programmers and project leaders and usually can be very time-consuming. With the methodology presented here, however, a good technical writer, involved in the process from the second stage on, could write the user's documentation.

4.2 *Training*

It is well known that training is best handled by specialists. Nevertheless, the ability to teach a user does not come naturally, and a poor presentation of the system can spoil an installation. It is a big mistake to assume that because the system is technically correct the users will be willing to change their daily procedures and routines. Through this methodology, the specialists become familiar with the users and can understand the way the system will be used, so a better training plan can be generated.

4.3 *Schedule*

Figure 5 shows a typical schedule of a "traditional" software development cycle, while Figure 6 depicts a typical schedule for the development of a software system using the methodology presented here. These Figures are similar in the way they address the set of tasks that must be performed by a software system. They are different in the way they reflect the order in which the tasks are carried out. With this methodology there will almost always be a high degree of overlap among design and implementation activity. In the "traditional" system development, a good information specialist will take the time to do more design work up front, before committing resources to the implementation.

The jump from specification to design is possible because of a good understanding of the system provided by the user himself. Identifying the specific procedures needed is almost mechanical.

CONCLUSIONS

The information system design methodology presented here for development projects, starts with the ordering of the information supply and demand of the project.

The complete development cycle is based on good understanding of the

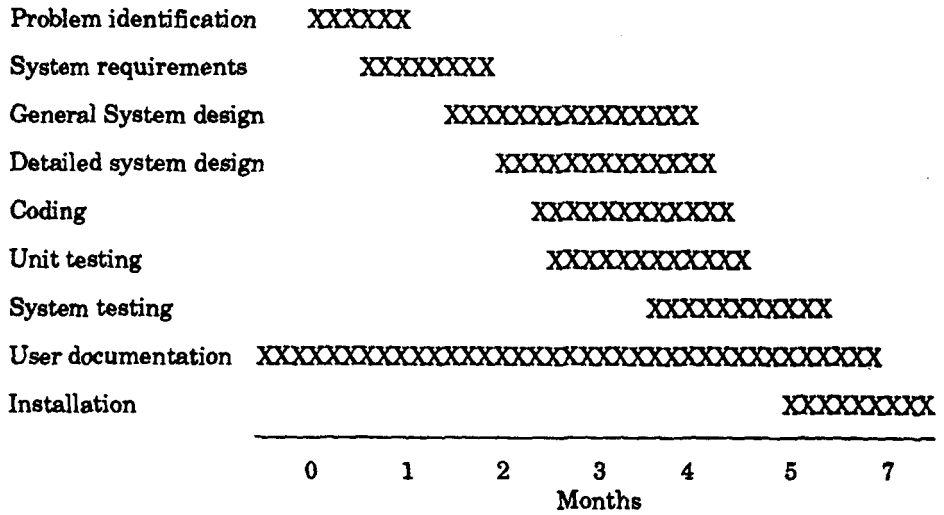


FIGURE 5

Typical "traditional" software development schedule

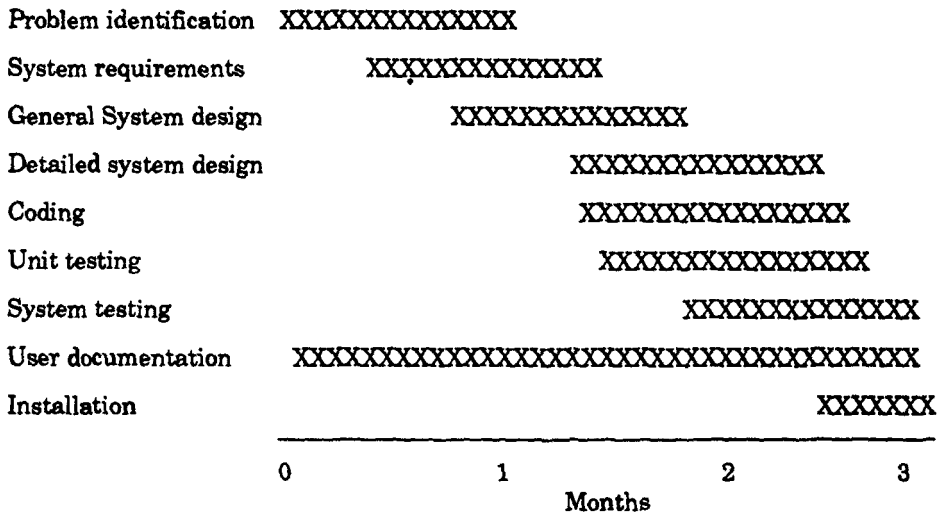


FIGURE 6

Typical software development schedule using the methodology presented here

social, organizational and human aspects of the project. Through the whole process, not only is the information system being developed, but the personnel at the same time is being trained and aided to solve their problems, contributing to true technology transfer.

In order to have a traceable evolution of the system's development, a set of complementary techniques were chosen through which requirements, specifications, and details could be expressed. The requirement statements are broken down into greater detail and become the specifications. The specification statements are broken down into greater detail until sufficient detail is reached so that code can be generated. When changes are made at a lower level, they are reflected upward.

Additionally, the described methodology leads the participants to:

- Think, and know more about the nature, objectives, strategies, procedures and organizational structure of their institution.
- Determine which process chains, activities, and functions are critical to the achievement of the goals on which they should concentrate, and to which they should contribute to assure that these main blocks receive special care on their behalf.
- Identify the system's capacity assessment as they seek solutions to operational problems.

The set of techniques mentioned here have been used in different development programs of Central America in a pencil-and-paper approach. While many computing tools are available in the market, given the need for integration and the complex specifications of a public institution in Latin America, further work is required to computerize and automate the full methodology and make it easily adaptable to changes.

RESUMEN

DISEÑO DE SISTEMAS DE INFORMACION PARA EL DESARROLLO DE PROYECTOS EN CENTROAMERICA

Los proyectos de desarrollo que existen en la actualidad en Centroamérica, particularmente los programas de ayuda alimentaria y de educación alimentaria nutricional, están operando bajo restricciones gerenciales severas que limitan su potencial de alcanzar impactos positivos y medibles en las comunidades y familias a las que van dirigidos. De esta cuenta, el análisis operacional y los sistemas de información representan herramientas gerenciales básicas para mejorar la eficiencia de este tipo de proyectos.

En este documento se presenta una metodología para el desarrollo de sistemas de información que ha sido empleada por el Instituto de Nutrición de Centro América y Panamá (INCAP) para colaborar con diversos proyectos de desarrollo en los Países Miembros. La metodología está dirigida hacia la sistematización de las definiciones, en primer lugar, de la demanda y de las aplicaciones existentes de información, y en segundo lugar, de los tipos de respuestas requeridas o convenientes, tomando en consideración las tecnologías apropiadas y la capacidad de los diversos usuarios.

El ciclo completo de desarrollo se basa en la buena comprensión de los aspectos sociales, organizacionales y humanos del proyecto. A través de todo el proceso, no sólo se desarrolla y documenta el sistema de información, sino que al mismo tiempo se

capacita al personal y se le ayuda a resolver sus propios problemas, contribuyendo así a la verdadera transferencia de tecnología.

Ninguna técnica nueva de desarrollo de sistemas de información forma parte de la metodología. Lo que se ha hecho es seleccionar cuidadosamente elementos de técnicas y prácticas existentes, modificando herramientas apropiadas para generar una metodología única, en la que los requerimientos, especificaciones y detalles del sistema de información puedan ser expresados.

BIBLIOGRAFIA

1. Picado, X. & W. Crowther. El control en el sector público: Administración es más que control, control es más que información e información es más que datos. *Rev. Centroamericana de Administración Pública*, Número 6, enero- junio, 1984.
2. Montealegre, J. R. Sistemas de Información en los Programas de Alimentación a Grupos en centro América y Panamá: Enfoques. Guatemala, Instituto de Nutrición de Centro América y Panamá, 1987.
3. Ward, P. *System Development Without Pain*. New York, N. Y., Yourdon Press, 1984.
4. Martin, J. & C. McClure. *Structured Techniques for Computing*. Englewood Cliffs, New Jersey, Prentice-Hall, 1985.
5. Martin, J. *System Design From Probably Correct Constructs*. Englewood Cliffs, New Jersey, Prentice Hall, 1985.
6. Gane, C & F. Sarsson. *Structured System Analysis: Tools and Techniques*. 1 st. Inc., New york, N. Y., 1977.
7. Yourdon, E. & L. Constantine. *Structured Design*. New York, N. Y., Yourdon Press, 1978.
8. Warnier, J. D. *Logical Construction of Systems*. New York, N. Y., Van Nostrand Reinhold, 1981.
9. Boddie, J. *Crunch Mode: Building Effective Systems on a Tight Schedule*. Englewood Cliffs, New Jersey, Yourdon Press Computing Series, Prentice-Hall, Inc., 1987.