Talib Damij
Faculty of Economics
University of Ljubljana
Kardeljeva ploscad 17
1000 Ljubljana
Slovenia

Library of Congress Cataloging-in-Publication Data
Damij, Talib.
Tabular application development for information systems: an object-oriented methodology/Talib Damij.
p. cm.
Includes bibliographical references and index.

ISBN 978-1-4612-6523-8 ISBN 978-1-4613-0095-3 (eBook)
DOI 10.1007/978-1-4613-0095-3
1. Object-oriented methods (Computer science) I. Title.
QA76.9.O35 D36 2000
005.1'17—dc21

00-059585

Printed on acid-free paper.

Originally published by Springer-Verlag New York, Inc in 2001
Softcover reprint of the hardcover 1st edition 2001

Production managed by Timothy Taylor; manufacturing supervised by Jeffrey Taub.
Composition by Impressions Book and Journal Services, Inc., Madison, WI.

9 8 7 6 5 4 3 2 1

ISBN 978-1-4612-6523-8

Talib Damij
Faculty of Economics
University of Ljubljana
Kardeljeva ploscad 17
1000 Ljubljana
Slovenia

Library of Congress Cataloging-in-Publication Data
Damij, Talib.
Tabular application development for information systems: an object-oriented methodology/Talib Damij.
p. cm.
Includes bibliographical references and index.

ISBN 978-1-4612-6523-8 ISBN 978-1-4613-0095-3 (eBook)
DOI 10.1007/978-1-4613-0095-3
1. Object-oriented methods (Computer science) I. Title.
QA76.9.O35 D36 2000
005.1'17—dc21

00-059585

Printed on acid-free paper.

Originally published by Springer-Verlag New York, Inc in 2001
Softcover reprint of the hardcover 1st edition 2001

Production managed by Timothy Taylor; manufacturing supervised by Jeffrey Taub.
Composition by Impressions Book and Journal Services, Inc., Madison, WI.

9 8 7 6 5 4 3 2 1

ISBN 978-1-4612-6523-8

To my father, who supported me all his life,
to my mother, brothers, and sister for their concern, and
to my family for their patience.

Preface

There are many very interesting information systems development methodologies, including the Object Modeling Technique (Rambough et al.), Object-Oriented Software Engineering (Jacobson et al.), Object-Oriented Analysis and Design (Booch), Object-Oriented Analysis and Design (Martin and Odell), Unified Modeling Language (Booch, Jacobson and Rambough) and other well-known methodologies.

Each of these methodologies is an excellent tool capable of enabling the analyst to develop a well-designed and successfully implemented information system. The only condition for achieving such success is that the analyst has to be adequately familiar with the methodology used.

The purpose of this book is to introduce a new object-oriented methodology called Tabular Application Development (TAD). The reason for developing a new methodology is that TAD represents a new idea, which is simple and very different from the ideas used in the above-listed and other methodologies.

TAD is designed to approach the subjects of Business Process Reengineering and Information Systems Development from a new perspective, which is unique to this methodology.

This book is intended for academic purposes and for researchers and practitioners in the fields of Information Systems Development Methodologies and Business Process Reengineering.

The real world of any organization, for which we intend to develop an information system, is a complex system consisting of many business processes, work processes, activities and tasks. This fact requires that the analyst invest a lot of time, work and knowledge in order to understand the organization's functioning and to keep the whole process of information systems development completely visible to him or her. This is the only way to enable the analyst to create a linkage between different levels of the enterprise and also among the organization's different parts.

Different methodologies use different ways and approaches to transform the real world into information systems models. How to transform reality into such models is essential not only for developing a successful

information system but also for helping users to optimize their work and to make their organization more efficient and successful.

TAD methodology tries to understand the real world using several tables. For this reason it is called Tabular Application Development. The reasons for using tables to model the real world are as follows:

- tables are very useful for representing events and occurrences in real life;
- tables can be easily surveyed, corrected and extended;
- tables are very visible and easily understandable.

Therefore, the analyst can find any information about business processes, work processes, activities, tasks or any event almost at once. This fact gives the analyst the good feeling that everything is proceeding without any misunderstanding or confusion and also that any mistake or problem can be searched, found and corrected quickly.

In addition, tables are easily understandable by the users. Many times we find ourselves in situations where users start to understand the tables and help us develop them without any previous explanation. This is an interesting case, which proves the simplicity of using tables to discover, represent and understand the real world.

The concept of the methodology discussed is broken down as follows:

- first, collect information about the behavior of the real world into different tables;
- second, identify and implement changes by analyzing the contents of the tables; and
- third, use the information gathered in the changed tables to develop the information system of the organization.

Talib Damij
Slovenia

Acknowledgments

I would like to express my special appreciation to Professor Michael Kahn, Washington University, for reviewing this book and for his constructive comments and questions after reading my first manuscript about TAD methodology. He convinced me that TAD methodology deserves to be known widely and encouraged me to write about it.

I would like to express my gratitude to Professor Janez Grad, University of Ljubljana, Slovenia, for reviewing this book and for his interesting comments and suggestions. He encouraged me to develop a case-tool to support TAD methodology.

I owe a lot of thanks to Dr. Anthony Byrne, Jozef Stefan Institut, Slovenia, and Paul Morris for patiently doing all the proofreading.

Talib Damij
Slovenia

Contents

PART I

INTRODUCTION

The first part includes three chapters. Chapter 1, Overview, introduces the organization and content of the book. Chapter 2, Object-Oriented Concepts, discusses object-oriented concepts. Chapter 3, Methodology Preview, gives a brief account of TAD methodology.

CHAPTER 1

Overview

The book is organized in four parts: Introduction, Methodology, Applications and TAD Case-Tool.

The first part consists of three chapters. The first chapter gives a brief introduction to the content of the book. The second chapter describes basic object-oriented concepts such as objects, classes, attributes, associations, operations and inheritance. The third chapter introduces the key concept and the central idea on which TAD methodology is founded.

The second part consists of six chapters and deals with the description of TAD methodology. In addition to the theoretical explanation of the methodology, a case study ("Tourist Organization") is used to show the implementation of the discussed ideas, steps and phases.

Chapter 4 discusses the first phase (problem definition). In this phase, the real world of the problem to be solved is identified and reduced to understandable terms. This is achieved by developing the entity table. This table is developed and completed during interviews with the users. The purpose of developing the entity table is to identify the organization's structural scheme, vital outputs and analyses related to the strategic plan and goals, and the organization's decision support problems.

Chapter 5 describes the second phase of the methodology (systems functioning). This phase analyzes the functioning of the organization. This is achieved by identifying every activity accomplished in the framework of the organization. In order to do this, we organize interviews with the system's users. The aim of these interviews is to develop the activity table.

Each activity could include one or more tasks. To understand the functioning of the system, we need detailed information about tasks in order to identify all the circumstances in which they are accomplished. Identification of tasks and their characteristics is achieved by developing the task table.

After completing the activity and task tables, we continue our work by defining the work processes and business processes of the organization. Defining work processes is achieved by grouping the activities into suitable groups. Each of these groups represents a determined work

T. Damij, *Tabular Application Development for Information Systems*

process. We go on grouping the defined work processes into suitable groups. Each of these groups introduces a certain business process.

Chapter 6 deals with the third phase (business process reengineering). To carry out business process reengineering in an organization, we establish a project team, which consists of the analyst and representatives of the management at the strategic, business and operational levels. The project team's work leads to reengineering the entity, activity and task tables, which is essential in creating a more successful organization.

Chapter 7 introduces the fourth phase (object model). This phase identifies the objects of the system, their attributes and their operations. Information about object classes is obtained by analyzing the task, entity and activity tables. The identified object classes, attributes and associations are then used to develop the object model of the system.

Chapter 8 describes the fifth phase (design). This phase deals with designing the system and preparing it for implementation. This is achieved by creating the operation table, which links the functioning of the system with its static structure, and by developing the application model. Development of the application model is derived from the information collected in the entity and activity tables.

After completing the operation table, we then write a set of algorithms that define in detail the content of the operations of the object model.

Chapter 9 represents the final phase of the methodology (implementation). This phase involves choosing a convenient database management system to implement the object model of the system and translating the written algorithms related to the operations of the object and application model into program codes.

Part III of the book has four chapters. These chapters discuss four applications: a gastroenterological internal clinic; a clinic for small animals; a payment movement; and a student office. All applications are presented in reduced form because the real models of these information systems are much larger and therefore not of a convenient size for presentation in this book.

Part IV has just one chapter, which introduces a case-tool to support the TAD methodology.

CHAPTER 2

Object-Oriented Concepts

Using object-oriented concepts, we try to develop an information system that gives a realistic representation of the real world. The information system developed is created as a reflection of reality and implemented to act, behave and work as the real system does.

An information system developed through object-oriented concepts and object-oriented programming languages enables us to reuse system components. This is a very important feature of object-oriented modeling, as it contributes to the reduction of the programming code.

In this chapter we will discuss the following object-oriented concepts:

- objects,
- classes,
- attributes,
- associations,
- operations,
- inheritance.

These object-oriented concepts are described in many well-known books that deal with object-oriented programming and object-oriented methodologies for information systems development. For this reason, we introduce each of these concepts first by listing some interesting definitions given by well-known authors and then by adding a brief additional explanation.

2.1 Objects

What is an object? The best way to answer this question is by listing a few definitions the reader can find in many well-known books on object-oriented modeling and object-oriented programming:

T. Damij, *Tabular Application Development for Information Systems*

An object is anything, real or abstract, about which we store data and those operations that manipulate the data (Martin, Odell, 1992).

An object is a tangible or intangible thing that can be classified and uniquely identified (Sanders, 1995).

An object is simply something that makes sense in an application context (Rumbaugh et al., 1991).

An object is characterized by a number of operations and a state which remembers the effect of these operations (Jacobson et al., 1996).

In accordance with these definitions, we can say that an object is anything identified in the process of information systems development and recognized by its properties (attributes) and behavior (operations).

For example, Order, Shipment, Invoice and Customer are examples of objects recognized in a sales organization. Patient, Doctor, Disease, Medication and others are objects of a hospital. Student, Professor and Course are objects that could be registered in a student office system.

Identifying the objects in the process of developing an information system for a certain organization enables us to understand the real world in the framework of the considered organization and provides a solid foundation for the system's implementation.

It is important to emphasize that each object has an identity that represents it. The identity (ID) is the object's property that uniquely distinguishes it from other objects. For example, invoice no. 1346, invoice no. 2612 and invoice no. 5270 are objects in the sales organization. These objects are presented by their identity numbers.

2.2 Classes

An object class describes a group of objects with similar properties (attributes), common behavior (operations), common relationships to other objects and common semantics (Rumbaugh et al., 1991).

A class of objects can be constructed, described and qualified by the objects' attributes and by other classes. A class, then, is constructed from components of attributes and other classes. This is referred to as the class aggregation abstraction (Sanders, 1995).

A class represents a template for several objects and describes how these objects are structured internally. Objects of the same class have the same definition both for their operations and for their information structures (Jacobson et al., 1992).

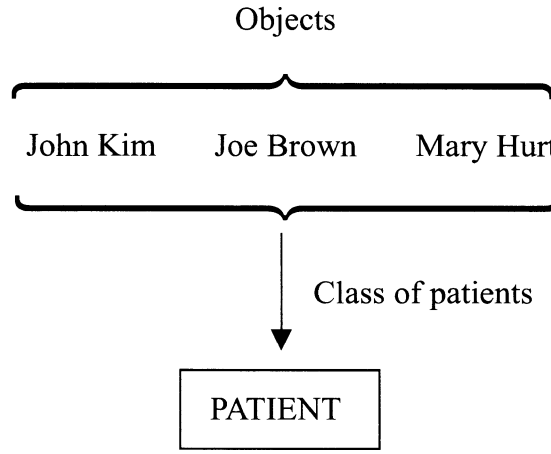


FIGURE 2.1. Class of objects

The term *'class'* is an abbreviation for the term *'object class'* and is often used instead of it.

In accordance with these definitions, we may say that in the process of application development we identify different objects. For each group of objects with similar properties, similar behavior and similar relationships, we create an abstraction called the object class.

Figure 2.1, for example, shows that patients John Kim, Joe Brown and Mary Hurt are objects with similar properties, behavior and relationships. Therefore, for this group of objects we created an object class called PATIENT that describes what is common to all these objects (patients).

Each object in a certain class is called an *object instance* in this class. For this reason we can say that John Kim, Joe Brown, Mary Hurt and other patients are object instances in the class PATIENT.

2.3 Attributes

> An attribute is a data value held by the object in a class. Each attribute has a value for each object instance. Different object instances may have the same or different values for a given attribute. Each attribute name is unique in a class (Rumbaugh et al., 1991).

Therefore, we can say that an attribute is a property of objects in a class. For example, Patient-Id, Name, Address, Age, Weight and Sex are

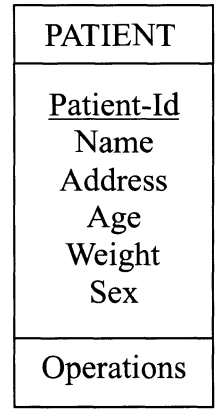


FIGURE 2.2. Class PATIENT

attributes of class PATIENT. Customer-Id, Name, Address and Phone, for example, are attributes of class CUSTOMER.

Each object has one or more attributes that represent and distinguish it from other objects. These attributes are called *candidate keys*. A candidate key is a minimal set of attributes that uniquely identifies and represents an object.

Figure 2.2 shows that attribute Patient-Id is a candidate key, or identifier, in class PATIENT by its being underlined. Therefore, attribute Patient-Id, in accordance with the above-given definition, is a minimal set of attributes that uniquely identifies each object instance in class PATIENT.

2.4 Associations

To analyze the associations or relationships among attributes and also among classes, we use the concept of semantic network modeling. The semantic network concept was developed by Quillian (1968) as a way to model human memory and represent the meaning of words. The essence of this approach is that the meaning of a word—its semantics—can be determined by its relationship to other words (Sanders, 1995).

Three structural primitives of semantic network modeling are used to describe the associations between attributes and classes. These structures are

- Is-part-of,
- Is-associated-with,
- Isa.

The **Is-part-of** structure is used to describe something that can be created or constructed from a number of determined elements or components.

Let us reconsider the following definition, given by Sanders (1995):

> A class of objects can be constructed, described and qualified by the objects' attributes and by other classes (the Is-part-of structure). A class, therefore, is constructed from components of attributes and other classes. This is referred to as the class aggregation abstraction.

From this definition we can say that each class consists of a number of attributes. An attribute in the framework of a certain class *Is-part-of-this class* because it is an element or a component of the class. For example, attribute Patient-Id Is-part-of class PATIENT, Name Is-part-of PATIENT and Address Is-part-of PATIENT. In other words, we may say that a class can be considered as an aggregation of its attributes; see Figure 2.3.

The same concept can be used to analyze the relationships among attributes. An attribute can be considered as an aggregation of other attributes. For example, attribute Street Is-part-of attribute Address, City Is-part-of Address and State Is-part-of Address.

In accordance with the represented definition, a class may contain one or more classes in addition to its attributes.

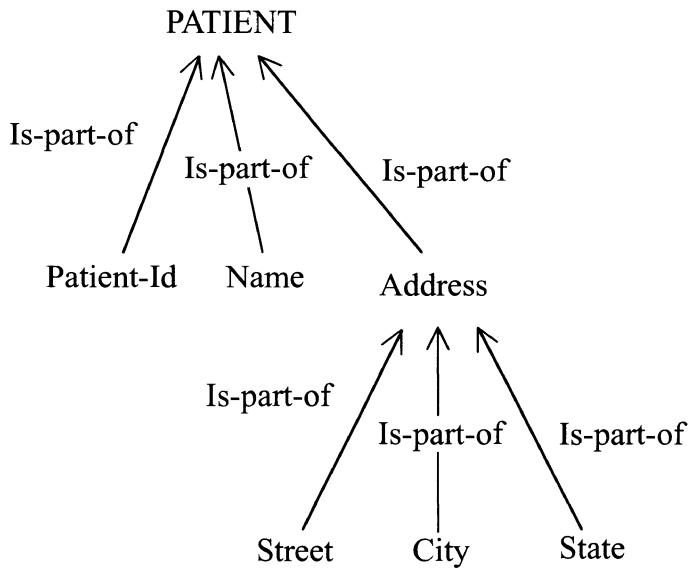


FIGURE 2.3. Is-part-of structure

Figure 2.4 shows the use of the Is-part-of structure to relate class ORDER with its attributes. Furthermore, we see that the Is-part-of structure is used to relate class ORDEREDITEM with class ORDER.

The **Is-associated-with** structure is used to describe the relationships or associations among classes. A class can interact with other classes and can affect the attribute values of other classes through associations or collaborations (the Is-associated-with structure). This is referred to as the *class association abstraction* (Sanders, 1995).

Considering the above definition, we can say that some class is connected with one or more classes. To represent a relationship between two classes, we use the Is-associated-with structure. Such a relationship is called an **association.** An association could be one-to-one, one-to-many or many-to-many.

A one-to-one association exists between two classes where every object instance in each class is related to one and only one object instance in the other class. For example, Figure 2.5 shows that class SHIPMENT Is-associated-with class INVOICE. This is an example of a one-to-one

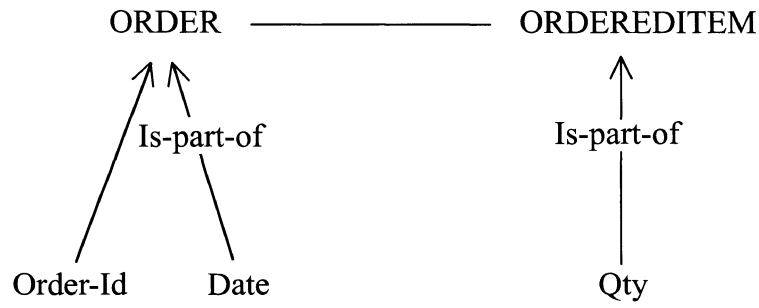


FIGURE 2.4. Is-part-of structure showing one class related to another

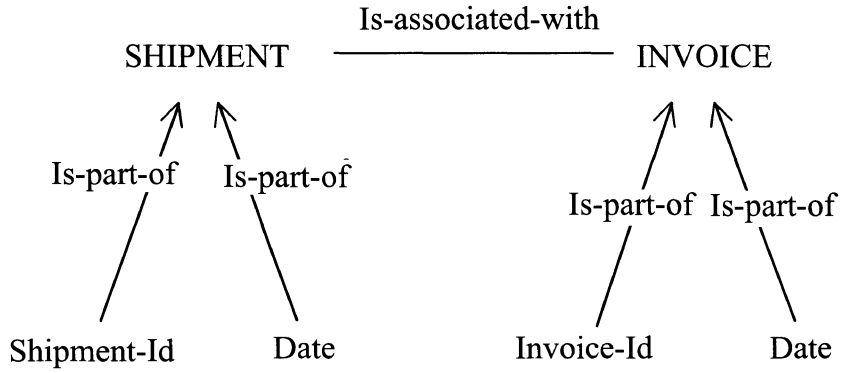


FIGURE 2.5. Is-associated-with structure (one-to-one association)

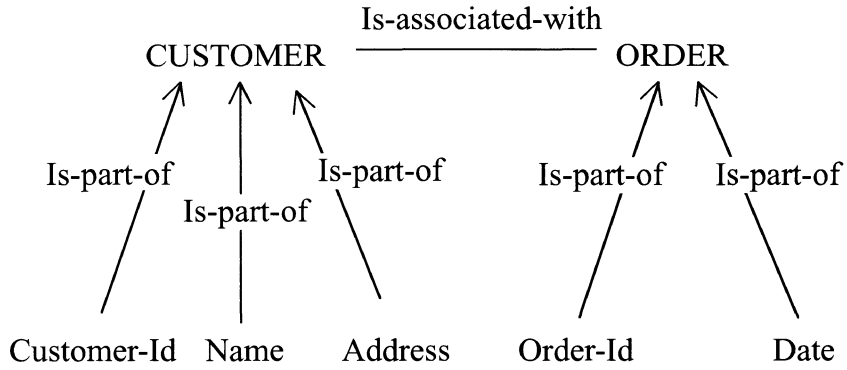


FIGURE 2.6. Is-associated-with structure (one-to-many association)

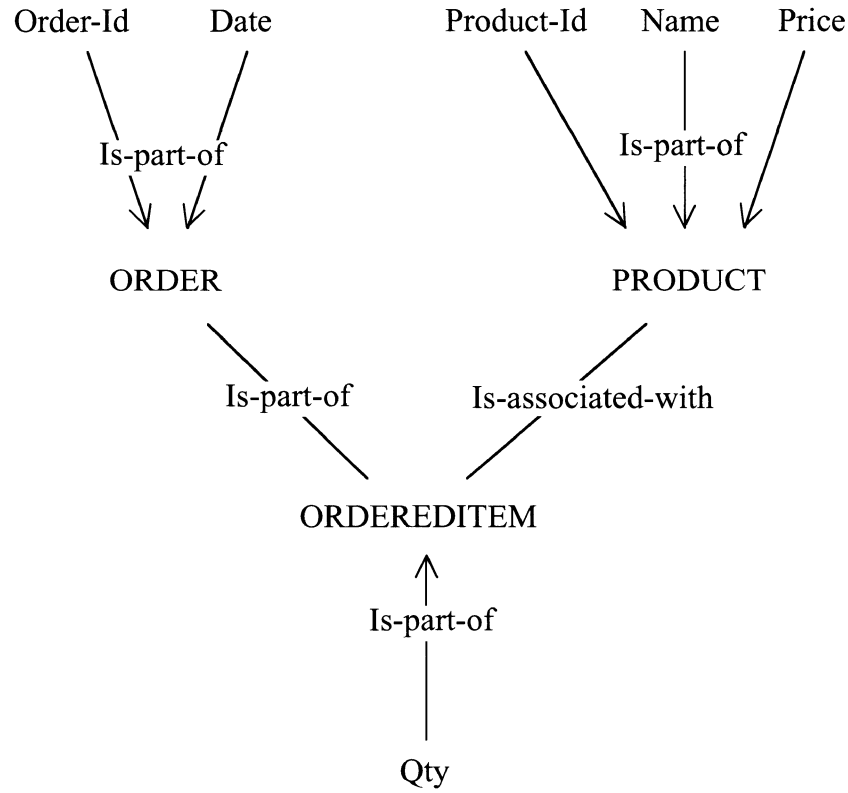


FIGURE 2.7. An association class (many-to-many association)

association, because organizations usually create and send an invoice for each shipment.

A one-to-many association exists between two classes where each object instance in the first class is related to none, one or many object instances in the second class; each object instance in the second class is related to one and only one object instance in the first class.

Figure 2.6 represents an Is-associated-with structure between classes CUSTOMER and ORDER. This is an example of a one-to-many association because each customer may have none, one or more orders; each order is related to only one customer.

A many-to-many association exists between 2 classes where every object instance in one class is related to none, one or more object instances in the other class. In this situation we usually construct such an association as a class; this is an association class.

Figure 2.7 shows an example of a many-to-many association where one order orders one or more products and one product is ordered by none, one, or more orders.

Corresponding to the previous definition given by Sanders (1995), that a class is constructed from components of attributes and other classes, we can say that class ORDEREDITEM is a component of class ORDER because an order contains the ordered items.

As we see from Figure 2.7, the association between the classes ORDER and PRODUCT is constructed as an association class ORDEREDITEM.

Until now we have stressed the Is-part-of and Is-associated-with structures. The **Isa** structure is used to describe inheritance between classes. For this reason, this structure will be discussed in Section 2.6.

2.5 Operations

As discussed in Section 2.1, an object is anything identified in the process of information systems development and recognized by its properties (attributes) and behavior (operations).

> An **operation** is a function or transformation that may be applied to or by objects in a class (Rumbaugh et al., 1991).

In addition to its attributes, each class contains a number of operations, which represent the behavior of objects in this class. Figure 2.8 shows, for example, that Accept, Cancel, and Modify are operations of class ORDER.

Methods specify the way in which operations are encoded in software (Martin, 1993). Thus a method is the implementation of a determined

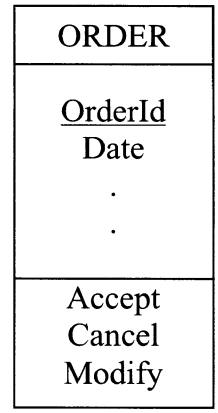


FIGURE 2.8. Class ORDER with its attributes and operations

operation. Therefore, it is an algorithm or program that implements an operation in a class.

The operations of any class can access only the data structures of this class. If a certain class needs to obtain information from another class, it has to send a message to that class and request the kind of information required. For example, if object A needs to use or manipulate data structures of object B, then object A has to call a certain operation of object B.

This is a very important feature of object-oriented modeling called encapsulation. **Encapsulation** means that using the data structures of a certain object is possible only via its own operations. Thus encapsulation hides the data structures of an object from other objects and protects them from corruption.

The same operation may be implemented differently in different classes. Such an operation is referred to as *polymorphic*. This means that the same operation may react differently or may perform different work in different classes. Therefore, **polymorphism** means that the same operation can be implemented in various classes differently to perform different tasks and to give different responses.

2.6 Inheritance

Inheritance is the most powerful feature of object-oriented modeling. Inheritance means identifying and defining a system of hierarchies between classes, where a subclass can inherit attributes and operations from a higher superclass.

A class can be related to other classes through subclasses (the Isa structure). Thus a subclass can inherit attributes from a super-

class, and it can have its own unique attributes. This is referred to as the "class generalization and specialization abstraction" (Sanders, 1995).

Inheritance can be added in two directions: by generalizing common aspects of existing classes into a superclass (bottom up) or by refining existing classes into specialized subclasses (top down) (Rumbaugh et al., 1991).

Generalization means creating a higher superclass for similar characteristics (common attributes and operations) of a determined group of existing classes. The best way to share similar characteristics is by putting them in one place from where they can be used by classes where they are in common. For this purpose we construct a superclass and collect the common characteristics of existing classes in it. The original classes are related to the superclass by the Isa structure, which means that these classes inherit the common characteristics (attributes and operations) from the superclass.

In other words, we can say that the **Isa** structure is used to describe the relationships or associations among the created superclass and the original classes and also to represent that these classes inherit common characteristics from the superclass. Furthermore, the classes can have their own characteristics.

Figure 2.9 represents an example of generalization, where a sales organization receives orders from customers and sends its own orders to different suppliers. In this case we have two kinds of orders that have many similarities: We have the two classes CUST-ORDER and ORG-ORDER with many common characteristics. In such a situation it is important to create a new superclass called ORDER and collect the common characteristics in it. Classes CUST-ORDER and ORG-ORDER are

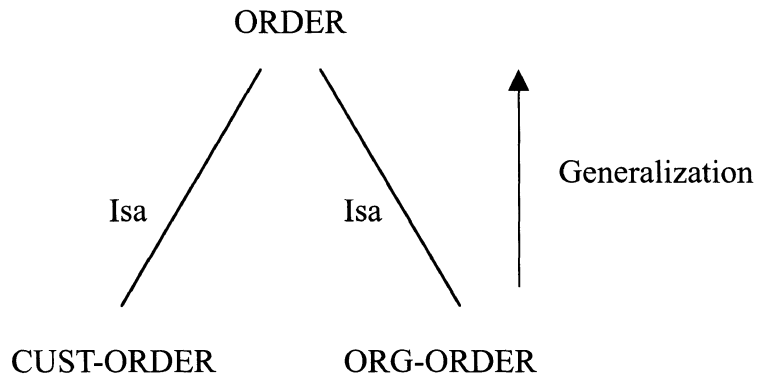


FIGURE 2.9. An example of generalization

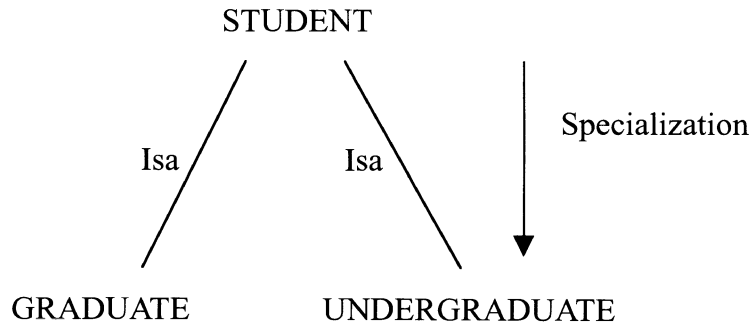


FIGURE 2.10. An example of specialization

related to the superclass ORDER by the Isa structure. For this reason, they inherit all its attributes and operations.

Specialization is refining existing classes into specialized subclasses. The created subclasses inherit the characteristics of the classes to which they are related. In addition, they can have their own unique characteristics. Figure 2.10 shows an example of specialization; class STUDENT is refined into two specialized subclasses, GRADUATE and UNDERGRADUATE.

Generalization and specialization are two different analyses of the same relationship. Generalization is a bottom-up analysis of classes, whereas specialization is a top-down analysis of classes.

Inheritance is a very important and powerful feature of object-oriented modeling. It enables us to reuse the existing data and methods already contained in existing classes. For this reason, it contributes greatly to the reduction of the programming code. In addition, by using inheritance we can perform the needed modifications in one place.

CHAPTER 3

Methodology Preview

The following six chapters of Part II, Methodology, deal with introducing an object-oriented methodology called Tabular Application Development (TAD). This methodology uses several tables to describe the functioning of the organization. These tables are then analyzed in order to identify the necessary changes that have to be implemented to improve and optimize the functioning of the organization. Implementation of the suggested changes in the organization's operation is then put into effect by developing the information system of the enterprise.

The reasons for using tables to represent the real world are as follows:

- the tables are very useful for representing events;
- the tables can be easily surveyed, corrected and extended;
- the tables are very visible and understandable.

TAD methodology is based on the following concept: Any organization is considered to have a number of business processes. Each business process includes one or more work processes. A work process consists of a set of activities. Finally, every activity includes one or more tasks performed by one or more users. A task is an elementary work responsibility performed by a certain employee in the framework of a determined activity.

This concept enables us to discover the real world of the organization. To do this we start by identifying every activity (A) performed in the framework of the organization. This is achieved by organizing interviews with the employees.

After the process of activity identification is completed, we then try to group the identified activities into suitable collections. Each of these collections represents a determined work process (WP). The analyst places the defined work processes into convenient groups. Each group introduces a certain business process (BP). Together all the business processes represent the whole organization.

This concept is simple, understandable and effective and leads the analyst to discover the functioning of the organization in an easy manner.

T. Damij, *Tabular Application Development for Information Systems*

Figure 3.1 shows the concept of TAD methodology used in discovering the functioning of an organization X.

Using the above-described concept, the analyst first identified eight activities by organizing interviews with the employees of the organization; these activities are numbered A1–A8.

The identified activities were then grouped into four work processes: WP1–WP4. Furthermore, the defined work processes were grouped into two business processes, BP1 and BP2, which represent the entire organization X.

An activity could be stated in the framework of one or more work processes. The project team should analyze such a situation and decide whether or not it is possible to remove this redundancy; see Business Process Reengineering in Chapter 6.

TAD methodology has six phases. The first deals with the problem definition. This phase identifies the problem to be solved by creating the entity table as a result of organizing interviews with the management of the organization at different levels. The entity table represents the analyses and outputs related to strategic plan and goals, business goals and objectives, operational goals and objectives and decision support problems.

The second phase identifies the functioning of the system by developing the activity and task tables. The activity table is used to collect information about the activities, work processes and business processes. The task table is used to describe in detailed form the tasks performed

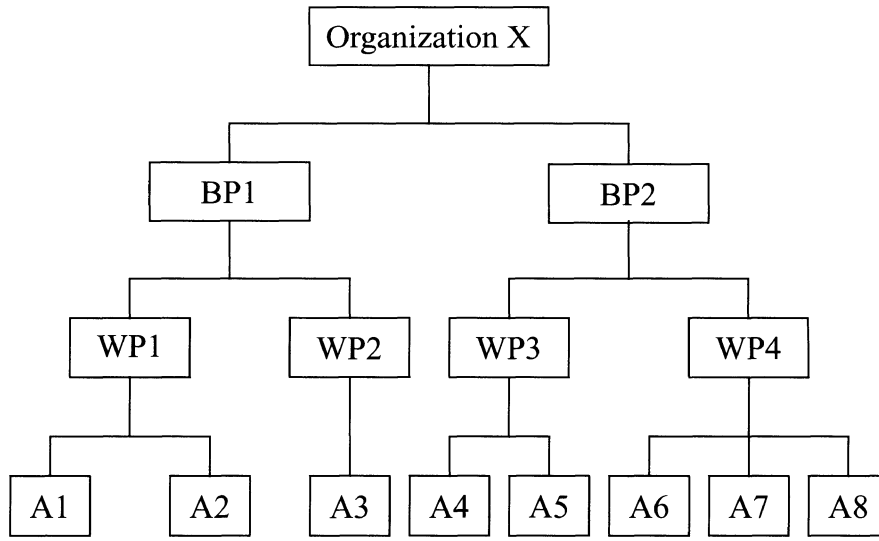


FIGURE 3.1. The real world of the organization

in the framework of the activities, their characteristics and conditions of performance related to them.

The third phase discusses business process reengineering by analyzing the real world of the organization defined in the tables created in the previous phases, identifying changes and trying to find solutions to the existing problems.

The fourth phase identifies the objects of the system, their attributes and their associations from the information, which is collected by the analyst and registered in the entity, activity and task tables through the process of interviewing the users. The result of this phase is the object model of the system.

The fifth phase defines the operations of the object model by analyzing the content of the activity and task tables. Furthermore, this phase deals with designing the system by transforming the information collected in the entity and activity tables into an application model of the system. In addition, we write algorithms to define in detail the operations of the classes of the object model and the created application model.

The sixth phase implements the system. This is achieved by implementing the object and application models of the system.

PART II

METHODOLOGY

Part II has six chapters. Each of these chapters discusses a specific phase of TAD methodology.

Chapter 4 deals with the first phase; this is problem definition. Chapter 5 represents the second phase: systems functioning. Chapter 6 discusses the third phase: business process reengineering. Chapter 7 deals with the fourth phase, the object model. Chapter 8 introduces the fifth phase: design. Finally, Chapter 9 represents the sixth phase, implementation.

CHAPTER 4

Problem Definition

The first phase of TAD methodology is problem definition. In this phase, the real world of the problem to be solved is identified and reduced to understandable terms. The best way to begin this phase is to interview all users. We first organize meetings and interviews with management and then continue with other users.

We start the process of interviews with the top management. The reason for this starting point is to start the process of information systems, development by analyzing the strategic plan of the organization and defining the strategic goals. Usually we need two or more meetings with the top management to define its requirements and information needs and also to answer our questions about the organization and to understand the information required in its strategic and daily work.

Therefore, in this phase we focus on defining the management's requirements. To achieve this goal, we have to give the management enough time to analyze its information needs.

The result of the interviews is to define a number of analyses that offer information about the strategic plan and goals, business objectives and goals and operational objectives.

To summarize, the purpose of interviewing the top management is to identify

- the organization's strategic plan and goals,
- vital outputs and analyses related to the strategic plan and goals,
- the organization's decision support problems and
- the organization's structural scheme.

After defining the information at the enterprise level, we continue with the management first at the business and then at the operational levels (department or unit). To do this we create a plan of interviews in accordance with the hierarchical scheme of the organization. This plan has to be discussed with the top management and created according to its recommendations.

T. Damij, *Tabular Application Development for Information Systems*

To accomplish this plan, we have to get the necessary approval and support of the top management. This approval is essential for conducting successful interviews. It is also extremely important to inform each interviewee of the top management's approval and support before beginning the interviews.

The purpose of interviewing the management at the business and operational levels is to obtain information about

- business objectives and goals,
- operational objectives and goals,
- the structure of each department or unit (entity),
- vital outputs and analyses related to business and operational objectives and goals and
- the entity's decision support problems.

TAD methodology uses the term "entity" to define a user, group of users, unit, department or anything of importance in the system's functioning. An entity is any source of information that is part of the system or is connected with the system by some interaction. Therefore, an entity may be internal or external. An internal entity is inside the system and takes part in the system's operation. An external entity is not part of the system, but it has one or more interactions with the system. Thus an entity may be a user or any other source that sends input to the system, participates in some activity or task in the system or receives output from the system.

Interviews should be organized only with the internal entities. Internal entities inform us about the behavior of external entities.

Identification of the entities, their goals and objectives, the needed outputs and analyses, and their decision support problems will be achieved by developing the **entity table.** This table is developed and completed during the interviews with the management at the enterprise (top), business and operational levels.

The entity table is structured as follows: The columns of the table represent the entities and the rows of the table represent the analyses the entities require. An asterisk in any square(i,j) in the entity table means that the entity defined in column j requires the analysis defined in row i, where i ranges from 1 to the number of analyses and j ranges from 1 to the number of entities.

To implement TAD methodology, we will use an interesting application. This application carries out business process reengineering and information systems development in a tourist organization.

4.1 Tourist Organization

The Tourist Organization is a large organization that owns seven tourist centers. Each center has at least one hotel and other tourist facilities. One of these centers will be used as a case study to implement the ideas of TAD methodology.

This center has a beautiful, small, classic hotel and another housing facility, which is bigger and more modern. It also has a number of other buildings.

The organization has a computer application system installed in the Information Office. All other departments send their documents to the Information Office and get the necessary outputs. This office takes care of inserting and updating the data in the database as well as printing different outputs from it.

Corresponding to the first phase of TAD methodology, we began by organizing two interviews with the top management. As a result of the first interview we asked the top management to define the strategic plan and goals of the enterprise and also to identify a number of vital analyses related to the strategic plan and goals. In addition, we identified the structure of the organization, which consists of the following entities: Top Management, Information Office, Purchasing, Marketing, Accounts, Reception, Buffet and Restaurant.

At the second meeting with the top management we discussed the plan for the interviews; we prepared the plan for this meeting, taking into consideration the structural scheme of the organization. In addition, we discussed the list of analyses and outputs the top management prepared after the first meeting and introduced as essential for its daily and strategic work.

Table 4.1. Entity table of Tourist Organization

Requirements	Entity / Analysis	Top Management	Purchasing Dept.	Marketing Dept.	Accts. Dept.	Restaurant
Reports	1. Information about expenses in each center	★	★	★	★	
	2. Information about turnover of each center	★		★	★	
	3. Information about guests	★		★	★	★
	4. Information about contracts	★		★	★	
	5. Information about suppliers	★	★		★	★
	6. Information about hotel occupation	★	★	★	★	★
	7. Information about guests' complaints	★		★		★
Decision Support Problem	8. Nutrition planning	★	★			★

After developing part of the entity table according to the requirements and expectations of the top management, we continued our work by organizing several interviews with the management of the previously mentioned departments to define their information needs and decision support problems.

Table 4.1 shows the entity table of the Tourist Organization. It shows the organization's structure and some important analyses connected with its strategic, business and operational goals. This table introduces only part of the real entity table. It therefore contains only seven analyses and one decision support problem required by the top management and other important entities. Entities such as Reception and Buffet are not defined in Table 4.1 because the management of these entities did not require any analysis.

CHAPTER 5

Systems Functioning

The second phase of TAD methodology deals with discovering and understanding the functioning of the organization for which we intend to develop an information system or in which we wish to carry out business process reengineering.

An essential precondition for the development of an information system or for the successful implementation of business process reengineering in an organization is to understand completely the current world of the modeled organization. The analyst has to form a clear picture of the functioning of the system as a whole, the functioning of every part of the system and the work of every group of users, or individual user. In addition, the analyst has to define in detail every task or event that actually occurs in the organization.

In order to achieve this goal, we have to continue the work that we started in the first phase. This means that we have to organize at least one interview with the representative of each entity, group of users and, if necessary, with each user of the system.

The purpose of this phase is to understand the functioning of the system first by identifying every atomic activity in the system, second by grouping these activities into suitable groups of work processes and third by grouping work processes into suitable groups of business processes.

This phase has three steps. The first step identifies the activities of the system. The second step describes in detail the tasks performed in the framework of the identified activities. The third step defines the work processes by grouping the activities of the system and also defines the business processes by grouping work processes together.

5.1 Activities

The aim of the interviews with the management at different levels organized in the first phase is to complete the entity table of the system

T. Damij, *Tabular Application Development for Information Systems*

with the analyses related to the strategic, business and operational goals and objectives. The aim of the interviews organized in the second phase is to define the functioning of the system.

The first step deals with identifying and analyzing every activity performed in each part of the organization. Identification and analysis of the organization's activities are achieved by organizing further interviews with the representatives of every internal entity in the framework of the organization. This is important when we are dealing with developing an information system for the whole organization. Otherwise, we have to concentrate our analysis on only the part of the organization concerned.

Actually, the first phase and the current step are connected and performed simultaneously, because along with creating the entity table in the first phase we begin defining the work of the management at different levels and register every activity performed. However, along with developing the activity table during the interviews with the users in the *current* step, we update, extend and complete the entity table with any new user requirements, information needs and decision support problems. We need to identify and exactly describe each user's activities performed inside the identified entities.

This step deals with the development of the activity table, which is essential for identifying and defining the functioning of the system. In addition, this table represents a solid foundation for all further steps and phases of TAD methodology.

Along with the activity table, we develop another table—the task table—as discussed in the next step. In practice, we usually develop these tables simultaneously, but to keep the presentation simple and clear, we prefer to introduce them separately.

5.1.1. ACTIVITY TABLE DEVELOPMENT

An effective way to identify the activities or any event that occurs in the system is achieved by developing an **activity table.** Using a table to represent the real world is very simple, visible and understandable. As mentioned, as we develop the activity table we simultaneously develop another table, the task table, which is very important in defining detailed information about tasks. The development of this table is discussed later.

The activity table is organized as follows: The entities of the system are represented in the columns of the table and the activities are listed in the rows of the table. Each activity occupies one row and each entity occupies one column of the table.

An entity may represent different users at different levels; for example, top management, a department of the organization, a unit of the organization, a group of users or an individual user in a determined department or unit.

Such organization of the activity table enables us to transform the real world into the table. This work is very effective in defining the linkages involved, first among the entities of the system that are concerned in carrying out a determined activity (horizontal linkage), and second among the activities (more precisely, tasks) performed by any particular internal entity (vertical linkage).

A non-empty square(i,j) in the activity table represents a certain task performed by an entity defined in column j in the framework of the activity defined in row i, where i ranges from 1 to the number of activities and j ranges from 1 to the number of entities.

From the structure of the activity table we can give a definition of the terms "task" and "activity." A *task* is elementary work performed by a determined entity in the framework of a certain activity. An *activity* is a collection of tasks connected with an input or output and is performed by one or more entities.

Developing the activity table is a result of interviews organized with the internal entities defined in the columns of the table. In the rows of the activity table we first register each activity identified during an interview and then link this activity with the entities (in the columns) that cooperate in carrying it out. To enter a newly identified activity in the activity table, we first register this activity in the current row of the table and then connect the entered activity with those entities in the columns that perform a task in the framework of this activity.

After every interview, we have to rethink the activity table and list all activities one by one in order to sort them in the order in which they occur in the real world. To make the activity table represent the real world, we link the activities horizontally and vertically. The purpose of defining horizontal and vertical connections is to define the order in which they occur in the real world.

Horizontal linkage means that each activity must be connected with those entities in the columns that are involved in performing tasks in the framework of a particular activity. For this purpose we use the following procedure:

> For every activity defined in row i, where i ranges from 1 to the number of activities, we list the entities in the columns and try to link the current activity with each of these entities. If any connection exists between activity(i) and entity(j), where j ranges from 1 to the number of entities, then letter S or T is written in square(i,j).

Letter S in square(i,j) means that entity(j) is a source entity for activity(i). This entity(j) performs a determined task in the framework of activity(i) (creates, completes, sends and so on) defined in square(i,j). Letter T in square(i,j) means that entity(j) is a target entity for activity(i). This entity(j) accepts or registers an output from source entities.

Usually, each activity is connected with two entities: the source and target entities. An activity could be related to only one entity. This occurs when an entity performs determined work for its purpose and is not linked to any other entity. On the other hand, an activity could be connected with more than two entities. This is possible when an entity (entities) receives input from another entity (entities) and sends it to one or more entities.

Therefore, any activity may have one or more source entities and also a number of target entities. For this reason the letters S and T, used to indicate a certain activity, are also indexed by the index of the source entities of the task as it occurs in the framework of the treated activity.

For example, considering activity(i), let us define that entity(x) is a source and entity(y) is a target entity. To do so we write S_x in square(i,x) and T_x in square(i,y), where x and y are indices of the source entity x and target entity y, as the following table shows.

Entity / Activity	$Entity_x$	$Entity_y$
$Activity_i$	S_x	T_x
$Activity_j$		

Vertical linkage exists among tasks performed in the framework of different activities by a determined internal entity. In this manner we use the procedure stated below:

> For every internal entity defined in column j, where j ranges from 1 to the number of internal entities, we list all activities in the rows. For every activity defined in row i, where i ranges from 1 to the number of activities, we try to determine if activity(i) is a predecessor of one or more activities indicated by non-empty squares in column j. If so, then letter P is written in square(i,j) and letter U is written in each square that is indicated as a successor of activity(i).

In other words, considering only the internal entities, we use the letters P and U to connect the activities in which a certain internal entity is involved. This is carried out in order to connect the tasks of any internal entity defined in a determined column.

Letter P in square(i,j) means that activity(i) is a predecessor of some activity (activities) indicated by U in column j. Letter U in square(i,j)

means that activity(i) is a successor to another activity (activities) indicated by P in column j.

Any activity may have one or more predecessors and also one or more successors. To solve this problem, we use the index of the predecessor activity to link the predecessor activity with its successors. For this reason, the letters P and U are indexed by the index of the predecessor activity. For example, considering entity(x), let us define that activity(i) is a predecessor to activity(j). To do this we write P_i in square(i,x) and U_i in square(j,x), where i and j are indices of the predecessor activity i and successor activity j:

Entity / Activity	$Entity_x$	$Entity_y$
$Activity_i$	P_i	
$Activity_j$	U_i	

5.1.2. TOURIST ORGANIZATION

To develop the activity table and complete the entity table of the Tourist Organization, we organized interviews at the department level, corresponding to the plan of interviews created in the previous phase.

In addition to the interviews with the top management entity, we organized interviews with the following entities: Information Office, Purchasing, Marketing, Accounts, Reception, Buffet and Restaurant. In every department, we first interviewed the management of the department and then continued with other users. A number of users were interviewed twice in order to understand their daily work. The results of these interviews were used to create the activity table of the organization. After every interview we rethought and rewrote the activity table.

Table 5.1 shows a reduced activity table, which has 19 activities and 10 entities. The first eight entities are internal and the last two are external. The last entity is "Guest/Customer," where "Guest" means a particular guest and "Customer" means an organization that signs a contract with the Tourist Organization for its employees.

The first activity, "Make contracts with suppliers," means that the Purchasing Department gets a contract from the supplier and sends it to management for approval. For this reason we write S_9 in square(1,9), T_9 in square(1,3), S_3 in square(1,3) and T_3 in square(1,1).

The second activity, "Confirm supplier's contract," means that management signs the contract and sends it back to the Purchasing Depart-

Table 5.1. Activity table of Tourist Organization

Activity \ Entity	1. Top Management	2. Information Office	3. Purchasing Dept.	4. Marketing Dept.	5. Accounts	6. Reception	7. Buffet	8. Restaurant	9. Supplier	10. Guest/ Customer
1. Make contracts with suppliers	T_3 P_1		T_9,S_3 P_1						S_9	
2. Confirm supplier's contract	S_1 U_1	T_3	T_1,S_3 U_1,P_2						T_3	
3. Create an order		T_3,T_7,T_8	S_3 U_2,P_3				S_7 P_3	S_8 P_3	T_3,T_7,T_8	
4. Accept shipment		T_9					T_9 U_3	T_9 U_3	S_9	
5. Accept supplier's invoice	T_5	T_5	T_5 U_3,P_5		T_9,S_5				S_9	
6. Confirm invoice			S_3 U_5		T_3 P_6					
7. Make supplier's payment		T_5			S_5 U_6				T_5	
8. Create advertising material	T_4,S_1 P_8			S_4,T_1,S_4 P_8		T_4 P_8				T_4
9. Accept demand	T_{10} P_9			T_{10} P_9		T_{10}				S_{10}
10. Send offer	S_1 U_8,U_9,P_{10}	T_4		T_1,S_4 U_8,U_9,P_{10}						T_4
11. Sign contracts with customers	S_1 U_{10}	T_4		T_1,S_4 U_{10}						T_4
12. Accept reservation						T_{10} U_8,P_{12}				S_{10}
13. Make hotel occupation plan						S_6 U_{12}				
14. Register guests		T_6				T_{10},S_6 U_{12},P_{14}				S_{10}
15. Make plan of activities	T_6					S_6 U_{12},U_{14}				
16. Create customer's invoice		T_6			T_6 P_{16}	S_6 U_{14},P_{16}				T_6
17. Accept customer's payment					T_6 U_{16}	T_{10},S_6 U_{16}				S_{10}
18. Register lost property		T_6				S_6 U_{14},P_{18}				
19. Return lost property		T_6				S_6 U_{18}				T_6

ment. This department sends one copy of the contract to the supplier and one copy to the Information Office. This is why we write S_1 in square(2,1), T_1 in square(2,3), S_3 in square(2,3) and T_3 in square(2,2) and square(2,9).

The third activity, "Create an order," means that Purchasing, Buffet or Restaurant may create an order. Each of these entities sends a copy of the order to the supplier and another copy to the Information Office. To indicate this we write S_3 in square(3,3), T_3 in squares (3,2) and (3,9) S_7 in square(3,7), T_7 in squares (3,2) and (3,9), S_8 in square(3,9) and T_8 in squares (3,2) and (3,9).

All other activities are defined using the same procedure.

Furthermore, concerning the Purchasing Department entity, we find that the first activity is a predecessor to the second activity. For this reason we write P_1 in square(1,3) and U_1 in square(2,3). The second activity is a predecessor to the third activity. Therefore, we write P_2 in square(2,3) and U_2 in square(3,3). All tasks are connected horizontally and vertically using the same concept.

The reader may notice that the eighth activity is complex and that it is difficult to follow its tasks. This is because square(8,4) contains S_4 two times, so it is not clear which of them is connected to T_4 defined in squares (8,6) and (8,10).

The reason for this situation is that the eighth activity actually represents two activities: "Create advertising material" and "Confirm advertising material." The first is performed by the Marketing Department, which then sends it to Top Management. The second is accomplished by Top Management, which sends it back to the Marketing Department. This department then forwards it to the Reception and Guest.

This is a very interesting matter, which proves that the activity table forces the analyst to define activities at an elementary level. This means that she or he has to follow activities as they happen in the real world. Otherwise, the activity row of the table is complex and too difficult to understand.

We prefer to leave the eighth activity complex; we do not split it into the two mentioned activities so that we can warn the analyst about such situations. (In addition to Table 5.1, a new table is created, Table 5.1a, which shows the eighth activity split into two activities: "Create advertising material" and "Confirm advertising material.")

This proves that using the activity table to represent the functioning of the system is an excellent tool that enables us to analyze and check our work. Even if we made a mistake and defined a complex activity, we do not need much effort to discover that something is wrong with the defined activity. Such an activity would be very evident just by looking at the developed activity table to find out that the indicated activity is too difficult and could represent two or more merged activities.

Table 5.1a. Modified activity table of Tourist Organization

Entity / Activity	1. Top Management	2. Information Office	3. Purchasing Dept.	4. Marketing Dept.	5. Accounts	6. Reception	7. Buffet	8. Restaurant	9. Supplier	10. Guest/ Customer
1. Make contracts with suppliers	T_3 P_1		T_9,S_3 P_1						S_9	
2. Confirm supplier's contract	S_1 U_1	T_3	T_1,S_3 U_1,P_2						T_3	
3. Create an order		T_3,T_7,T_8	S_3 U_2,P_3				S_7 P_3	S_8 P_3	T_3,T_7,T_8	
4. Accept shipment		T_9					T_9 U_3	T_9 U_3	S_9	
5. Accept supplier's invoice	T_5	T_5	T_5 U_3,P_5		T_9,S_5				S_9	
6. Confirm invoice			S_3 U_5		T_3 P_6					
7. Make supplier's payment		T_5			S_5 U_6				T_5	
8. Create advertising material	T_4 P_8			S_4 P_8						
9. Confirm advertising material	S_1 U_8,P_9			T_1,S_4 U_8,P_9		T_4 P_9				T_4
10. Accept demand	T_{10} P_{10}			T_{10} P_{10}		T_{10}				S_{10}
11. Send offer	S_1 U_9,U_{10},P_{11}	T_4		T_1,S_4 U_9,U_{10},P_{11}						T_4
12. Sign contracts with customers	S_1 U_{11}	T_4		T_1,S_4 U_{11}						T_4
13. Accept reservation						T_{10} U_9,P_{13}				S_{10}
14. Make hotel occupation plan						S_6 U_{13}				
15. Register guests		T_6				T_{10},S_6 U_{13},P_{15}				S_{10}
16. Make plan of activities	T_6					S_6 U_{13},U_{15}				
17. Create customer's invoice		T_6			T_6 P_{17}	S_6 U_{15},P_{17}				T_6
18. Accept customer's payment					T_6 U_{17}	T_{10},S_6 U_{17}				S_{10}
19. Register lost property		T_6				S_6 U_{15},P_{19}				
20. Return lost property		T_6				S_6 U_{19}				T_6

5.2 Tasks

We have described how to define the activities of the system. Each activity consists of one or more tasks and occupies one row of the activity table. Tasks of any activity are indicated by non-empty squares in the activity row of the activity table. As mentioned before, a task is work or an event performed by a certain entity in the framework of a determined activity.

We often find ourselves in situations where we need more information about the activities. Furthermore, we especially need detailed information about every task performed in the framework of any activity defined in the activity table. The purpose of this information is to identify all the circumstances in which each task is accomplished and to define the preconditions or conditions that need to be checked, fulfilled or tested through performing the task.

This step deals with giving detailed information about tasks performed by the internal entities only. This is achieved by developing the task table.

5.2.1. TASK TABLE DEVELOPMENT

An effective way to identify tasks, their characteristics and the circumstances linked with them is by developing the **task table.** As mentioned earlier, the task table is developed at the same time as the activity table.

We develop these two tables part by part, through interviews with users. After every interview, we rethink both tables and define the order in which the activities are listed in the activity table as they occur in the real world. We also carefully describe every task defined in the task table.

The task table is organized as follows: The tasks are represented in the rows of the table and the characteristics of the tasks are defined in the columns. Thus, each task, defined by a non-empty square in the activity table, occupies one row in the task table.

Each task is represented by its code K_{ij}, where the letter K means task, and i and j are indices of row i and column j of the activity table where the task is indicated. K_{ij} means a task defined in the framework of activity(i) performed by internal entity(j).

In the columns of the task table we define task characteristics in terms of

- description,
- time,
- condition and
- input/output.

Description: The Description column is used to write a short and essential description of the particular task defined in the current row of the task table.

Time: The Time column is used to determine a value of the time for each task to indicate that entity(j) defined in the activity table needs the determined time value to perform task K_{ij}. Time values collected in the Time column of the task table may become a very useful parameter should we wish to use it in the process of business process reengineering, as discussed later.

Usually it is difficult to get information about the time needed to perform a determined task. Despite this difficulty, we try to obtain answers from users, at least about those tasks with particularly long task durations.

Condition: The Condition column is used to express that the performance of the task in progress K_{ij} requires that one or more conditions be fulfilled. Therefore, in this column, we very carefully define every condition connected with the task defined in the current row of the task table. We also define conditions that link the current task to its predecessors if such linkages exist.

Input/Output: The Input/Output column is used to indicate which inputs or outputs either are needed to perform the current task or are connected in any way with task K_{ij}.

In addition to these characteristics, the analyst may define other interesting task characteristics if necessary.

Along with the tasks' characteristics, the task table represents the activities defined in the activity table. The purpose of this congruence is to indicate in detail the linkages among the activities and their tasks. Therefore, in the first column of the task table we list the activities as they are defined in the activity table. As mentioned above, each task occupies one row in the task table. For this reason, each activity occupies one or more rows, depending on the number of tasks performed in the framework of the activity.

Developing the activity and task tables is an iterative process. Some interviews have to be repeated a number of times to arrive at a precise understanding of the user's work. If anything is not understandable, we have to organize a new interview with the responsible users until everything is clear.

5.2.2. TOURIST ORGANIZATION

We developed the task table of the Tourist Organization, part by part, along with the interviews organized with the internal entities of the or-

ganization. At each interview we extended the activity table for the activities mentioned by the interviewee and we simultaneously extended the task table for the tasks performed in the framework of new activities.

Together with the interviewee, we defined all the characteristics of every new task: a short and clear description, the time value needed for its performance, conditions connected with it, and inputs and outputs accepted, created or sent by it.

Table 5.2 represents only part of the organization's task table because the real task table contains many tasks and there is no need to show all of them. In addition, the table presented clearly shows the importance and the means of developing the task table. The activity table of the Tourist Organization (Table 5.1a) contains 20 activities. To represent the organization's task table, we describe in detail only a few of these activities and define the tasks performed in the framework of these activities.

The first column of Table 5.2 shows only six activities defined in the activity table (Table 5.1a). The first activity has two tasks: $K_{1,1}$ and $K_{1,3}$.

Table 5.2. Task table of Tourist Organization

Activity	Characteristic Task Code	Description	Time	Condition	Input/ Output
1. Make contracts with suppliers	$K_{1,3}$	Purchasing Dept. accepts contract from supp. and sends it to Mgmt. for confirmation			Contract
	$K_{1,1}$	Mgmt. accepts supp. contract from Purchasing Dept.			Contract
2. Confirm supplier's contract	$K_{2,1}$	Management confirms contract and sends it back			Contract
	$K_{2,3}$	Purchasing accepts confirmed contract and sends it to supp.			Contract
3. Create an order	$K_{3,3}$	Purchasing Dept. creates order and sends it to supplier		Check if the supplier has a contract	Order
	$K_{3,7}$	Buffet sends order to supplier		Check if the supplier has a contract	Order
	$K_{3,8}$	Restaurant sends order to supplier		Check if the supplier has a contract	Order
4. Accept shipment	$K_{4,7}$	Buffet accepts supplier's shipment		Check the ordered products	Shipment
	$K_{4,8}$	Restaurant accepts supplier's shipment		Check the ordered products	Shipment
5. Accept supplier's invoice	$K_{5,5}$	Accounts accepts supplier's invoice and sends it to the Purchasing Dept. and Management for confirmation		Check supplier's shipment	Invoice
6. Confirm invoice	$K_{6,3}$	Purchasing Dept. accepts invoice, confirms it and sends it to Accounts			Invoice

The second activity includes two tasks, $K_{2,1}$, $K_{2,3}$. The third activity contains three tasks: $K_{3,3}$, $K_{3,7}$ and $K_{3,8}$. The fourth activity has two tasks, $K_{4,7}$ and $K_{4,8}$. The fifth activity contains three tasks: $K_{5,1}$, $K_{5,3}$ and $K_{5,5}$. The sixth activity has one task, $K_{6,3}$.

All other tasks are defined using the same procedure.

As the reader may notice, tasks $K_{5,1}$ and $K_{5,3}$ are not defined in Table 5.2. Actually, we can define these tasks in particular rows as "Top Management accepts supplier's invoice from Accounts" and "Purchasing Department accepts supplier's invoice from Accounts." This information is redundant because it is contained in the description of task $K_{5,5}$. For this reason, the definition of such tasks in the task table could be ignored. For the same reason we left task $K_{6,5}$ undefined in Table 5.2. In addition, we ignored the tasks defined in the second column of the activity table, because the Information Office only stores documents connected with the tasks performed by other entities.

5.3 Work Processes and Business Processes

After completing the process of interviews with the users, we discuss and complete the activity and task tables. This goal is achieved by organizing a joint meeting with the main, important or all representatives of the internal entities defined in the columns of the activity table.

The aim of the joint meeting is to discover and correct any mistakes in the activity and task tables and also to persuade users to cooperate in completing these tables. We first introduce the content of the activity and task tables to the users. Usually we start by explaining the activities defined in the activity table one by one. For every activity we introduce the tasks performed in the framework of the activity. Furthermore, for every task we carefully explain the circumstances linked with it.

If any disagreement or incorrectness is noticed about any activity or task, then it has to be discussed until the misunderstanding or mistake is corrected and the situation is made completely clear.

A very important result of the analyst's presentation or discussion of each activity defined in the activity table is to check carefully its vertical linkages with other activities. This means that we try, together with users, to control whether the tasks in every column of the activity table (considering only the columns of the internal entities) are correctly connected vertically. These carefully defined relationships among the activities lead to the creation of the order in which the events of the system occur in the real world.

These relationships enable us to develop a process model of the system, as discussed later. Furthermore, the defined connections among the

activities are very important should we wish to develop other diagrams such as an interaction diagram.

When the content of the activity table is discussed and the representatives agree that the activity and task tables reflect the real world, then the main part of the joint meeting is completed.

In the second part of the joint meeting, the analyst introduces to the representatives how he or she grouped the activities defined in the activity table into suitable groups. Actually, the analyst prepares a suggestion about grouping the activities before the joint meeting takes place. Therefore, before the meeting, the analyst has to rethink and analyze the activity table carefully and prepare a tentative grouping of the activities into named groups. Such a suggestion is then discussed and corrected by the representatives in the meeting. Usually, users suggest moving some activities from one group to another and also naming the groups by more convenient names.

The result of the second part of the meeting is to place the activities of the activity table in suitable groups. Each of these groups or collection of activities introduces a work process of the system.

A **work process** is a collection of one or more activities, which are usually followed in a determined order in carrying out a distinguishable set of tasks to produce a well-defined output or result. From this definition we may conclude that each work process occupies one or more rows in the activity table, depending on the number of activities it contains.

The analyst usually goes on with this work and tries to group the work processes into suitable collections; each of these collections is a **business process.** Such a suggestion for defining business processes must be discussed with, and may be corrected by, the representatives in the joint meeting.

From this explanation we may conclude that each business process consists of one or more successive work processes and occupies one or more rows in the activity table.

> A business process is a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer (Hammer, Champy, 1993).

Each business process actually represents a determined subsystem of the organization. Developing the activity and task tables leads to the discovery of all relevant activities and their tasks; all work processes are defined by grouping the activities in convenient collections and all business processes are identified by grouping the work processes in appropriate collections. For this reason we can say that creating the activity and task tables leads to the discovery of the whole system and its subsystems.

5.3.1. TOURIST ORGANIZATION

After completing the interviews we organized a joint meeting with representatives from the identified entities of the Tourist Organization. At this meeting we presented their work corresponding to the information collected in the activity table. The purpose was to persuade them to follow our explanation and to correct any mistakes in the activity table.

In Table 5.3 we give the final activity table of the Tourist Organization as a result of the joint meeting with the representatives of the organization. In our example the final task table is the same as Table 5.2, because no changes have been made to the content of this table in the framework of the current step. These tables are completed by defining the work processes and also by grouping these work processes into the business processes of the systems.

Table 5.3 introduces the final activity table of the Tourist Organization. This table has 20 activities, which are grouped in four work processes. The first work process is "Purchasing," which consists of the first seven activities and looks after settling business with suppliers. The second work process is "Marketing," which has five activities and deals with obtaining customers. The third work process is "Register," which has four activities and deals with residing guests. The last work process is "Departure," which contains four activities and deals with guests' departure. The first two work processes, Purchasing and Marketing, are grouped into a business process called "Sales." Furthermore, the last two work processes, Register and Departure, are also grouped into a particular business process, called "Guest."

Table 5.3. Final activity table of Tourist Organization

Business Process	Work Process	Entity / Activity	1. Top Management	2. Information Office	3. Purchasing Dept.	4. Marketing Dept.	5. Accounts	6. Reception	7. Buffet	8. Restaurant	9. Supplier	10. Guest/ Customer
Sales	1. Purchasing	1. Make contracts with suppliers	T_3 P_1		T_9,S_3 P_1						S_9	
		2. Confirm supplier's contract	S_1 U_1	T_3	T_1,S_3 U_1,P_2						T_3	
		3. Create an order		T_3,T_7,T_8	S_3 U_2,P_3				S_7 P_3	S_8 P_3	T_3,T_7,T_8	
		4. Accept shipment		T_9					T_9 U_3	T_9 U_3	S_9	
		5. Accept supplier's invoice	T_5	T_5	T_5 U_3,P_5		T_9,S_5				S_9	
		6. Confirm invoice			S_3 U_5		T_3 P_6					
		7. Make supplier's payment		T_5			S_5 U_6				T_5	
	2. Marketing	8. Create advertising material	T_4 P_8			S_4 P_8						
		9. Confirm advertising material	S_1 U_8,P_9			T_1,S_4 U_8,P_9		T_4 P_9				T_4
		10. Accept demand	T_{10} P_{10}			T_{10} P_{10}		T_{10}				S_{10}
		11. Send offer	S_1 $U_9U_{10}P_{11}$	T_4		T_1,S_4 $U_9U_{10}P_{11}$						T_4
		12. Sign contracts with customers	S_1 U_{11}	T_4		T_1,S_4 U_{11}						T_4
Guest	3. Register	13. Accept reservation						T_{10} U_9,P_{13}				S_{10}
		14. Make hotel occupation plan						S_6 U_{13}				
		15. Register guests		T_6				T_{10},S_6 U_{13},P_{15}				S_{10}
		16. Make plan of activities	T_6					S_6 U_{13},U_{15}				
	4. Departure	17. Create customer's invoice		T_6			T_6 P_{17}	S_6 U_{15},P_{17}				T_6
		18. Accept customer's payment					T_6 U_{17}	T_{10},S_6 U_{17}				S_{10}
		19. Register lost property		T_6				S_6 U_{15},P_{19}				
		20. Return lost property		T_6				S_6 U_{19}				T_6

CHAPTER 6

Business Process Reengineering

The third phase of TAD methodology deals with using the information collected in the entity, activity and task tables, developed in the previous two phases, to carry out business process reengineering (BPR) in the organization concerned. More precisely, this phase deals with identifying and implementing changes to improve or optimize the functioning of the system.

The third phase has two steps. The first step deals with analyzing and reengineering the entity, activity and task tables. The second step transforms the activity table into a process model of the system.

If the top management of the organization is not interested in carrying out BPR in its organization or the reader is interested only in information systems development, then skip to Section 6.2.

In recent years BPR has become a very important way of ensuring changes in an organization's structure and functioning to create a better, more competitive and successful enterprise. BPR is a method for identifying, defining and implementing change (Watson, 1994). This is achieved by defining an appropriate linkage among the strategic, business and operational levels of an organization.

At the enterprise level of the organization, top management focuses on the strategic plan and goals by developing the organizational values, vision, mission and objectives. The enterprise level is concerned with the implementation of the strategic plan and goals at business and operational levels. The enterprise level continually seeks breakthrough improvement in key business processes and is most appropriate for change (Watson, 1994).

At the business level of the organization, the strategic plan and goals are deployed within the context of its market environment and translated into business objectives and goals. This level tries to meet customers' needs and expectations and develop appropriate relationships with customers, suppliers and business partners.

T. Damij, *Tabular Application Development for Information Systems*

At the operational level of the organization, the business objectives and goals are translated into action plans. The principal work of the operational level is maintaining a reliable flow of both products and services to customers.

The result of the previous two phases is the development of the entity, activity and task tables. These tables generate a clear and visible linkage among the enterprise, business, and operational levels of the organization. The enterprise level is introduced by the analyses and outputs related to the strategic plan and goals required by the top management and defined in the entity table. The business level is represented by the analyses and outputs connected to business goals and objectives required by the business management and defined in the entity table. In addition, the business level is introduced by a set of the organization's business processes defined in the activity table. The operational level is introduced by a set of analyses linked to operational goals required by the operational management and defined in the entity table. Furthermore, the operational level is represented by a number of work processes and activities defined in the activity table. This level is also related to the task table, where each task or activity is described in detail.

6.1 Table Analysis and Reengineering

The entity, activity and task tables reflect the real world of the organization as a whole and also the reality of each entity in the framework of the organization. For this reason the analyst has to spend enough time to create, understand and analyze these tables. This fact is essential in TAD methodology, because carefully created entity, activity and task tables lead to the successful implementation of BPR.

Having these tables in mind gives a clear and visible picture of the functioning of the organization and offers the analyst a good opportunity of participating in creating a better and more effective organization. This goal is achieved by precise analysis of the entity, activity and task tables, suggesting changes and improvements and giving solutions for existing problems.

To do this, the analyst establishes a project team. This team consists of the analyst and representatives of the enterprise, business and operational levels. Usually we include influential and important members from the executive levels of the organization. The project team analyzes the existing strategic plan and goals, makes the needed changes and defines a new strategic plan and goals if necessary. The team members also update the entity table in accordance with the suggested new reality of the enterprise.

The team continues its work by considering the business processes defined in the activity table and creating a linkage between enterprise and business levels. This is achieved by making any necessary changes in the existing business processes or defining new business processes corresponding to the new strategic plan and goals.

The team then analyzes the work processes defined in the activity table and defines an appropriate linkage between business and operational levels. This linkage is achieved by making changes in the existing processes or by creating new work processes in accordance with changes made in the business processes.

The BPR team continues making the necessary changes in the activities listed in the activity table and also in the tasks defined in the task table. The aim of these changes is to optimize the time and resources needed to perform any task, activity or work process.

In addition to making changes in the strategic plan and goals, business processes and work processes, the project team has to analyze the obligations of the internal entities defined in the columns of the activity table and find a solution to the existing problems of each entity.

The project team's work leads to reengineering the entity, activity and task tables, which is essential in creating a more successful organization. This goal is achieved by

- defining new strategic goals, making the needed changes in the existing ones, or removing obsolete goals;
- creating an appropriate linkage between the strategic goals and business processes by defining new business processes, changing the existing ones, or removing unneeded business processes;
- creating a convenient linkage between the business processes and work processes by translating the changes made at the business level into the operational level of the organization;
- removing redundant activities or tasks;
- moving activities or tasks from one entity to another if these activities or tasks could be accomplished more easily or effectively;
- shortening the time needed to perform time-consuming activities or tasks;
- closing unneeded entities (departments, units);
- opening new entities (departments, units) as required.

TAD methodology makes the achievement of BPR possible using the information collected in the entity, activity and task tables. These tables enable the team to analyze the strategic goals, business processes, work processes, activities and tasks from various points of view. Furthermore, carefully analyzing the activity table leads to discovery of unneeded en-

tities or of the need to establish new entities. The team can also easily find redundant activities or tasks and obtain other useful information.

The task table contains valuable information about the time needed to perform any task. Therefore, the team can quickly find the most time-consuming tasks and warn the responsible people about them.

Usually table analysis and reengineering is an iterative process that needs to be repeated several times. Each further analysis may help in creating a better solution. BPR should be repeated from time to time to ensure the organization is as effective and competitive as possible. This fact is important, because having the strategic goals, business processes and functioning of the organization represented by tables enables us to perform an analysis of such tables quickly, without spending a lot of resources, and whenever necessary. This characteristic is very important in making TAD methodology useful, practical and effective not only in the field of Information Systems Development Methodologies but also in the field of Business Process Reengineering.

6.1.1. TOURIST ORGANIZATION

Corresponding to the preceding explanation, we carefully analyzed the entity, activity and task tables (Tables 4.1, 5.3 and 5.2) and created new activity and task tables. These are Table 6.1 and Table 6.2, respectively, which represent a better solution compared to the previous functioning of the Tourist Organization.

The new solution suggests a number of changes.

1. The first change suggested is to close the entity "Information Office." This office has a computer application system, which is used to store documents accepted from other entities and to print outputs and analyses requested by the Top Management and other entities. In the new solution every entity will be connected with the new information system and will be able to do its own work in an interactive manner. Therefore, there is no need for a special Information Office.
2. The second change suggested is to establish a new entity called "Housekeeping," because we found that nobody is in charge of controlling the work in the hotel rooms and other important facilities.
3. The third change suggested is to move many of the Top Management's tasks to other entities. This suggestion gives Top Management more time to lead the organization.
4. The fourth change concerns removing redundant activities; for example, the second activity, "Create an order," is removed from the entities Buffet and Restaurant. The new solution suggests unifying the accomplishment of this task to be performed only by the Purchasing Department.

Table 6.1. Reengineered activity table of Tourist Organization

Business Process	Work Process	Activity / Entity	1. Top Management	2. Purchasing Dept.	3. Marketing Dept.	4. Accounts	5. Reception	6. Buffet	7. Restaurant	8. Housekeeping	9. Supplier	10. Guest/ Customer
Sales	1. Purchasing	1. Make contracts with suppliers	T_2 P_1	T_9,S_2 P_1							S_9	
		2. Confirm supplier's contract	S_1 U_1	T_1,S_2 U_1,P_2							T_2	
		3. Create an order		S_2 U_2,P_3				T_2 P_3	P_2 P_3	T_2 P_3	T_2	
		4. Accept shipment		T_9 P_4				T_9 U_3	T_9 U_3	T_9 U_3	S_9	
		5. Accept supplier's invoice		T_4 P_5		T_9,S_4					S_9	
		6. Confirm invoice		S_2 U_3,U_4,U_5		T_2 P_6						
		7. Make supplier's payment				S_4 U_6					T_4	
	2. Marketing	8. Create advertising material	T_3 P_8		S_3 P_8							
		9. Confirm advertising material	S_1 U_8		T_1,S_3 U_8,P_9		T_3 P_9					T_3
		10. Accept demand			T_{10},T_5 P_{10}		T_{10},S_5					S_{10}
		11. Send offer	T_3 P_{11}		S_3 $U_9U_{10}P_{11}$							T_3
		12. Sign contracts with customers	S_1 U_{11}		T_1,S_3 U_{11}							T_3
Guest	3. Register	13. Accept reservation			T_5		T_{10},S_5 U_9,P_{13}					S_{10}
		14. Make hotel's occupation plan					S_5 U_{13}					
		15. Register guests					T_{10} U_{13},P_{15}					S_{10}
		16. Make plan of activities	T_5				S_5 U_{13},U_{15}					T_5
	4. Departure	17. Create customer's invoice				T_5 P_{17}	S_5 U_{15},P_{17}					T_5
		18. Accept customer's payment				T_5 U_{17}	T_{10},S_5 U_{17}					S_{10}
		19. Register lost property					S_5 U_{15},P_{19}					
		20. Return lost property					S_5 U_{19}					T_5

Table 6.2. Reengineered task table of Tourist Organization

Activity	Characteristic Task Code	Description	Time	Condition	Input/ Output
1. Make contracts with suppliers	$K_{1,2}$	Purchasing Dept. accepts contract from supp. and sends it to Mgmt. for confirmation			Contract
	$K_{1,1}$	Mgmt. accepts supp.'s contract from Purchasing			Contract
2. Confirm supplier's contract	$K_{2,1}$	Management confirms it and sends it back			Contract
	$K_{2,2}$	Purchasing accepts confirmed contract and sends it to supp.			Contract
3. Create an order	$K_{3,2}$	Purchasing creates order and sends it to supplier. A copy of it to Buffet, Restaurant or Housekeeping		Check if the supplier has a contract	Order
4. Accept shipment	$K_{4,2}$	Purchasing Department accepts supplier's shipment		Check the ordered products	Shipment
	$K_{4,6}$	Buffet accepts supplier's shipment		Check the ordered products	Shipment
	$K_{4,7}$	Restaurant accepts supplier's shipment		Check the ordered products	Shipment
	$K_{3,8}$	Housekeeping accepts supplier's shipment		Check the ordered products	Shipment
5. Accept supplier's invoice	$K_{5,4}$	Accounts accepts supplier's invoice and sends it to the Purchasing Dept. for confirmation		Check supplier's shipment	Invoice
6. Confirm invoice	$K_{6,2}$	Purchasing accepts invoice, confirms it and sends it to Accounts			Invoice

6.2 Process Model

This step may be considered to be optional because the activity table seems to be a more visible and transparent way of displaying and understanding the real world. Nevertheless, TAD methodology tries to define a very easy way to develop the process model, because many analysts are used to creating data-flow diagrams and other diagrams to understand the behavior of the system.

This step deals with creating the process model by describing a procedure that enables the analyst to create a data-flow diagram quickly and without much effort. To develop the process model of the system, we transform the activity table into the model. The whole process model is developed in accordance with the information collected in the activity table. In this manner, we use the following procedure:

We list all activities in the rows of the activity table. For every activity defined in row i, where i ranges from 1 to the number of

activities, we list all entities in the columns. For every entity defined in column j, where j ranges from 1 to the number of entities, we use the following roles:

- Each non-empty square(i,j) is transformed into an elementary process (circle) if entity(j) is an internal entity. This elementary process represents the task performed by entity(j) in the framework of activity(i). For this reason we write in it the indices i and j.
- Otherwise, each non-empty square(i,j) is transformed into a source (rectangle) if entity(j) is an external entity, and then we write the name of entity(j) in it.

After transforming the whole table, we connect the sources and processes horizontally and vertically:

- Horizontally we link each process or source indicated by S_j in square(i,j) by arrows with those processes and sources indicated by T_j in row i, where j is the index of the source entity.
- Vertically we link every process indicated by P_i in square(i,j) by arrows with those processes indicated by U_i in column j, where i is the index of the predecessor activity.

Using the described procedure, we transform the whole activity table into a single DFD (**data-flow diagram**). This is an elementary-level DFD, which represents the whole system. Each process is an elementary process and represents a determined task. Each group of horizontally connected elementary processes depicts an activity of the activity table.

To indicate each work process, we draw a border around the elementary processes and sources related to the work process considered. Therefore, inside each border a certain work process is defined at the elementary level.

To denote every business process, we also draw a border around those work processes that belong to the business process considered. Therefore, each border shows a determined business process at the elementary level.

The whole DFD represents the activity table. Every business process is indicated by a determined part of this DFD. Each work process is also introduced by a certain marked part of the DFD.

We may continue this task to create separate DFDs at different levels by grouping processes into suitable groups. These are the activity-level DFD, the work-process–level DFD, the business-process–level DFD and the system-level DFD (context diagram).

To create the activity-level DFD, we group elementary processes belonging to an activity into a single process. Each of the newly created

processes represents an activity in the activity table. For this reason, we write the name of the activity inside the process.

To define work-process–level DFDs, we group all the processes of every work process in the activity-level DFD into a single process. Each of the newly drawn processes introduces a determined work process. Therefore, we write the name of the work process inside the process.

We continue defining a business-process–level DFD (top-level DFD) by grouping all processes of each business process in the work-process–level DFD into one process. Each of the newly defined processes represents a certain business process. For this reason, we write the name of the business process inside the process.

All processes of the business-process–level DFD may be grouped into a single process, giving the context diagram, or system-level DFD. Therefore, the name of the system is written in it.

In addition to data-flow diagrams, without problems we can use the information collected in the activity and task tables to create other diagrams, which may be needed to increase the understanding of the system.

6.2.1. Tourist Organization

The process model of the Tourist Organization was developed using the content of Table 6.1. Figure 6.1 shows the process model. This is an elementary-level DFD, which includes two business-process–level parts.

The first business-process–level part represents the business process "Sales." This part consists of two work-process–level parts: "Purchasing" and "Marketing." The second business-process–level part introduces the business process "Guest." This part of the DFD also contains two work-process–level parts: "Register" and "Departure."

The first work-process–level part represents the work process "Purchasing." Figure 6.1 shows that this part of the DFD contains 17 elementary processes and one source. These processes are obtained by transforming the non-empty squares in the first seven rows of the activity table (Table 6.1) into elementary processes. The sources are obtained by transforming non-empty squares of the external entities.

The second work-process–level part shows the work process "Marketing." This part is created from the information collected in rows 8 to 12. It contains nine elementary processes and one source. The third work-process–level part introduces the work process "Register." This part is developed by transforming the information collected in rows 13 to 16. It has six elementary processes and one source. The last work-process–level part represents the work process "Departure." This part of the DFD is created using the information in the last four rows. It has six elementary processes and one source.

Furthermore, Figure 6.2 shows the activity-level DFD obtained by

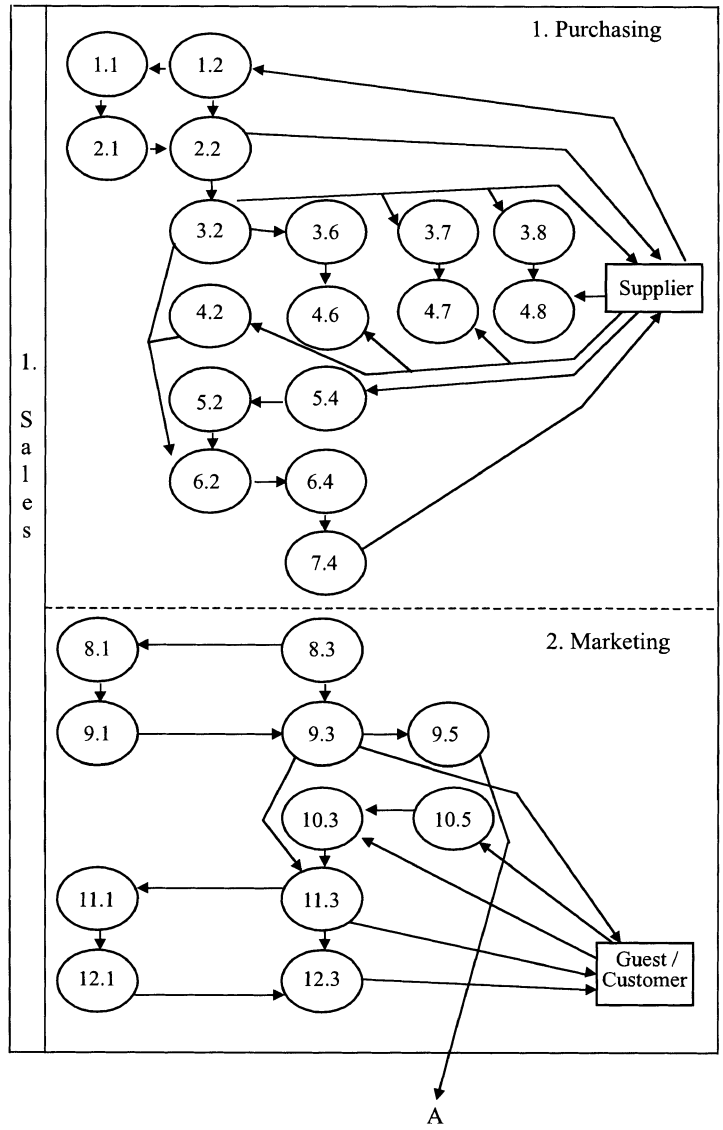


FIGURE 6.1. Process model of Tourist Organization

grouping elementary processes that belong to a certain activity into a single process. Each of these processes represents a determined activity. This DFD contains 20 activity-level processes.

Figure 6.3 shows the work-process–level DFD created by grouping the processes of each work process in the activity-level DFD into one process. Each process in this DFD introduces a certain work process. The

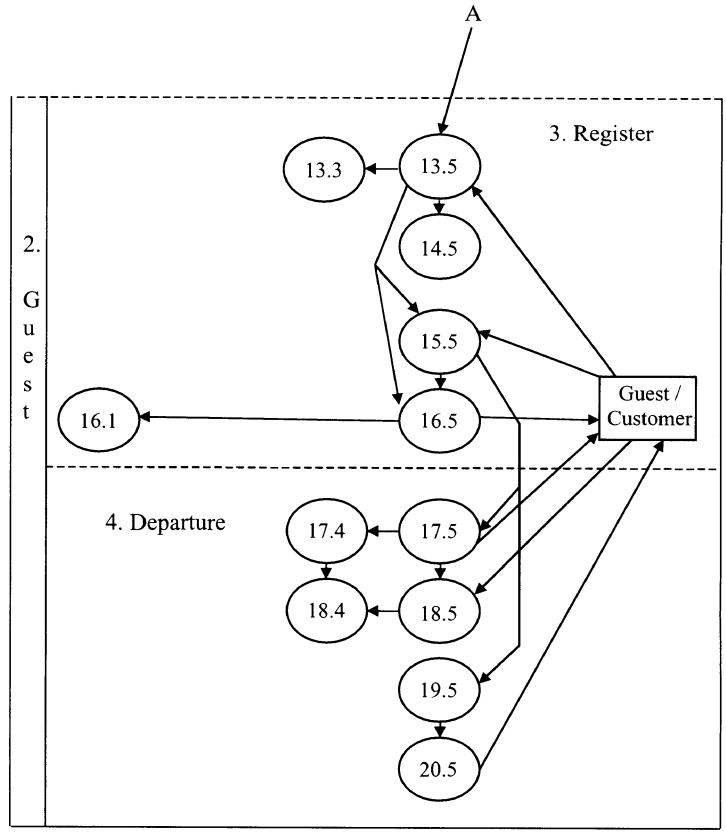


FIGURE 6.1. (Continued) Process model of Tourist Organization

newly created DFD has four work-process–level processes: "Purchasing," "Marketing," "Register" and "Departure."

Figure 6.4 shows the business-process–level DFD created by grouping the processes of each business process in the work-process–level DFD into a single process. Each process in the newly created DFD represents a determined business process. This DFD contains two business-process–level processes: "Sales" and "Guest."

Figure 6.5 introduces the context diagram of the Tourist Organization. This DFD has only one process, which represents the Tourist Organization.

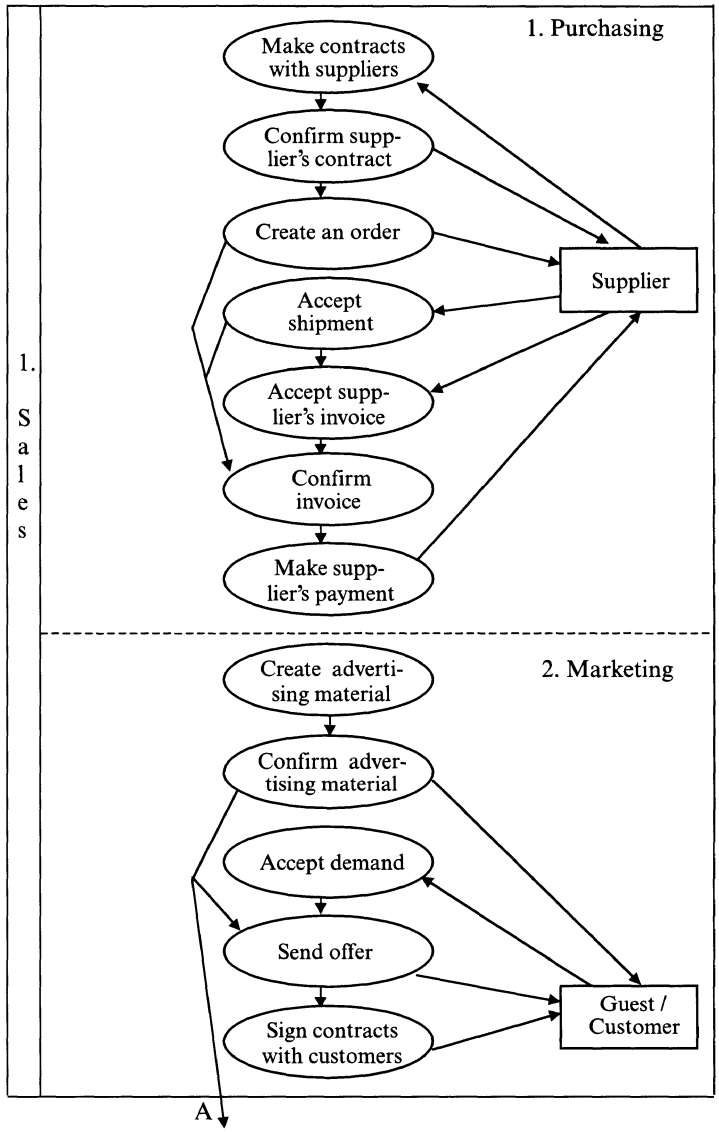


FIGURE 6.2. Activity-level DFD

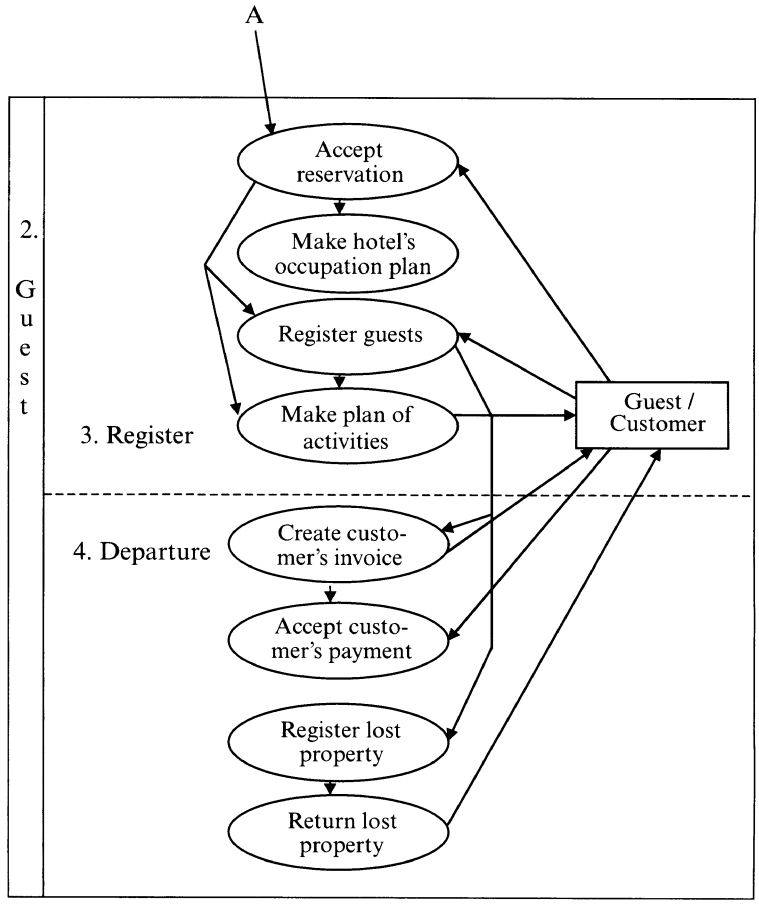


FIGURE 6.2. (Continued) Activity-level DFD

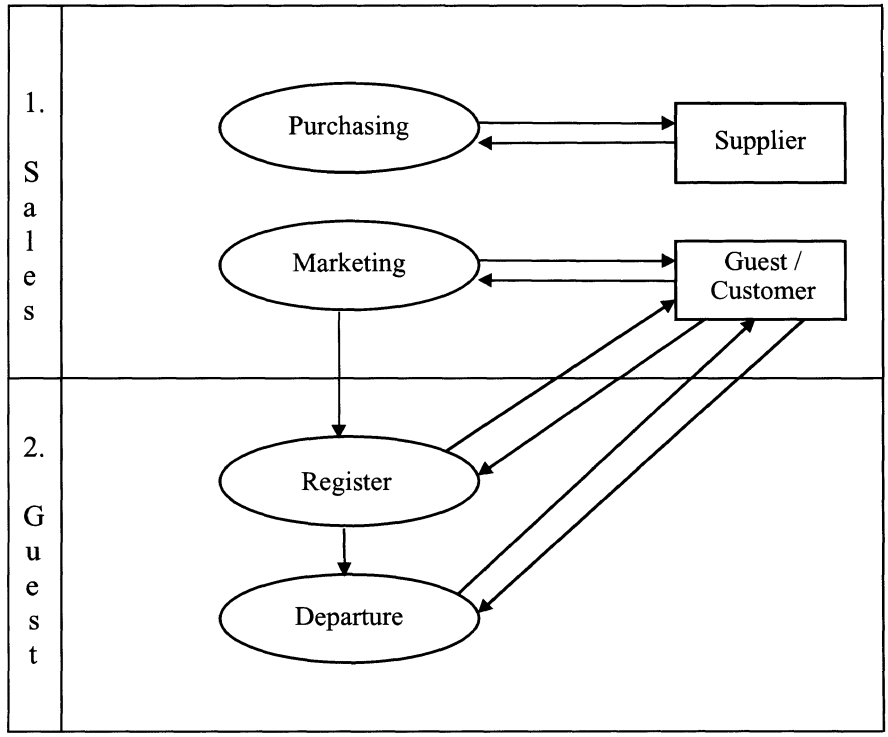


FIGURE 6.3. Work-process–level DFD

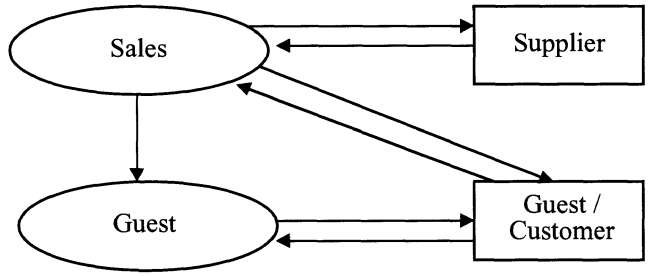


FIGURE 6.4. Business-process–level DFD

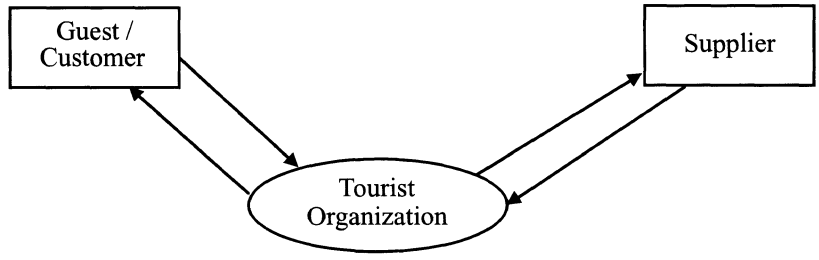


FIGURE 6.5. Context diagram

CHAPTER 7

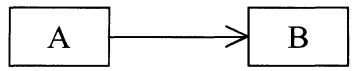
Object Model

The fourth phase of TAD methodology deals with the development of the object model of the system. This phase has six steps: The first five steps create the initial object model and the sixth completes the object model.

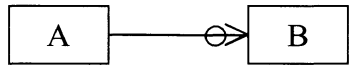
First let us reintroduce some important definitions about an object and the object class.

Corresponding to the definitions of "object" given in Section 2.1, we can say that an object is anything identified in the process of information systems development and recognized by its properties (attributes) and behavior (operations).

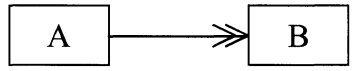
Furthermore, let us represent the notations used by TAD methodology to introduce the associations existing between object classes and between attributes.



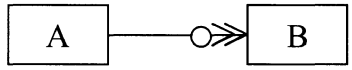
Every object instance of class A is associated with only one object instance of class B.



Every object instance of class A is associated with zero or one object instance of class B.



Every object instance of class A is associated with one or more object instances of class B.



Every object instance of class A is associated with zero, one or more object instances of class B.

T. Damij, *Tabular Application Development for Information Systems*

Before we introduce the above-mentioned steps of the current phase, it is essential to explain two important concepts: structures of semantic network modeling and functional dependence.

7.1 Semantic Network Modeling

In object modeling we use three structures of semantic network modeling to carry out three types of relationships that can exist between classes and attributes. These structures are Is-part-of, Is-associated-with and Isa (cf. Section 2.4).

The **Is-part-of** structure is used to describe that something can be created or constructed from a number of determined elements or components. Figure 7.1 shows an example of implementing the Is-part-of structure. Name Is-part-of Patient and Address Is-part-of Patient are used to describe the aggregation of attributes. Furthermore, Ordered-Item Is-part-of Order is an example used to describe the aggregation of classes.

The **Is-associated-with** structure is used to describe the relationships between classes. Such a relationship could be a one-to-one, one-to-many or many-to-many association. Figure 7.1 shows that Order Is-associated-with Doctor is used to describe the relationship between the classes Order and Doctor. The class Order Is-associated-with Patient is used to describe the relationship between the classes Order and Patient.

Sometimes it is useful to model an association as a class (Rumbaugh et al., 1991). Thus classes are associated via an association class. Such an association class is OrderedItem, which creates a relationship between classes Order and Item; see Figure 7.1.

The **Isa** structure is used to describe inheritance between classes to share a common structure.

Figure 7.2 shows an example of **generalization:** Payment is a superclass, which was created to generalize common attributes and operations of the classes PatPayment and InsPayment, where PatPayment means that the patient pays for the treatment and InsPayment means that an insurance company pays for the treatment.

Top-down analysis, which is called **specialization,** means creating specialized subclasses from the existing classes.

Figure 7.3 introduces an example of specialization, where Male and Female are specialized subclasses created from the class Patient.

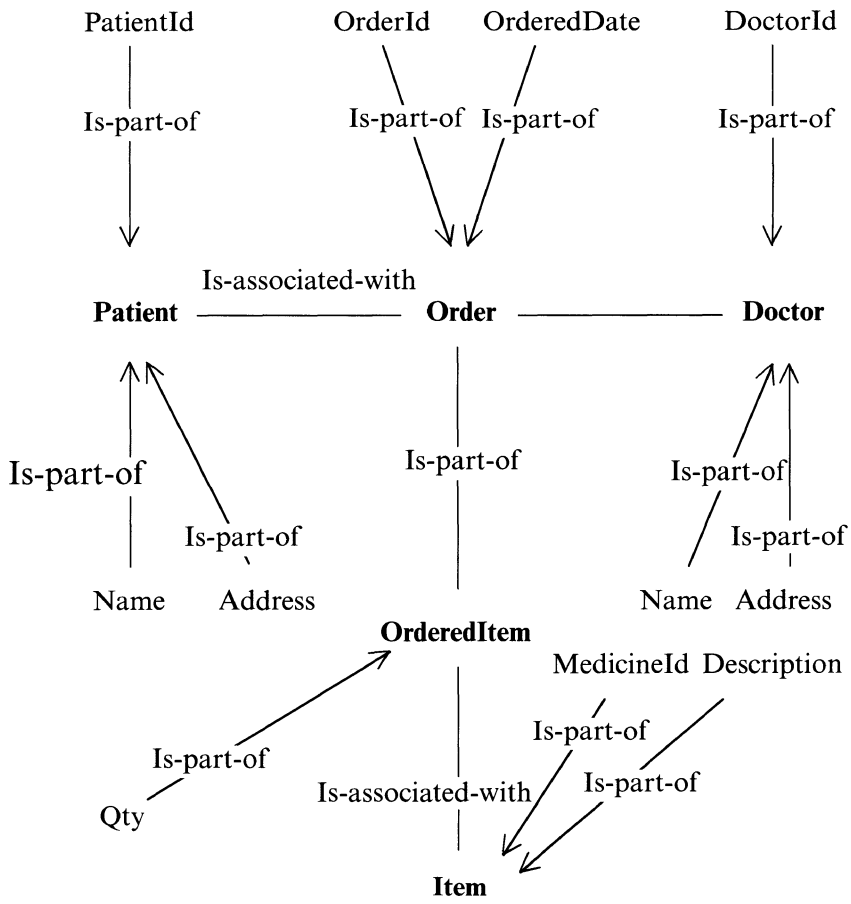


FIGURE 7.1. Is-part-of and Is-associated-with structures

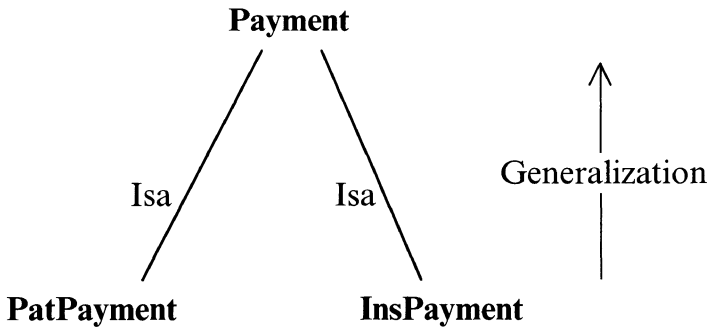


FIGURE 7.2. An example of generalization

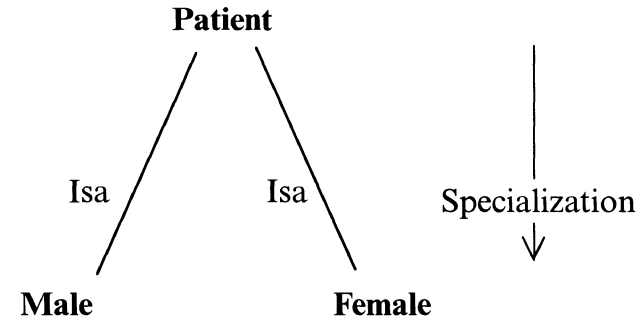


FIGURE 7.3. An example of specialization.

7.2 Functional Dependence

The aim of introducing functional dependence is to analyze in detail the relationships that exist between attributes. The following definition considers the relationships that may exist between attributes of a determined relation:

> An attribute B of a relation is functionally dependent on the attribute A of it if, at every instant of time, each A-value is associated with no more than one B-value (Vetter, Maddison, 1981).

Concerning the relationships that may exist between attributes contained by a certain user's document we can say:

> An attribute B of a certain document is functionally dependent on attribute A (attributes $A_1, \ldots, A_n$) if every value of attribute A (attributes $A_1, \ldots, A_n$) is associated with one and only one value of attribute B.

The notation used to indicate that attribute B is functionally dependent on attribute A is

$$A \rightarrow B$$

For example, PatientName is functionally dependent on PatientNumber because each value of attribute PatientNumber determines or identifies one value of attribute PatientName. CustomerName is functionally dependent on CustomerNumber because each value of attribute CustomerNumber relates to one and only one value of attribute CustomerName. Furthermore, attribute Qty is functionally dependent

on a combination of the attributes OrderNumber and ItemNumber because every one combination-value of OrderNumber and ItemNumber determines one value of attribute Qty. These functional dependencies are written as follows:

PatientNumber → PatientName
CustomerNumber → CustomerName
OrderNumber, ItemNumber → Qty

7.3 Object Model Development

The process of object model development is founded on information gained in the interviews with the users, especially by using the results of the previous three phases, which means carefully analyzing the entity, activity and task tables.

As mentioned, the fourth phase of TAD methodology has six steps. The first five steps deal with developing the initial object model, and the last step completes the object model and uses inheritance to define hierarchies between classes.

Functional dependence is used to analyze users' documents, which are obtained through interviews with the users. In the process of analyzing any document we usually identify attributes that belong to different object classes. Analyzing the functional dependencies that exist among attributes enables us to create a linkage between functional dependence and the structures of semantic network modeling. This leads to development of a procedure that could be used to transform the identified functional dependencies into the mentioned structures.

To achieve this, let us first define the term "object identity."

> Each object has an identity, which represents it. Object identity (identity attribute) is the property of an object, which uniquely distinguishes it from other objects (cf. Section 2.1).

First we try to develop the initial object model using the information collected in the task, activity and entity tables to identify object classes, their attributes and their associations. The inputs to the initial model development are

- the task table,
- the entity table,
- the activity table,
- expert knowledge.

We start by analyzing the task table. For this purpose, we analyze each column of the task table, especially the inputs and outputs (documents)

defined in the Input/Output column. Each of these documents has to be analyzed very carefully.

First Step

The first step of the current phase identifies the functional dependencies that exist between identity attributes and other attributes of every analyzed document.

The aim of the analysis of each document using the mentioned procedure is to identify the object classes that are contained in the document, their attributes and their associations. These object classes represent the foundation of the final object model, because the content of the documents analyzed represents important information about the system as a whole and its objects and their attributes and associations.

As an example, let us analyze two simplified documents, ORDER and CUSTOMER; see Figure 7.4.

It is not difficult to discover that the attributes contained by the ORDER document belong to three object classes: ORDER, CUSTOMER and ITEM. These classes are represented by three identity attributes: Order#, Customer# and Item#.

To analyze the considered document, we first identify the identity attributes this document contains and then try to find out which of the other attributes are functionally dependent on each of the identified identity attributes.

Concerning the ORDER document, we use the following analysis to identify the identity attributes and to analyze the relationships existing among the identity attributes and other attributes of the document.

Figure 7.5 shows that the attribute Date is functionally dependent on attribute Order# because each Order# value determines one Date value.

The attributes First Name, Surname, Address and City are functionally dependent on the attribute Customer# because every Customer# value identifies one value of each of the attributes First Name, Surname, Address and City.

Furthermore, the attributes Description and Price are functionally dependent on the attribute Item# for the same reason. We also find that the attribute Qty is functionally dependent only on a combination of attributes (Order# and Item#) because every attribute Qty value is determined only by a combination of Order# and Item# values.

Second Step

The second step deals with analyzing the relationships existing among the identity attributes themselves. Concerning document ORDER, we use the analysis shown in Figure 7.6 to define the associations existing among the identity attributes.

ORDER			
Order#:	O123	Date:	01/30/2000
Customer#:	C678		
First Name:	Ann	Surname:	Kim
Address:	3812 Lonyo	City:	Detroit

Item#	Description	Price	Qty
I001	Chair	$100	4
I002	Table	$250	2
I003	Wardrobe	$300	1

CUSTOMER	
Customer:	C678
First Name:	Ann
Surname:	Kim
Address:	3812 Lonyo
City:	Detroit
Phone:	555-1231
Status:	Good

FIGURE 7.4. Documents ORDER and CUSTOMER

Figure 7.6 shows that the identity attributes Customer# and Order# are associated by a one-to-many association because each Order# value determines only one value of the attribute Customer# and every Customer# value could be related to one or more Order# values.

Furthermore, we also find that the attributes Order# and Item# are associated by a many-to-many association because one Order# value could be related to one or more Item# values and every Item# value could be connected to none, one or more Order# values. Figure 7.6 shows that such an association is replaced by two one-to-many associations. These are associations between attributes Order# – Order#, Item# and Order#, Item# – Item#.

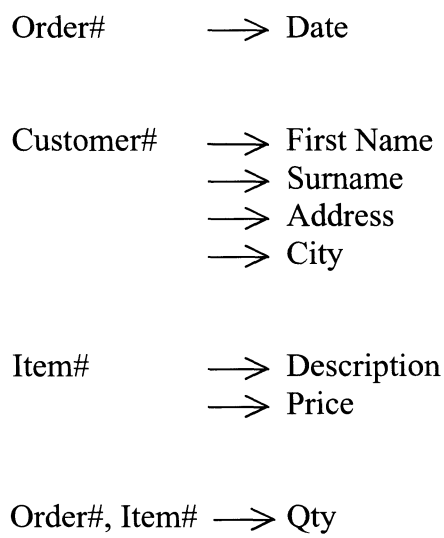


FIGURE 7.5. Functional dependencies among attributes

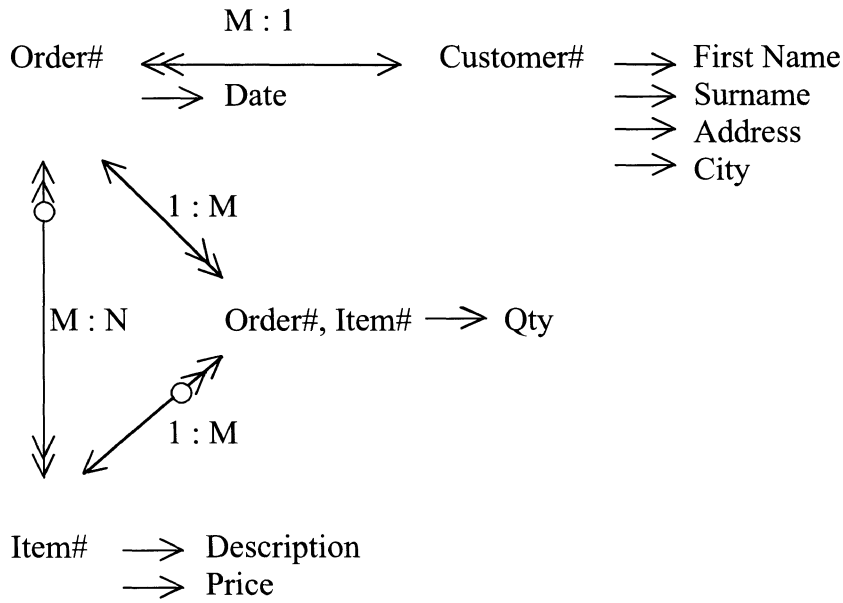


FIGURE 7.6. Associations among identity attributes

After completing analysis of the first document, we continue by analyzing the second document by repeating the first two steps.

Analyzing document CUSTOMER, we find that all its attributes belong to one object class: CUSTOMER. This object class is represented by its identity attribute Customer#. This means that all other attributes of this document are functionally dependent on the identity attribute, as the analysis in Figure 7.7 shows.

Third Step

The third step of the procedure starts when we complete analysis of all the documents using the previous two steps. The third step deals with integrating the analyses of all documents. We group functional dependencies of all analyses that are related to the same identity attributes.

Figure 7.8 shows the integration of analyses presented by Figures 7.6 and 7.7. This integration is done by extending the attributes related to

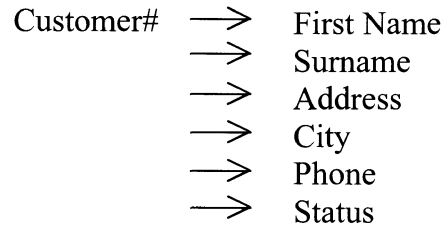


FIGURE 7.7. Functional dependencies among attributes

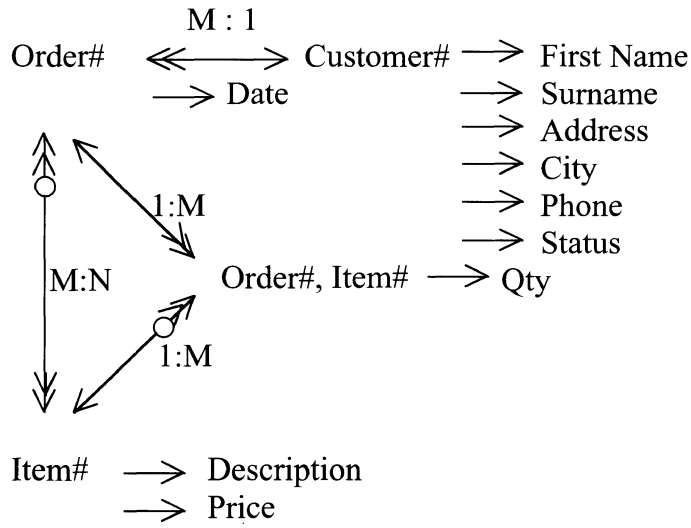


FIGURE 7.8. Integration of analyses

Customer# in Figure 7.6 by those attributes that are contained only in Figure 7.7.

Fourth Step

The fourth step replaces the identity attributes by the names of their object classes, connects all attributes of an object class to its class name and transforms the relationships identified into suitable structures of semantic network modeling.

Such replacement of identity attributes by their classes and transformation of functional dependencies and associations into the mentioned structures are shown in Figure 7.9.

From Figure 7.9, we may conclude the following:

1. every functional dependence on any identity attribute is transformed into an Is-part-of structure;
2. every identity attribute is replaced by its class name;
3. an identity attribute is also related to the class name by the Is-part-of structure;

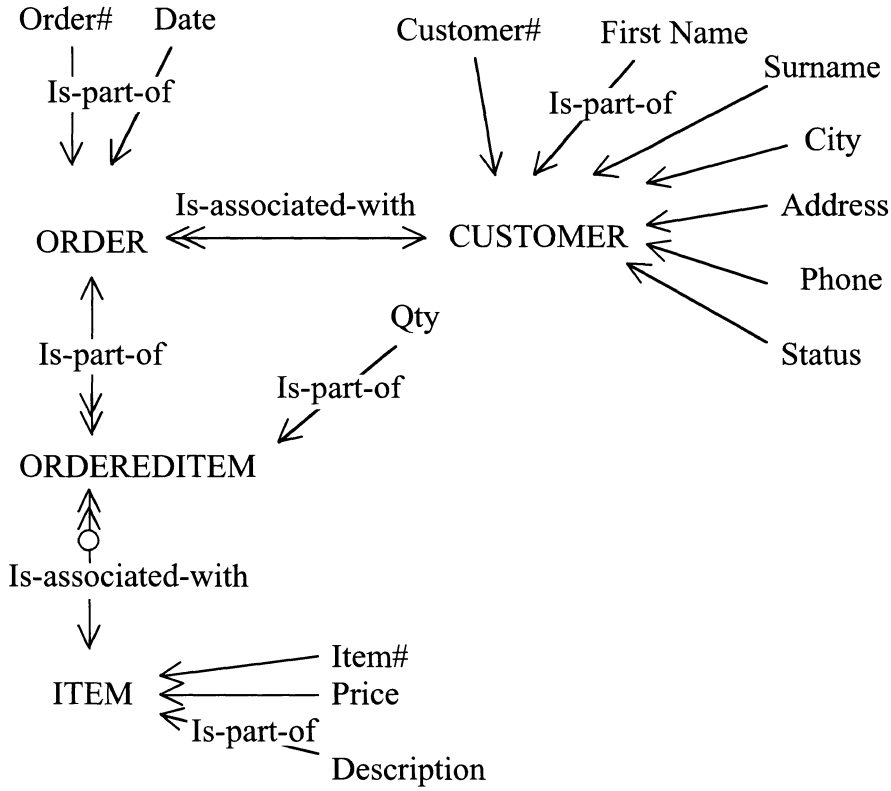


FIGURE 7.9. Defining classes and structures

4. a many-to-many association (for example, the association between ORDER and ITEM) is transformed into two one-to-many associations;
5. a one-to-many association could be transformed into an Is-associated-with (for example, the association between ORDER and CUSTOMER or between ORDEREDITEM and ITEM) or an Is-part-of structure (for example, the association between ORDER and ORDEREDITEM).

Fifth Step

The fifth step transforms the result of the above analysis (Figure 7.9) into an initial object model of the system, as shown in Figure 7.10.

Sixth Step

In the sixth, and final, step, the initial object model developed is extended and completed using other information gained from the task, activity and entity tables and also from expert knowledge. If any new object class, attribute or association is identified, the existing object model is extended by it. In addition, we try to complete the object model by implementing the inheritance among the classes.

For this purpose, we continue by considering the other columns of the task table; in particular, we carefully analyze each description defined in the Description column of the table and try to find out which

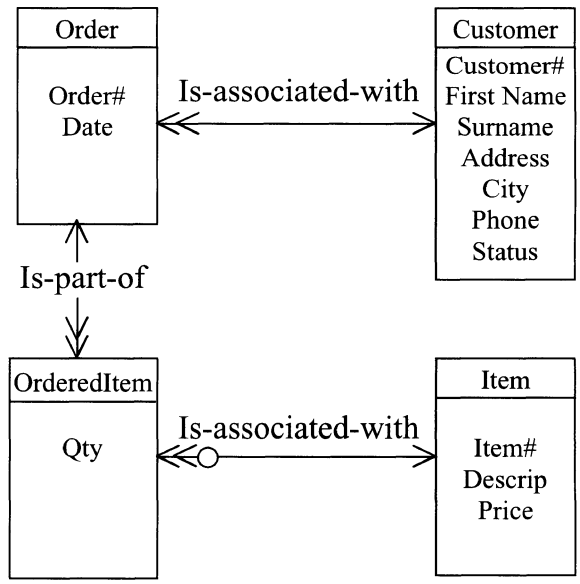


FIGURE 7.10. Initial object model

object classes, attributes or associations are connected with it. If any new object class, attribute or association is identified, the existing group of object classes is extended by it.

This work is continued by analyzing the entity table, particularly the outputs and reports defined in the rows of the table. Any newly discovered object class, attribute or association is added to the existing ones.

We continue the process by analyzing the activity table. We analyze the entities defined in the columns and also the business processes, work processes and activities listed in the rows of the table.

Finally, the analyst may add from his or her knowledge any other object class, attribute or association if it is necessary or useful to the application.

The result of this work is the initial object model of the system, which needs further analysis in order to transform it into the final object model. We complete the initial object model by using inheritance. For this purpose we try to organize the classes of the initial object model using inheritance (the Isa structure) to define hierarchies among classes and to share a common structure. Inheritance can be added in two directions: by generalizing common aspects of existing classes into a superclass (bottom-up) or by refining existing classes into specialized subclasses (top-down) (Rumbaugh et al., 1991).

7.3.1. TOURIST ORGANIZATION

In accordance with the previously described steps of the fourth phase, we used the procedure introduced in Section 7.3 to analyze the documents indicated in the Input/Output column of the task table (Table 6.2). The result of this analysis is the initial object model of the Tourist Organization, as shown in Figure 7.11.

After completing the analysis of all documents, and in accordance with the sixth step, we continued by analyzing the content of the Description column of the task table. We also carefully analyzed the entity and the activity tables. Any newly identified object class, attribute or association was added to the existing ones.

The initial object model shown here is simplified in comparison to the real initial object model and to the information given in the entity, activity and task tables.

In addition, we tried to implement the inheritance among the classes of the initial object model. Analyzing the initial model (Figure 7.11), we found two types of invoices with many similarities: SupInvoice and GueInvoice, where the first is connected with the suppliers and the second is related to the guests. In addition, we found two types of guests: OrdGuest and ConGuest; the first is an ordinary guest and the second is a guest connected with a contract. For this reason we create two superclasses, Invoice and Guest.

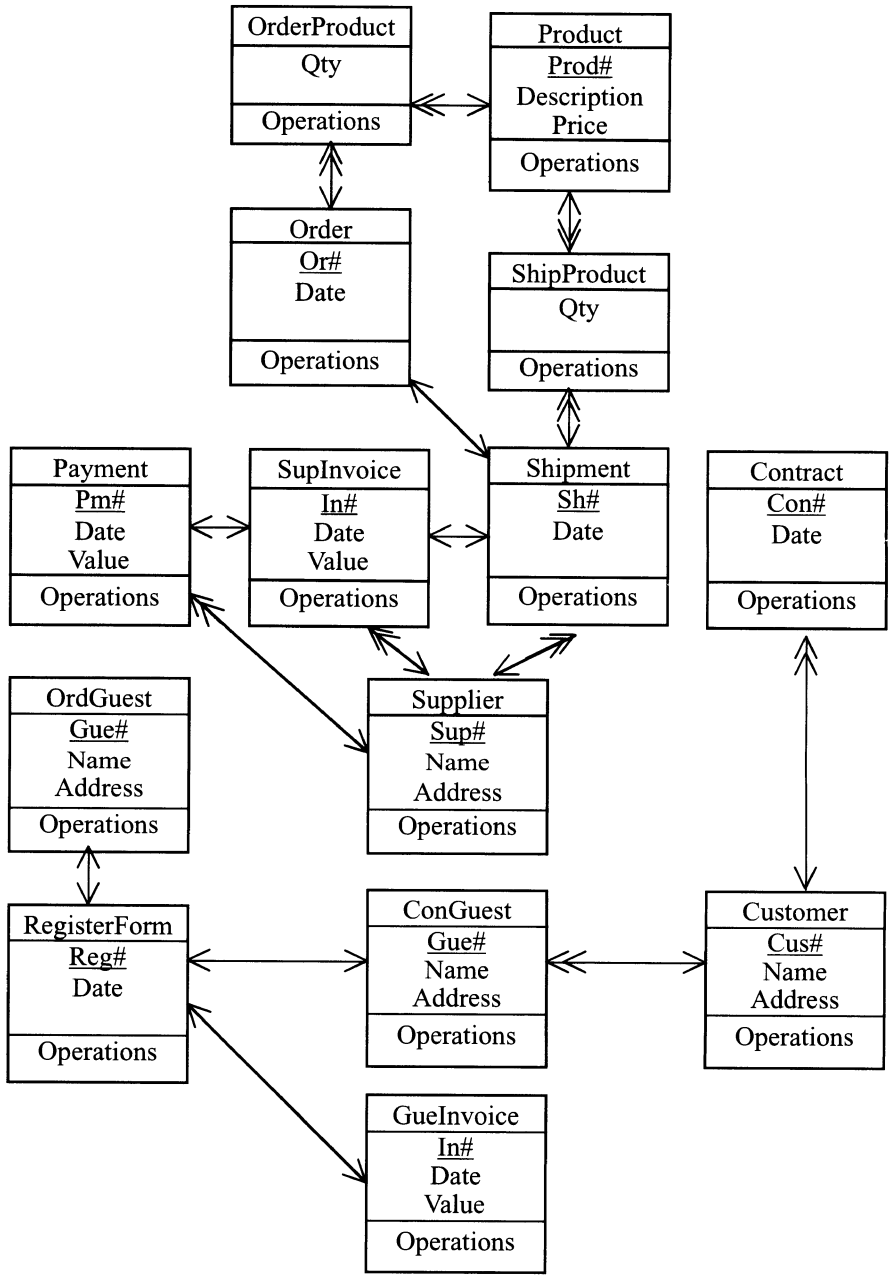


FIGURE 7.11. Initial object model of Tourist Organization

Furthermore, when analyzing the class Contract, we found that from this class we can create two specialized subclasses: SupContract and CusContract, where the first type of contract is linked to suppliers and the second is linked to customers.

Figure 7.12 shows the final object model of the Tourist Organization. This model is very simplified in comparison to the real one.

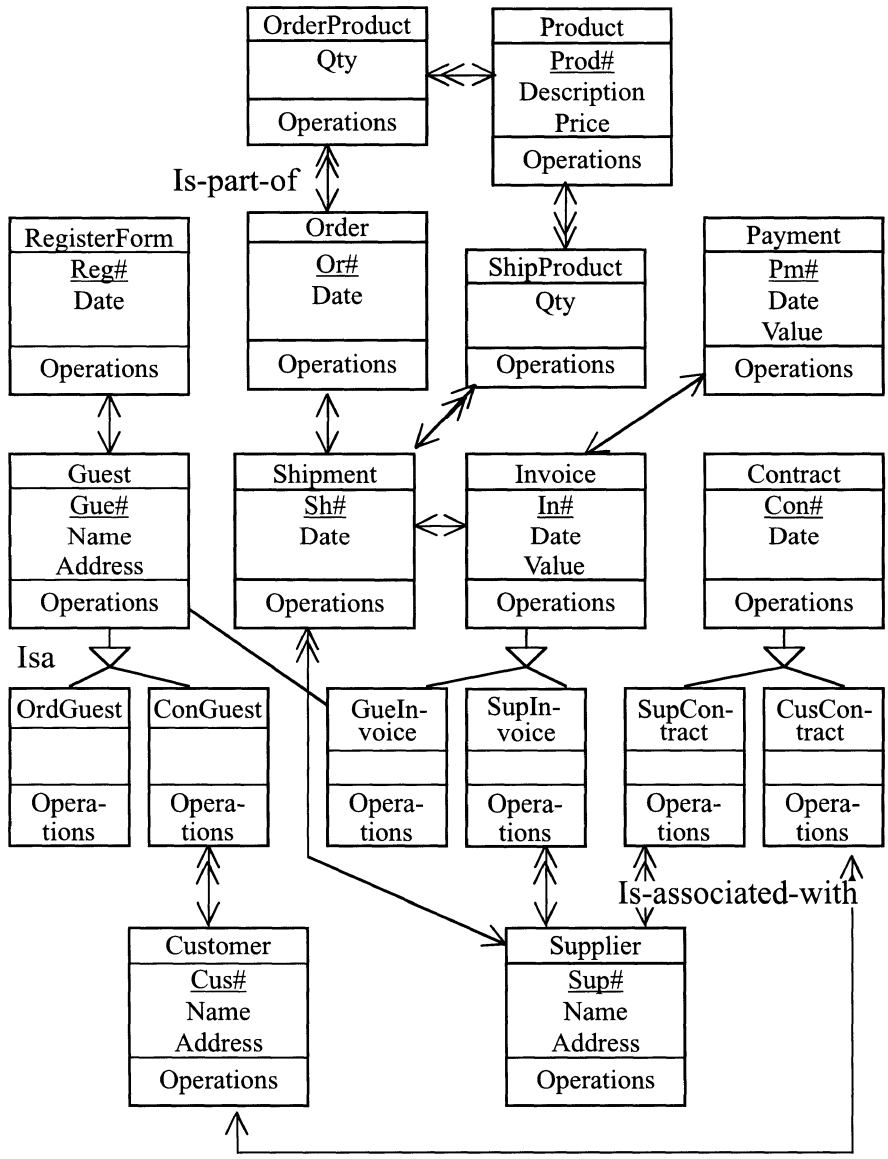


FIGURE 7.12. Final object model of Tourist Organization

CHAPTER 8

Design

The first four phases of TAD Methodology represent the analysis of the real world of the system by defining its functionality and static structure. The fifth phase of the methodology deals with designing the system and preparing it for implementation. This phase has two steps. The first step specifies the operations of the object model, and the second step develops the application model of the system.

8.1 Operations

The first step of the fifth phase deals with identifying the operations of classes of the object model and writing algorithms to define in detail the operations specified.

Identification of the operations requires creating a linkage among the tasks defined in the task table and the classes of the object model. The best way to do this linkage is by analyzing the tasks described in the task table one by one and trying to link each task with the classes involved in performing it. If any connection exists between the task considered and any class of the object model, then an operation is defined in the framework of the class by writing the name of the operation inside the class definition in the object model.

Usually we transform each task into an operation. In this case, the task is simple and deals with one event. For example, a certain class receives an input, creates a response, sends an output, and so on. Sometimes we identify more than one operation by analyzing the task description in the task table and the task's square in the activity table. In such situations, the task considered describes two or more successive events. For example, a determined class receives an input and forwards it to another class.

To perform this work we carefully analyze the task description defined in the Description column of the task table and we simultaneously analyze the content of the task square where the task is defined in the activity table.

T. Damij, *Tabular Application Development for Information Systems*

For every task defined in row k of the task table, where k ranges from 1 to the number of tasks, we try to analyze the description of task(k) in the task table and also square(i,j) in the activity table, where task(k) is defined. Indices i and j are indicated in the task code registered in the Task Code column and row k of the task table.

This analysis leads us to find out which classes of the object model are involved in performing task(k). This means that one or more operations have to be defined in the framework of each of the classes related to task(k).

To register the linkage created among each task, the operations connected with it, and the classes to which they belong, we develop a new table called the **operation table.** The operation table consists of three columns. The first column lists the tasks as they are defined in the task table. The second column represents the operations identified by analyzing the tasks listed in the first column. In the third column we register the classes that contain the listed operations.

Each operation occupies one row in the operation table. Any task may be connected with one or more operations. For this reason, each task occupies one or more rows of the operation table, depending on the number of operations related to the task considered; see Table 8.1.

To create the operation table, we use the following procedure.

Table 8.1. Operation table of Tourist Organization

Task code	Operation	Class
$K_{1,2}$	AccContract	Contract SupContract
$K_{1,1}$	AccContract	Contract SupContract
$K_{2,1}$	ConfContract	Contract SupContract
$K_{3,2}$	CreateOrder	Order OrderProduct
$K_{4,2}$	AccShipment	Shipment ShipProduct
$K_{4,6}$	AccShipment	Shipment ShipProduct
$K_{4,7}$	AccShipment	Shipment ShipProduct
$K_{4,8}$	AccShipment	Shipment ShipProduct
$K_{5,4}$	AcceptInvoice	Invoice SupInvoice
$K_{6,2}$	ConfInvoice	Invoice SupInvoice

1. Analyze the tasks defined in the task table one by one.
2. Identify every operation connected to the task considered by analyzing the task description in the task table and the content of the task square(i,j) in the activity table, where i and j are indicated in the Task Code column of the task table.
3. Connect every operation identified in step 2 with a determined class (classes) in the object model and write the name of the operation into the class definition.
4. Register the name of every operation and the name of the class in the framework which is defined in the Operation and Class columns of the operation table.

After completing the analysis of a determined task, identifying the operations connected with it, and registering these operations in the operation table and also into the classes of the object model, we then continue our work by defining each operation in detail.

We write an algorithm for each of the specified operations. The inputs needed to write such an algorithm are

- the task table, where every task and all its characteristics are described in detail,
- the object model and
- the operation table.

An operation algorithm has to be carefully written corresponding to the information collected in the task row of the task table. Such an algorithm includes the following characteristics:

- it is written in accordance with the task description given in the Description column of the task table;
- it implements the conditions related to the operation considered that are defined in the Condition column of the task table;
- it defines the inputs and outputs of the operation according to the content of the Input/Output column in the task table.

This step is essential, because it creates a very important linkage between the functioning of the system represented by the activity and task tables on one hand and the static structure of the system represented by the object model on the other. This step transforms the system functioning into operations of classes of the object model.

8.1.1. TOURIST ORGANIZATION

Using the above-described procedure, we define a linkage between the functioning of the system represented by the task and activity tables and

the static structure of the system represented by the object model. For this purpose we developed a new table—the **operation table**—and registered the operations in the Operation column of the table. These operations are identified as a result of carefully analyzing the task and activity tables.

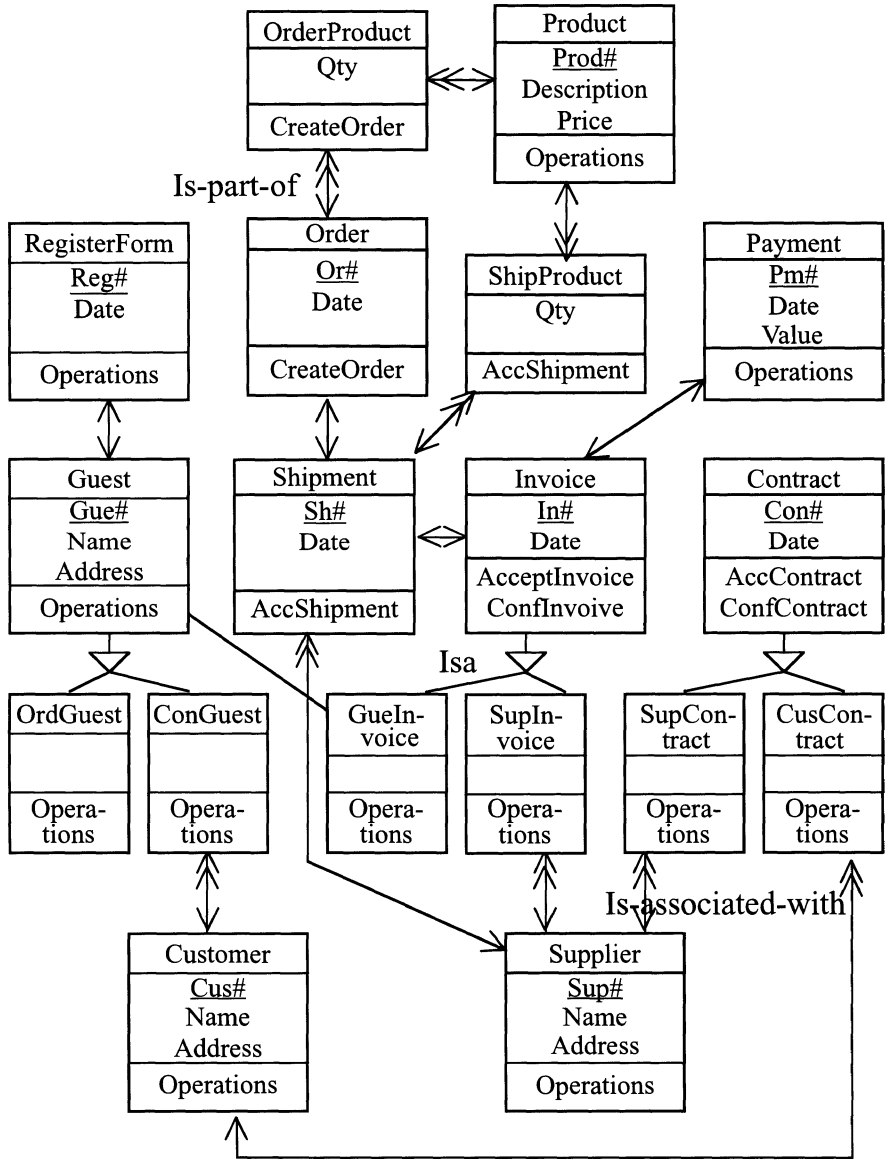


FIGURE 8.1. Object model of Tourist Organization

Table 8.1 shows the operation table of the Tourist Organization. In the second column of the table we listed six operations extracted from the tasks defined in the task table.

To relate every operation to a certain class, we listed in the third column the names of classes to which the operations belong.

Figure 8.1 shows the object model extended by the names of the operations defined in the operation table (Table 8.1).

Corresponding to the first step of the fifth phase, we continue with writing algorithms to define in detail the operations identified. To do so the analyst usually uses the preferred object-oriented programming languages such as Java, C++, Pascal or other languages and techniques to write a detailed algorithm for each operation defined in the operation table.

8.2 Application Model

The second step of the fifth phase deals with designing the system. The system's design model developed in this step is called the **application model** in TAD methodology. This model is then implemented in the next phase.

Development of the application model is derived from the information collected in the entity and activity tables. For this reason, the application model consists of several parts corresponding to the content of the two mentioned tables.

The entity table usually consists of two parts. The first part defines the reports required by the management at different levels. The second part defines the decision support problems. The first part is always present; the second part may be absent if no decision support problems were identified.

Therefore, the first part (if the entity table does not include any decision support problem) or the first two parts of the application model could be created by transferring information from the contents of the entity table to the relevant parts of the application model.

As stated in Chapter 5, the activity table represents the functioning of the system. This table consists of one or more business processes. To connect the development of the other parts of the application model with the content of the activity table, we copy information collected about each business process to the relevant part of the new application model.

First we develop those parts of the application model that are derived from the entity table. After that we continue by creating other parts of the application model that are related to the information stored in the activity table.

We start by developing the first two parts of the application model in accordance with the information registered in the entity table. To develop these two parts, we simply list the analyses defined in the rows of the entity table and create the parts corresponding to the content of the rows.

The first part of the application model is named "Reports" after the name of the first part of the entity table. The content of this part of the application model is derived from the entity table by transferring to it the names of all reports defined in the first part of the table.

This part of the application model is especially important for management because it deals with various analyses that are essential to management's decision making. These analyses provide the managers at different levels with all the necessary information about their strategic, business and operational plans and objectives. Therefore, the aim of these analyses is to help management optimize the functioning of the organization.

The second part of the application model is "Decision Support." The content of this part is derived from the second part of the entity table. Thus it includes those decision support problems defined in the entity table.

After completing the first two parts, we continue by developing the other parts of the application model in accordance with the information stored in the activity table and considering the internal entities only. These parts are derived from the first column of the activity table. Each of them represents a particular business process. Every business process may have one or more work processes corresponding to the content of the second column of the activity table. Every work process includes a number of activities in accordance with the content of the third column of the activity table.

To develop these parts of the application model, we create a new part of the application model for each business process defined in the first column of the activity table.

For every business-process–level part in the application model, we transfer to it all its work processes from the second column of the activity table. Furthermore, for each work-process–level part defined in the application model, we transfer to it all its activities from the third column of the activity table.

Thus we can conclude that the application model gives a clear picture of the requirements and decision support problems defined in the entity table. In addition, it includes all business processes, work processes and activities defined in the activity table. In other words, it represents the contents of the entity and activity tables.

This work is continued by analyzing each part of the developed application model to define every interface necessary to enable the users to interact with the system.

8.2.1. TOURIST ORGANIZATION

Figure 8.2 shows the application model of the Tourist Organization. This model consists of four parts.

We started by developing the first two parts; these are "Reports" and "Decision Support." The first part corresponds to the content of the

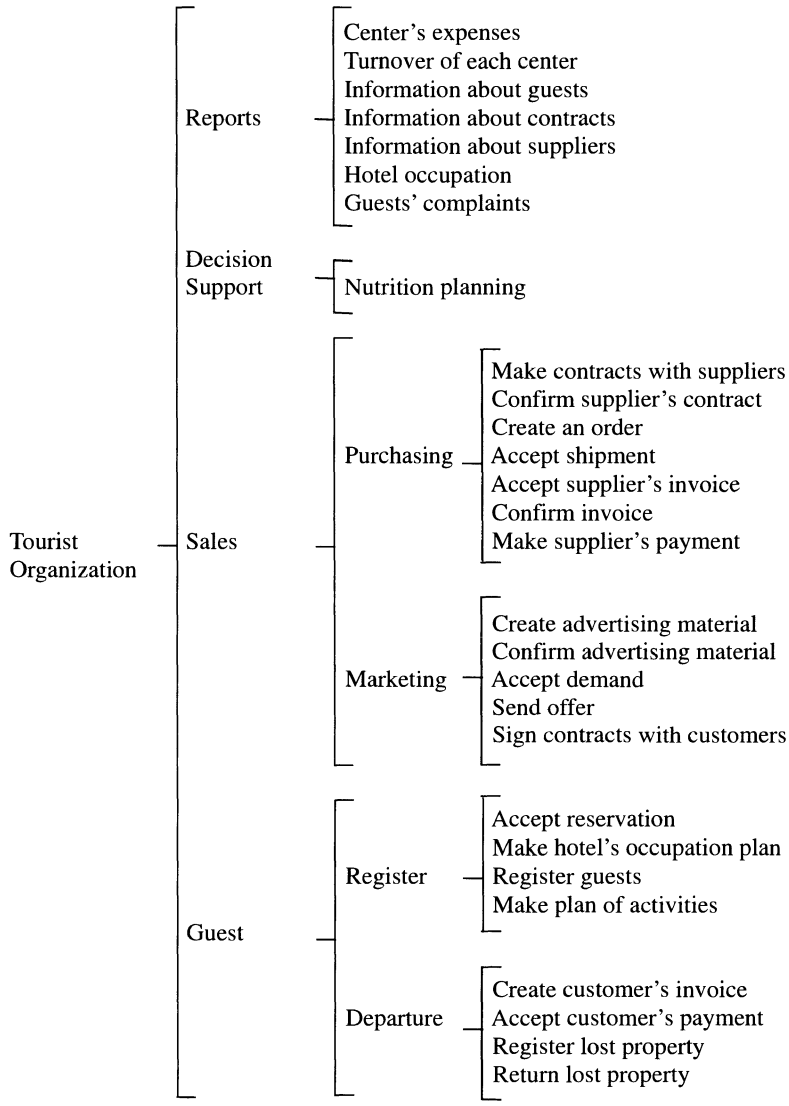


FIGURE 8.2. Application model of the Tourist Organization

entity table and contains 7 important analyses required by the management. These analyses deal with creating information about expenses, turnover, guests, contracts, suppliers, hotel occupancy and guest complaints. The second part introduces a decision support problem, nutrition planning.

Next we continued with creating the third and fourth parts of the application model, which were created corresponding to the content of the reengineered activity table (Table 6.1). These two parts represent the two business processes "Sales" and "Guest."

The first business process, "Sales," consists of two work processes: "Purchasing" and "Marketing." The work process "Purchasing" contains seven activities, which deal with defining the relationships between the organization and different suppliers. The second work process, "Marketing," includes five activities and deals with getting as many customers as possible.

The second business process, "Guest," has two work processes, "Register" and "Departure." The first work process, "Register," includes four activities and deals with accepting reservations and registering guests. "Departure" contains four activities, which accomplish everything related to the guest's departure.

PART III

APPLICATIONS

The third part consists of four chapters. These chapters discuss four other applications in addition to the first application—the tourist organization that was introduced in the second part.

Chapter 10 deals with an application that implements TAD methodology to develop an information system in a gastroenterological internal clinic. Chapter 11 introduces an application that discusses the use of TAD methodology for developing an information system in a clinic for small animals. Chapter 12 shows the development of an application of payment movement in a bank. Chapter 13 represents the last application contained in the book and considers the development of an information system in a student office.

All five applications are extracted from real examples and represented in a reduced manner in the book.

CHAPTER 9

Implementation

The sixth phase of TAD methodology deals with the implementation of the system developed in the previous phases. The inputs to the implementation phase are

- the object model,
- the application model,
- the algorithms related to both models.

The result of the fifth phase is a set of algorithms that precisely define the application model and the operations of all classes of the object model. These algorithms are translated into program code in the implementation phase.

The analyst usually starts the implementation phase by choosing a convenient database management system to implement the object model of the system. After that he or she translates the algorithms related to the operations of the object model, which were written in the design phase, into methods. This work is continued by translating the algorithms that define the application model into program codes.

In addition, in the current phase the analyst may implement user's data access. The analyst finds out which user or users have the right to access a determined action defined in the framework of the application model.

As discussed in the previous phase, the application model consists of several parts. The first two parts are derived from the entity table and the other parts are developed from the activity table.

Concerning the first and second parts ("Reports" and "Decision Support") of the application model, we can find information about a user's data access in the entity table. An entity(j) defined in column j of the entity table is allowed to access only those analyses contained by the first and second parts of the application model that are indicated by an asterisk in column j, where j ranges from 1 to the number of entities of the entity table.

T. Damij, *Tabular Application Development for Information Systems*

Concerning the other parts of the application model, the analyst can find information about a user's data access in the activity table. An entity(j) in the activity table is allowed to access only those activities or tasks specified in column j, where j ranges from 1 to the number of internal entities of the activity table.

Because this is not a book about programming languages, we will not discuss any particular programming language. Thus we prefer to leave it to the implementation team to choose its favored programming language and database.

CHAPTER 10

Internal Clinic

This chapter covers the second application (the first application was discussed throughout Chapters 4 to 9). The chapter illustrates the use of TAD methodology in developing a hospital information system. For this purpose, let us imagine a Gastroenterological Internal Clinic, which includes Care, Diagnostic, and Surgery departments. Usually, a patient is accepted by the Reception Office and then directed to the Care Department for treatment. If the doctors decide that a certain patient needs surgery, then the patient will be directed to the Surgery Department.

The current application deals with a real example, which is given in a reduced form. In reality, the Care and Surgery departments are located in different buildings, but they are closely connected.

This hospital had no information system and almost all activities were done manually. For this reason, the management could not obtain the information needed for its daily and strategic work in a timely way. Therefore, it decided to develop a hospital information system for its clinic.

10.1 Problem Definition

Corresponding to the first phase of TAD methodology, we began by organizing two interviews with the top management. At the first meeting we asked the top management to tell us about the structure of the hospital and its strategic plan and goals and to prepare a list of analyses and outputs that are important for its daily work and related to its strategic plan.

At the second meeting we discussed the interviews plan, which we prepared after the first meeting, taking into consideration the structural scheme of the clinic. In addition, we discussed the list of analyses and outputs that was prepared and introduced by the top management as essential for its daily and strategic work.

As a result of these interviews, we defined a set of vital analyses related to the strategic plan and goals of the clinic.

T. Damij, *Tabular Application Development for Information Systems*

After creating part of the entity table corresponding to the requirements and expectations of the top management, we continued to interview the management of each department in order to define their requirements and information needs. We continued our work by organizing several interviews with the management at the business and operational levels.

Table 10.1 represents a partial entity table of the clinic. The columns of the table show the structure of the clinic, which consists of the following entities: Management, Reception Office, Care Department, Endoscopy, Ultrasound, Surgery Department. The rows of the table represent some important and vital analyses connected with the clinic's strategic, business and operational goals.

In addition to the analyses, the top management defined two decision support problems, "Medical staff work plan" and "Diet nutrition planning," indicated in the last two rows of Table 10.1. These problems were represented as difficult and time-consuming.

Table 10.1 introduces only part of the real entity table. Therefore, it contains only nine analyses required by the top management and other important entities and two decision support problems.

10.2 Systems Functioning

10.2.1. Activity Table Development

To develop the activity table of the clinic, we organized interviews with the representatives of the following entities: Reception Office, Care Department, Endoscopy, Ultrasound and Surgery Department. In every department, we first interviewed the management of the department and then continued with other users. The purpose of these interviews was to identify the activities and tasks performed in the framework of every department of the hospital and to create the activity table.

Table 10.2 shows a reduced activity table of the clinic, which has 29 activities and 14 entities. The first five entities are internal and the last nine are external.

The first activity, "Create a registration," means the Reception Office obtains the patient's data from the patient and his or her personal doctor's order. Thus we write S_{14} in square(1,14), S_{13} in square(1,13), T_{14} and T_{13} in square(1,1). Furthermore, the Reception Office creates a registration form and sends it to the Care Department. For this reason, we write S_1 in square(1,1) and T_1 in square(1,2).

The second activity, "Write anamnesis," means that the Care Department creates an anamnesis form. To indicate this, we write S_2 in square(2,2). Furthermore, concerning the Care Department entity, we

Table 10.1. Entity table of the clinic

Requirements	Entity / Analysis	Top Management	Reception Office	Care Dept.	Endoscopy	Ultrasound	Surgery
Reports	1. Information about hospital occupancy	*	*				
	2. Information about patients and diseases	*		*	*	*	*
	3. Information about successful treatments	*		*	*	*	*
	4. Information about problematic diseases	*		*	*	*	*
	5. Information about doctor's occupation	*	*	*	*	*	*
	6. Information about medications used	*		*			*
	7. Information about hospital plan realization	*					
	8. Information about successful surgeries	*		*			*
	9. Information about surgery complications	*		*			*
Decision Support	10. Medical staff work plan	*		*	*	*	*
	11. Diet nutrition plan	*		*			*

find that the first activity is a predecessor to the second activity and also that the second activity is a predecessor to the third. For this reason, we write P_1 in square(1,2), U_1 in square(2,2), P_2 in square(2,2) and U_2 in square(2,3).

Additionally, the third activity is a predecessor to the fourth activity performed by the entity Care Department. For this reason we write P_3 in square(3,2) and U_3 in square(4,2). The fourth activity is a predecessor to the fifth activity. So we write P_4 in square(4,2) and U_4 in square(5,2).

All activities are defined and connected horizontally and vertically using the same procedure.

10.2.2. TASK TABLE DEVELOPMENT

We developed the task table of the Internal Clinic, part by part, along with the interviews, which were organized with the internal entities of the Internal Clinic. At each interview we extended the activity table with the activities mentioned by the interviewee and we simultaneously extended the task table for the tasks performed in the framework of new activities.

Together with the interviewee, we defined all the characteristics of every new task: a short and clear description; conditions connected with

Table 10.2. Activity table of Internal Clinic

Activity \ Entity	1. Reception Office	2. Care Dept.	3. Endoscopy	4. Ultrasound	5. Surgery	6. Laboratory	7. Transfusion Institute	8. X-ray	9. Pathology	10. Microbiology	11. External Institution	12. Insurance Co.	13. Personal Doctor	14. Patient
1. Create a registration	T_{13}, T_{14}, S_1	T_1 P_1											S_{13}	S_{14}
2. Write anamnesis		S_2 P_2 U_1												
3. Open temperature card		S_2 P_3 U_2												
4. Create order for laboratory		S_2 P_4 U_3				T_2								
5. Accept laboratory findings		T_6 P_5, U_4				S_6								
6. Create order for endoscopy		S_2 P_6 U_3	T_2 P_6											
7. Accept endoscopy findings		T_3 P_7, U_6	S_3 U_6											
8. Create order for ultrasound		S_2 P_8 U_3		T_2 P_8										
9. Accept ultrasound findings		T_4 P_9, U_8		S_4 U_8										
10. Create order for x-ray		S_2 P_{10} U_3						T_2						
11. Accept x-ray findings		T_8 P_{11}, U_{10}						S_8						
12. Create order for pathology		S_2 P_{12} U_3							T_2					
13. Accept pathology findings		T_9 P_{13}, U_{12}							S_9					
14. Create order for microbiology		S_2 P_{14} U_3								T_2				
15. Accept microbiology findings		T_{10} P_{15}, U_{14}								S_{10}				

Table 10.2. Activity table of Internal Clinic

Activity \ Entity	1. Reception Office	2. Care Dept.	3. Endoscopy	4. Ultrasound	5. Surgery	6. Laboratory	7. Transfusion Institute	8. X-ray	9. Pathology	10. Microbiology	11. External Institution	12. Insurance Co.	13. Personal Doctor	14. Patient
16. Create order for external specialist		S_2 P_{16} U_3									T_2			
17. Accept external specialist findings		T_{11} P_{17}, U_{16}									S_{11}			
18. Displace patient to Surgery Dept.		S_2 U_5, U_7, U_{11} $U_9 U_{13} U_{15} U_{17}$			T_2 P_{18}									
19. Get patient's consent					T_{14} P_{19} U_{18}									S_{14}
20. Create premedication form					S_5 P_{20} U_{18}									
21. Order for blood & blood derivatives					S_5 P_{21} U_{20}		T_5							
22. Accept blood & blood derivatives					T_7 U_{21}		S_7							
23. Create anesthetic protocol					S_5 P_{23} U_{19}, U_{20}									
24. Create surgery protocol					S_5 P_{24} U_{19}, U_{23}									
25. Create postsurgery form					S_5 P_{25} U_{24}									
26. Prescribe therapy		S_2 P_{26} $U_5 U_7 U_9 U_{11}$ $U_{13} U_{15} U_{17}$			S_5 P_{26} U_{25}									
27. Information about state of health	T_2, T_5, S_1 P_{27}	S_2 P_{27} U_{26}			S_5 P_{27} U_{26}								T_1	
28. Create release form	T_2, T_5, S_1 U_{27}, P_{28}	S_2 U_{27}			S_5 U_{27}									T_1
29. Create invoice	S_1 U_{28}											T_1		T_1

it; and inputs or outputs accepted, created or sent by it. After each interview we rethought and rewrote the activity and task tables. The result of these interviews is given in both tables.

The activity table of the clinic (Table 10.2) contains 29 activities. Table 10.3 shows only the tasks of the first 11 activities of the activity table

Table 10.3. Task table of Internal Clinic

Activity	Characteristic Task Code	Description	Time	Condition	Input/Output
1. Create a registration	$K_{1,1}$	Reception Office accepts patient's data and doctor's reception order from patient; creates registration form and sends it to Care Department			Medical card Reception order Registration form
	$K_{1,2}$	Care Department accepts registration form from Reception Office			Registration form
2. Write anamnesis	$K_{2,2}$	Care Department writes anamnesis to describe patient's state of health			Anamnesis form
3. Open temperature card	$K_{3,2}$	Care Department opens a temperature card for every newly accepted patient			Temperature card
4. Create order for laboratory	$K_{4,2}$	Care Department creates necessary laboratory orders and sends them to laboratory			Laboratory order
5. Accept laboratory findings	$K_{5,2}$	Care Department accepts findings from laboratory			Laboratory findings
6. Create order for endoscopy	$K_{6,2}$	Care Department creates the needed orders for endoscopal diagnostics and sends them to Endoscopy			Endoscopy order
	$K_{6,3}$	Endoscopy accepts orders from Care Department			Endoscopy order
7. Accept endoscopy findings	$K_{7,2}$	Care Department accepts findings from Endoscopy			Endoscopy findings
	$K_{7,3}$	Endoscopy sends findings to Care Department			Endoscopy findings
8. Create order for Ultrasound	$K_{8,2}$	Care Department creates and sends an order to Ultrasound			Ultrasound order
	$K_{8,4}$	Ultrasound accepts order from Care Department			Ultrasound order
9. Accept ultrasound findings	$K_{9,4}$	Ultrasound sends findings to Care Department			Ultrasound findings
	$K_{9,2}$	Care Department accepts findings of Ultrasound			Ultrasound findings
10. Create order for x-ray	$K_{10,2}$	Care Department creates an order for x-ray			X-ray order
11. Accept x-ray findings	$K_{11,2}$	Care Department accepts findings from X-ray Department			X-ray findings

(Table 10.2). This should be enough to give a clear picture of the structure of the clinic's task table.

The first activity has two tasks, $K_{1,1}$ and $K_{1,2}$. These tasks deal with registration of the patient's data by the Reception Office and Care Department, respectively. The second activity has one task, which is writing anamnesis; this is $K_{2,2}$. The third activity also has one task, which is to open the temperature card; this is $K_{3,2}$. The fourth activity has one task, create an order for laboratory; this is $K_{4,2}$. The fifth activity has one task, accept the laboratory findings; this is $K_{5,2}$.

All other tasks are defined using the same procedure.

10.2.3. Work Processes and Business Processes

After completing the plan of interviews, we called a representative of each department to a joint meeting. This meeting was important to correct mistakes or misunderstandings in the created activity and task tables and to make the final version of these tables.

At the joint meeting we first analyzed every activity defined in the activity table and considered each of its tasks described in the task table. In the second part of the joint meeting we defined work processes performed in the clinic by grouping the activities into suitable collections. We continued with defining the business processes of the clinic by grouping the work processes in suitable groups.

Corresponding to this step, we grouped the activities defined in Table 10.2 into five work processes: Registration, Diagnosis, Surgery, Treatment and Release. The first work process includes one activity and deals with the reception of the patient. The second work process has 16 activities and describes different possibilities of diagnosis. The third work process contains eight activities and describes the surgical process. The fourth work process has one activity. The last work process has three activities and deals with the patient's release.

From Table 10.4 we observe that after completing the second work process, the patient may go through the third, fourth and fifth work processes if surgery is required. Otherwise, after the second work process, he or she goes through the fourth and fifth work processes.

The defined work processes are grouped into one business process: Patient. Table 10.4 shows the defined work processes and business process.

Table 10.4. Activity table of Internal Clinic

Business Process	Work Process	Activity / Entity	1. Reception Office	2. Care Dept.	3. Endoscopy	4. Ultrasound	5. Surgery	6. Laboratory	7. Transfusion Institute	8. X-ray	9. Pathology	10. Microbiology	11. External Institution	12. Insurance Co.	13. Personal Doctor	14. Patient
Patient	Registration	1. Create a registration	T_{13},T_{14}, S_1	T_1 P_1											S_{13}	S_{14}
	Diagnosis	2. Write anamnesis		S_2 P_2 U_1												
		3. Open temperature card		S_2 P_3 U_2												
		4. Create order for laboratory		S_2 P_4 U_3				T_2								
		5. Accept laboratory findings		T_6 P_5,U_4				S_6								
		6. Create order for endoscopy		S_2 P_6 U_3	T_2 P_6											
		7. Accept endoscopy findings		T_3 P_7,U_6	S_3 U_6											
		8. Create order for ultrasound		S_2 P_8 U_3		T_2 P_8										
		9. Accept ultrasound findings		T_4 P_9,U_8		S_4 U_8										
		10. Create order for x-ray		S_2 P_{10} U_3						T_2						
		11. Accept x-ray findings		T_8 P_{11},U_{10}						S_8						
		12. Create order for pathology		S_2 P_{12} U_3							T_2					
		13. Accept pathology findings		T_9 P_{13},U_{12}							S_9					
		14. Create order for microbiology		S_2 P_{14} U_3								T_2				
		15. Accept microbiology findings		T_{10} P_{15},U_{14}								S_{10}				
		16. Create order for external specialist		S_2 P_{16} U_3									T_2			
		17. Accept external specialist findings		T_{11} P_{17},U_{16}									S_{11}			

Table 10.4. (Continued) Activity table of Internal Clinic

Business Process	Work Process	Activity / Entity	1. Reception Office	2. Care Dept.	3. Endoscopy	4. Ultrasound	5. Surgery	6. Laboratory	7. Transfusion Institute	8. X-ray	9. Pathology	10. Microbiology	11. External Institution	12. Insurance Co.	13. Personal Doctor	14. Patient
Patient	Surgery	18. Transfer patient to Surgery Dept.		S_2 U_5,$U_7$$U_{11}$ $U_9$$U_{13}$$U_{15}$$U_{17}$			T_2 P_{18}									
		19. Get patient's consent					T_{14} P_{19} U_{18}									S_{14}
		20. Create premedication form					S_5 P_{20} U_{18}									
		21. Order for blood & blood derivatives					S_5 P_{21} U_{20}		T_5							
		22. Accept blood & blood derivatives					T_7 U_{21}		S_7							
		23. Create anesthetic protocol					S_5 P_{23} U_{19},U_{20}									
		24. Create surgery protocol					S_5 P_{24} U_{19},U_{23}									
		25. Create postsurgery form					S_5 P_{25} U_{24}									
	Treatment	26. Prescribe Therapy		S_2 P_{26} $U_5$$U_7$$U_9$$U_{11}$ $U_{13}$$U_{15}$$U_{17}$			S_5 P_{26} U_{25}									
	Release	27. Information about state of health	T_2,T_5,S_1 P_{27}	S_2 P_{27} U_{26}			S_5 P_{27} U_{26}								T_1	
		28. Create release form	T_2,T_5,S_1 U_{27},P_{28}	S_2 U_{27}			S_5 U_{27}									T_1
		29. Create invoice	S_1 U_{28}											T_1		T_1

10.3 Business Process Reengineering

10.3.1. Table Analysis and Reengineering

The main goal in this application was to develop an information system for the Internal Clinic. Therefore, the scope of this work did not include the possibility of carrying out BPR in the clinic.

For this reason, we did not implement the first step of the third phase, which prescribes a precise analysis of the entity, activity and task tables. We continued our work at the second step, which deals with the development of the process model of the system.

Nevertheless, we analyzed Tables 10.1, 10.3 and 10.4. As a result of this analysis we found that the fifth work process needed to be accelerated. This work process contained three activities that deal with creating documents necessary to release the patient. These activities were time-consuming, and in reality the patient had to wait several hours to obtain these documents. We hope that implementing the information system in the clinic will solve this problem.

10.3.2. Process Model

The process model of the Internal Clinic was developed by implementing the procedure described in Section 6.2 and in accordance with the information collected in the activity table (Table 10.4).

Figure 10.1 shows the process model of the Gastroenterological Internal Clinic, which represents the business process "Patient." This elementary-level DFD consists of five parts. Each of these parts introduces a certain work process.

The first part introduces the work process "Registration," which was obtained using the information stored in the first row of the activity table. As Figure 10.1 shows, this part of the DFD contains two elementary processes (circles) and two sources (rectangles). The processes are obtained by transforming the non-empty squares of the internal entities into elementary processes. The non-empty squares of the external entities are transformed into sources.

The second part shows the work process "Diagnosis." This part was created using the information collected in rows 2 to 17. It contains a number of elementary processes and sources. The third part represents the work process "Surgery." This part was developed using the content of rows 18 to 25 of the activity table. It contains more elementary processes and sources.

The fourth part introduces the work process "Treatment," which was developed using the information stored in row 26. The fifth part of the

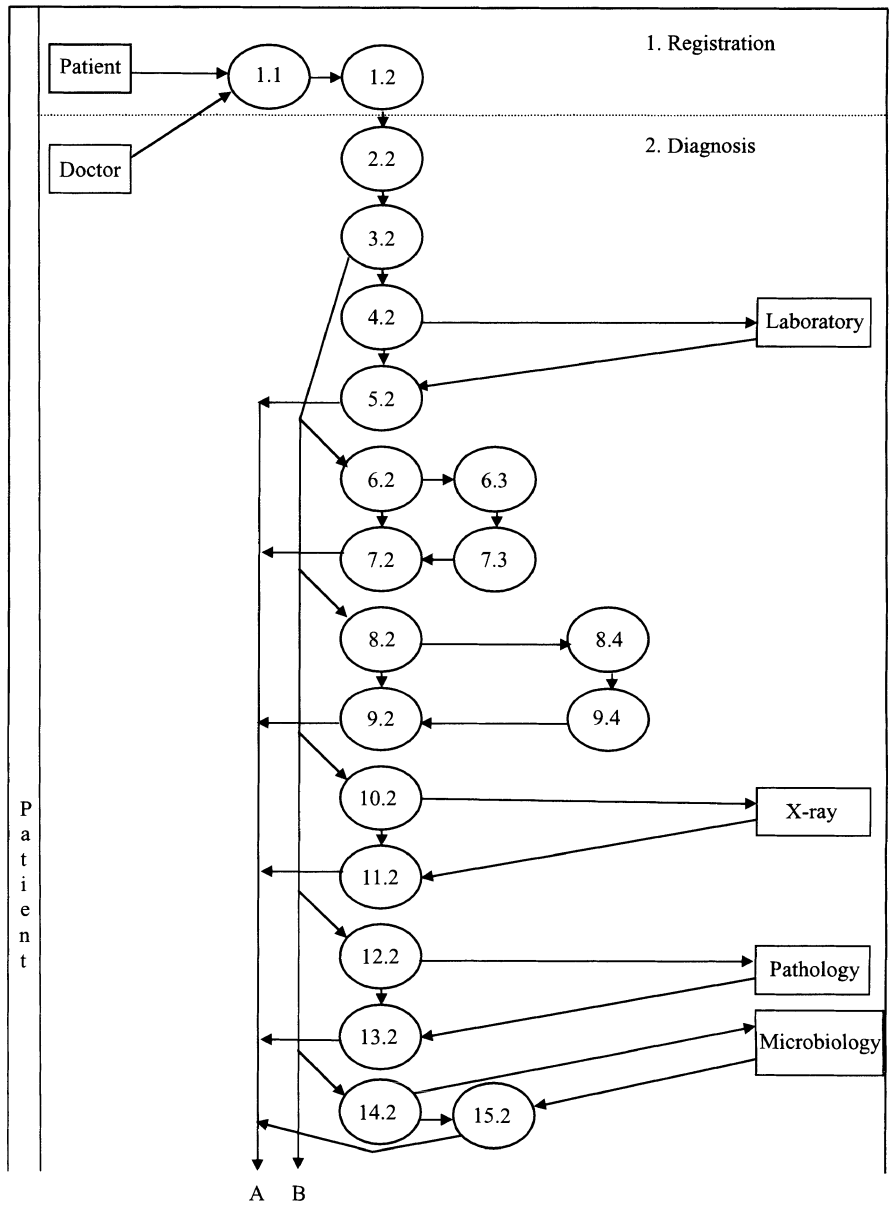


FIGURE 10.1. Process model of the clinic

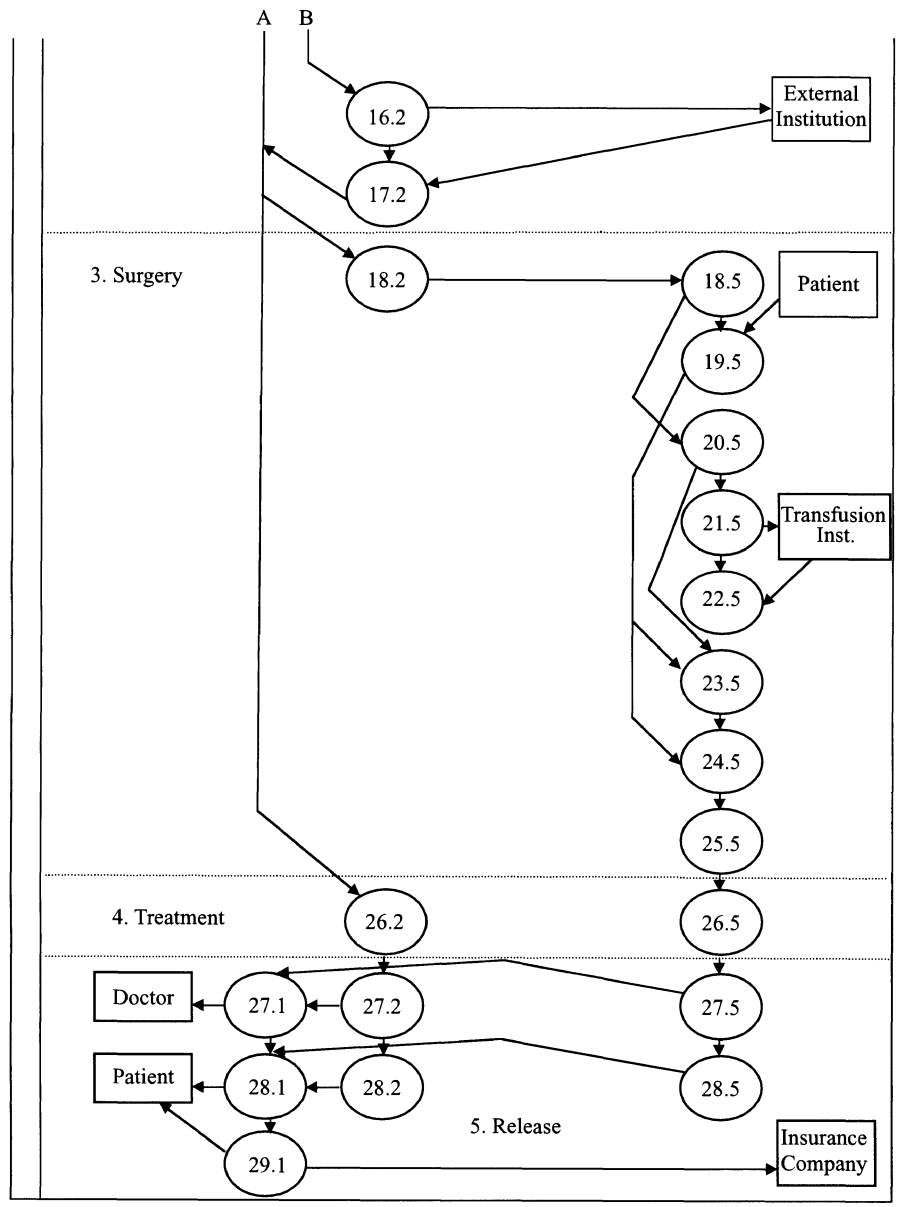


FIGURE 10.1. (Continued) Process model of the clinic

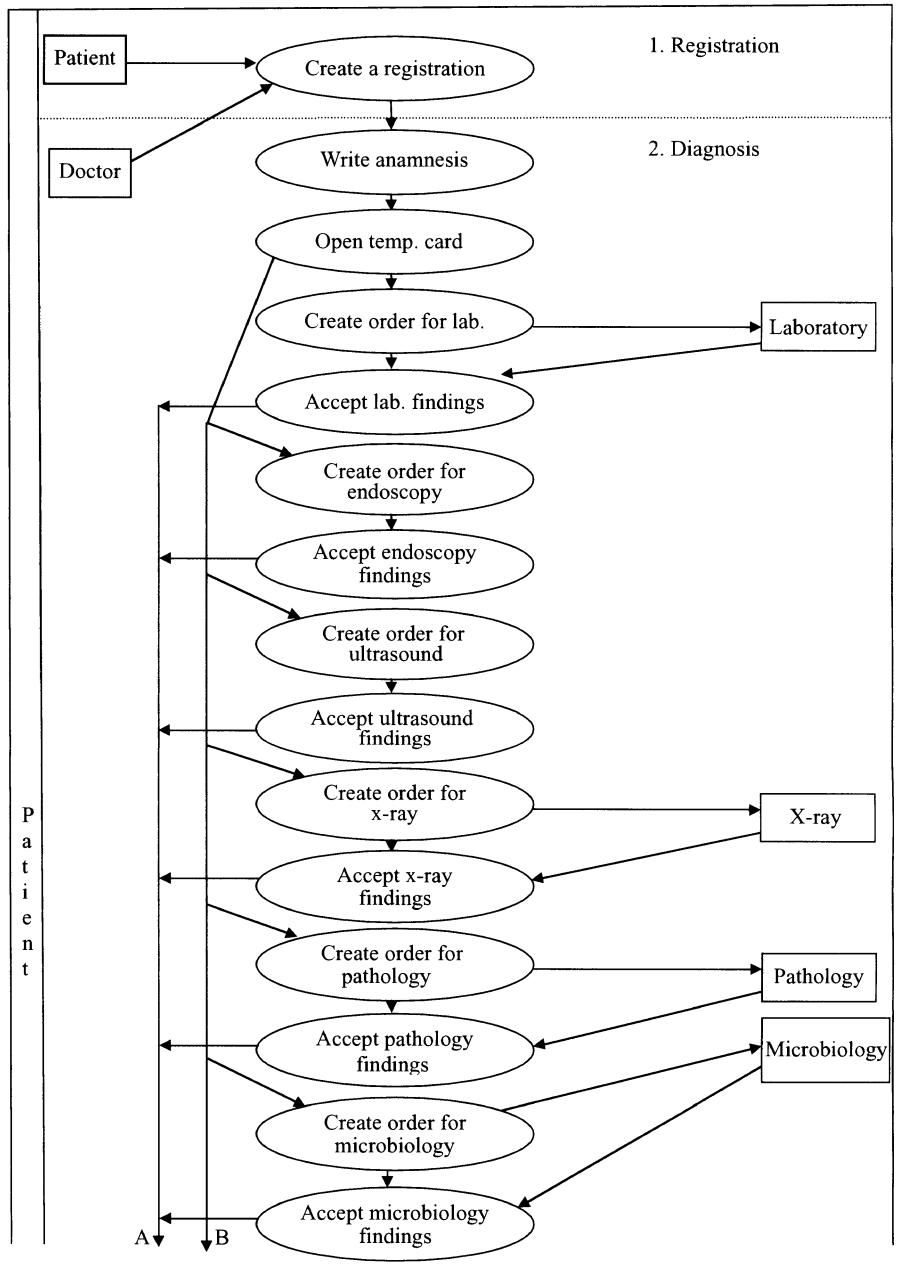


FIGURE 10.2. Activity-level DFD

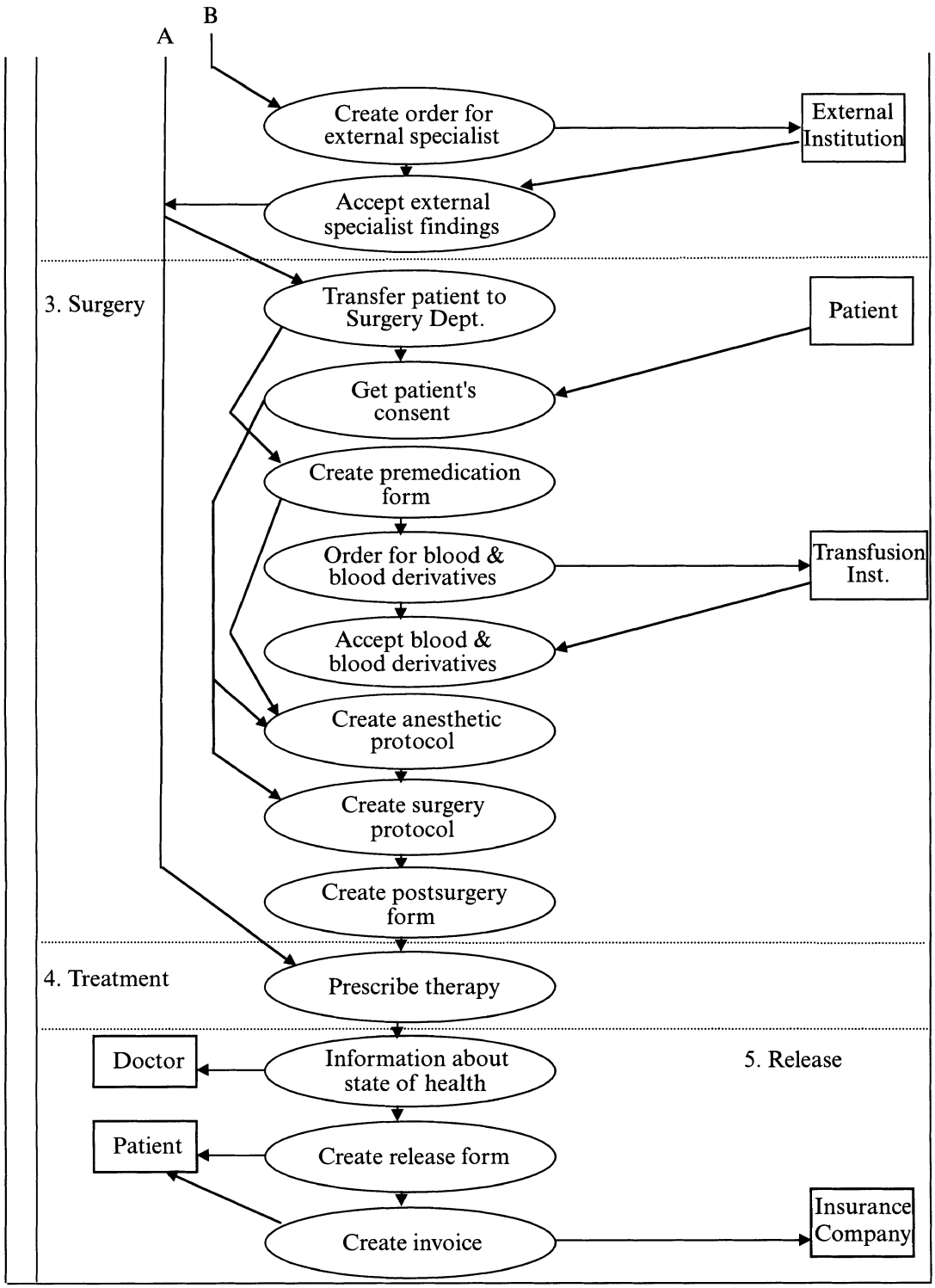


FIGURE 10.2. (Continued) Activity-level DFD

DFD shows the work process "Release." This part was created using the information collected in the last three rows.

In addition, we developed several DFDs at different levels by grouping processes of low-level DFDs into convenient groups. Figure 10.2 shows the activity-level DFD created by grouping the elementary processes of each activity into a single process. Each process in this DFD models an activity. This DFD contains 29 activity-level processes.

Figure 10.3 shows the work-process–level DFD, which is obtained by grouping processes of each work process in the activity-level DFD into a single process. Each process in the newly created DFD represents a

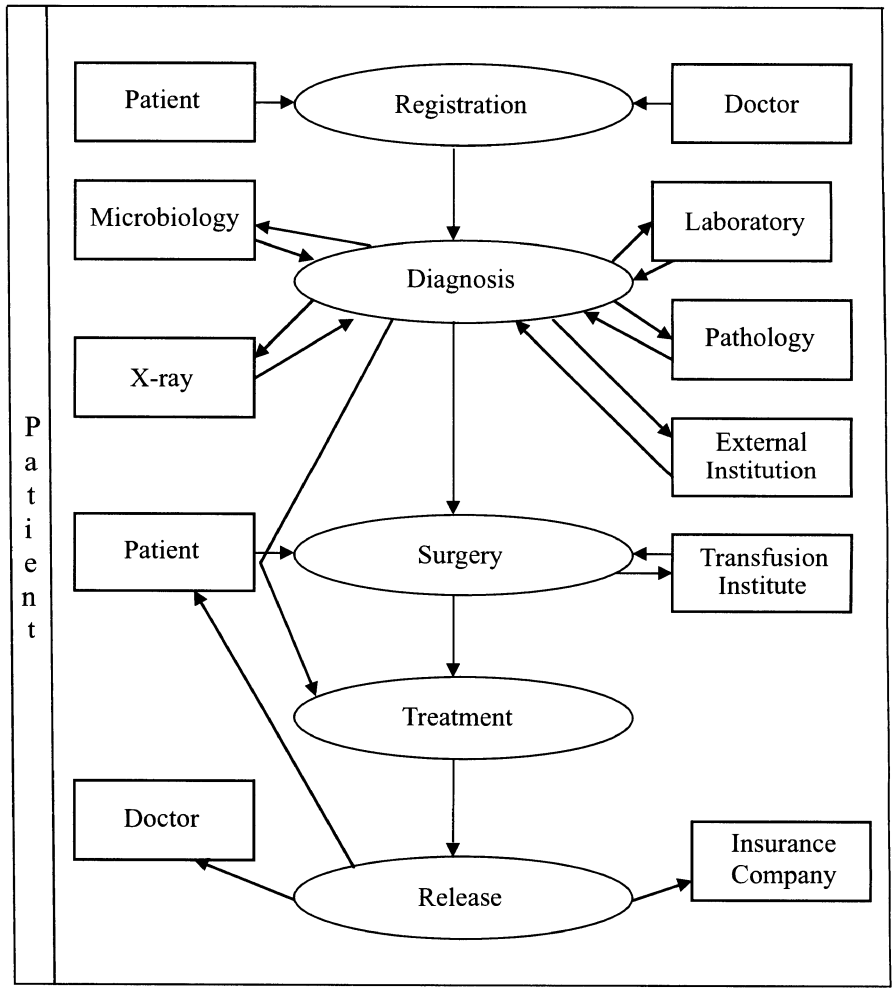


FIGURE 10.3. Work-process–level DFD

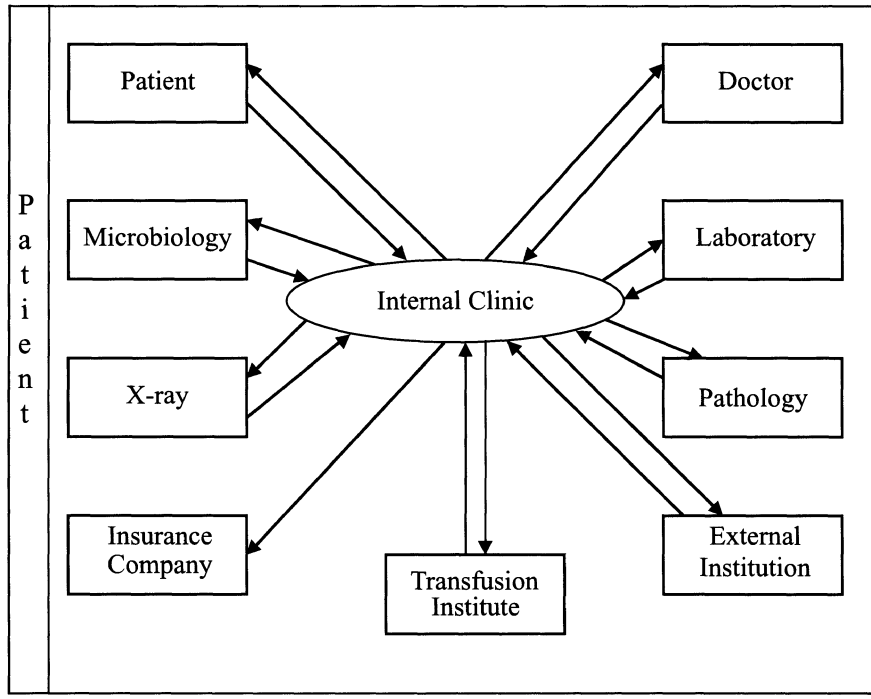


FIGURE 10.4. Context diagram

certain work process. This DFD consists of five processes: "Registration," "Diagnosis," "Surgery," "Treatment" and "Release."

Figure 10.4 introduces the business-process–level DFD. This DFD contains one process, which shows the business process "Patient."

10.4 Object Model

The initial object model of the Internal Clinic was developed by implementing the first five steps of the fourth phase described in Chapter 7, especially using the procedure developed in Section 7.3.

Corresponding to the mentioned procedure, we analyzed all users' documents indicated in the Input/Output column of the task table. For each document we first identified the identity attributes, defined the functional dependences on identity attributes, transformed the found functional dependencies and associations into suitable structures of semantic network modeling and finally developed the initial object model of the clinic.

In addition, we analyzed the entity, activity and task tables. We considered especially carefully every description defined in the Description

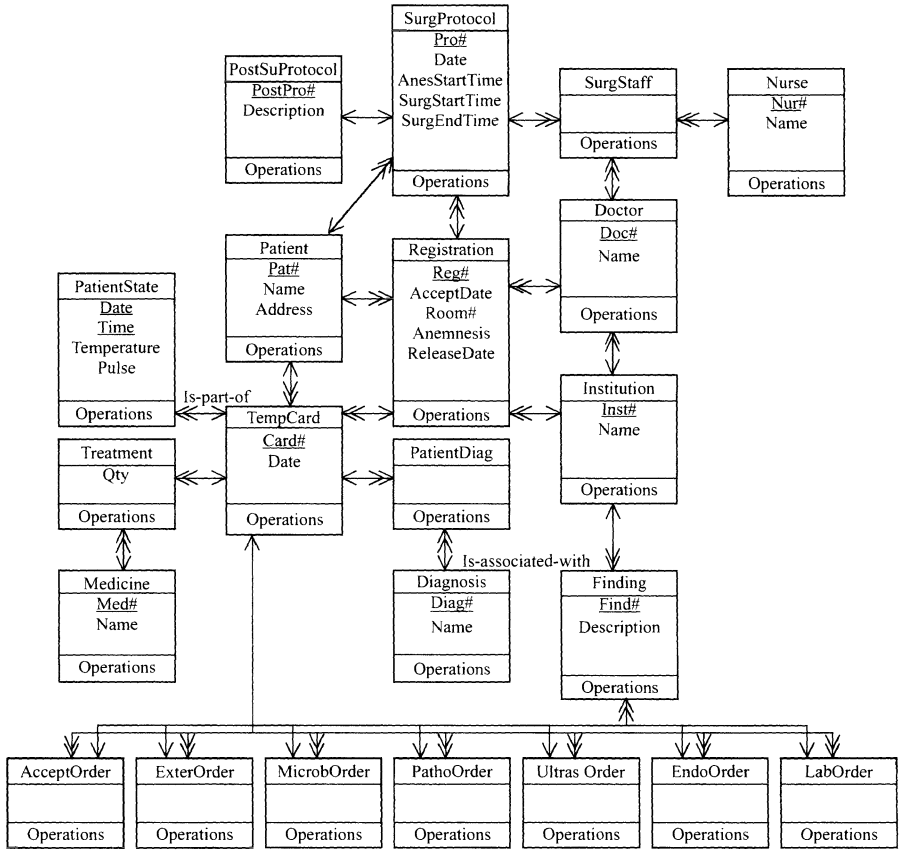


FIGURE 10.5. Initial object model of Internal Clinic

column of the task table. Any newly discovered class, attribute or association was added to the existing ones. The result of this work is the initial object model of the clinic as represented in Figure 10.5.

Figure 10.5 shows that the initial object model contains 19 object classes and 3 association classes. The object classes are Patient, Doctor, Nurse, Registration, SurgProtocol, PostSuProtocol, Institution, Finding, Diagnosis, TempCard, PatientState, Medicine, AcceptOrder, ExterOrder, MicrobOrder, PathoOrder, UltrasOrder, EndoOrder and LabOrder. The association classes are PatientDiag, SurgStaff and Treatment.

The reader may notice that object Nurse appears in the object model, but this object did not appear in the entity, activity or task table. This is because these tables are only partial tables, being simplifications of the real ones.

Corresponding to the sixth step of the procedure, we tried to use inheritance to organize the classes of the initial object model. In this manner we found seven object classes with similar characteristics; these are AcceptOrder, ExterOrder, MicrobOrder, PathoOrder, UltrasOrder, EndoOrder and LabOrder. These classes describe different types of orders used for different purposes in the hospital. For this reason we created a superclass Order, which contains the common attributes and operations of these classes.

We also created another superclass, Person, which has the common attributes and operations of classes Doctor and Nurse. This is because these two classes have many similar attributes and operations.

Figure 10.6 shows the completed object model of the Internal Clinic, which represents only part of the real object model of the clinic.

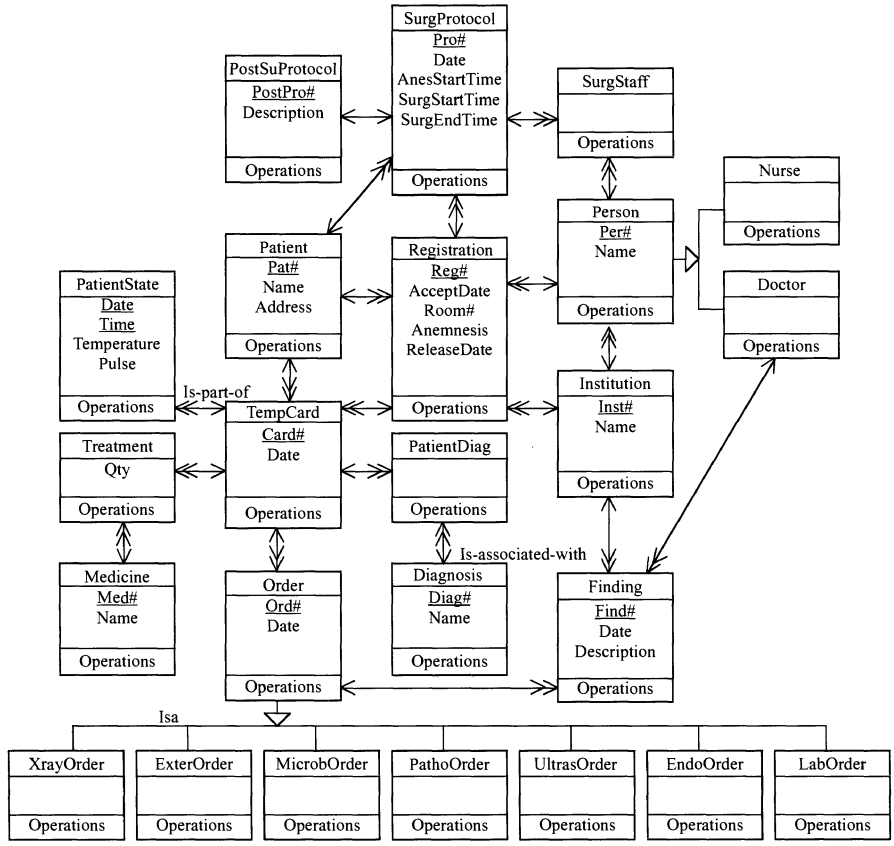


FIGURE 10.6. Object model of Internal Clinic

10.5 Design

10.5.1. OPERATIONS

Using the procedure described in Section 8.1, we defined a linkage between the functioning of the Internal Clinic described by the task and activity tables (Tables 10.3 and 10.4) and the object model of the clinic (Figure 10.6).

To achieve this linkage, we developed a new table—the operation table—and stored the specified operations in it. This table enabled us to define a linkage between the tasks and operations on the one hand and the operations and classes on the other. The operations are identified by precise analysis of the task table (Table 10.3), activity table (Table 10.4) and object model (Figure 10.6).

Table 10.5 shows the operation table of the Internal Clinic. In the first column of the table we listed 17 tasks. The second column represents 11 operations extracted from the listed tasks; these are CreateReg, WriAnamnesis, OpenCard, CreateLabOrd, AcceptFinding, CreEndoOrd, CreateUltOrd, CreXrayOrd, CrePathOrd, CreMicroOrd and CreateExtOrd. In the third column we defined the classes of the identified operations.

Figure 10.7 shows the object model extended with the operations listed in the second column of the operation table (Table 10.5). These

Table 10.5. Operation table of Internal Clinic

Task code	Operation	Class
$K_{1,1}$	CreateReg	Registration
$K_{2,2}$	WriAnamnesis	Registration
$K_{3,2}$	OpenCard	TempCard
$K_{4,2}$	CreateLabOrd	LabOrder
$K_{5,2}$	AcceptFinding	Finding
$K_{6,2}$	CreEndoOrd	EndoOrder
$K_{7,2}$	AcceptFinding	Finding
$K_{8,2}$	CreateUltOrd	UltrasOrder
$K_{9,2}$	AcceptFinding	Finding
$K_{10,2}$	CreXrayOrd	XrayOrder
$K_{11,2}$	AcceptFinding	Finding
$K_{12,2}$	CrePathOrd	PathoOrder
$K_{13,2}$	AcceptFinding	Finding
$K_{14,2}$	CreMicroOrd	MicroOrder
$K_{15,2}$	AcceptFinding	Finding
$K_{16,2}$	CreateExtOrd	ExterOrder
$K_{17,2}$	AcceptFinding	Finding

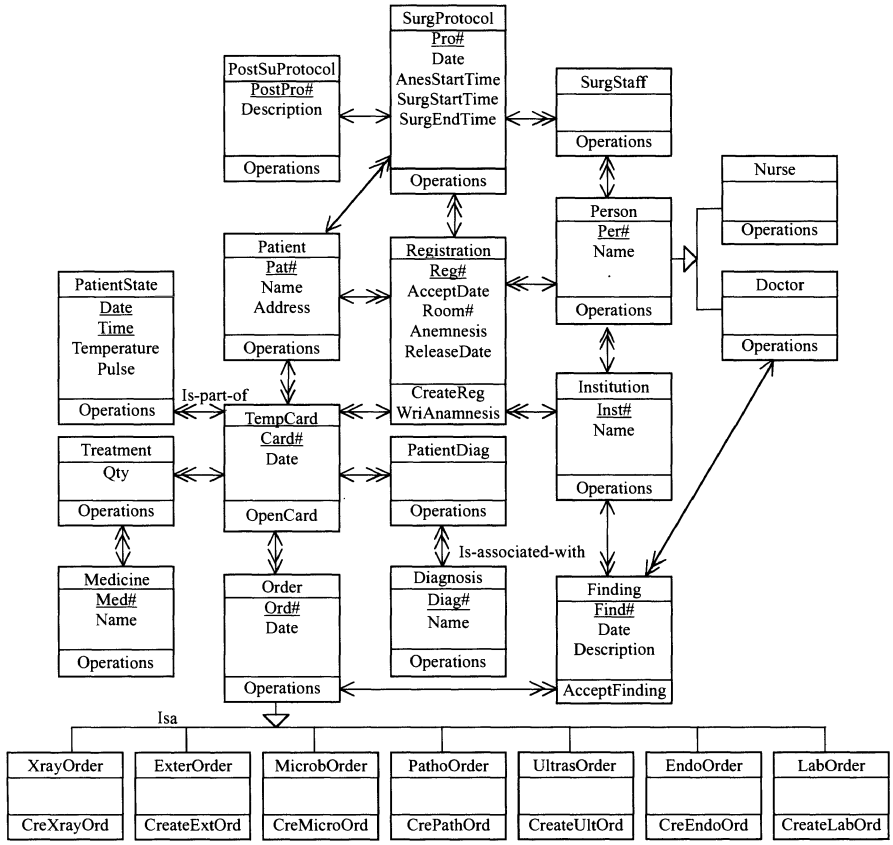


FIGURE 10.7. Object model of Internal Clinic with some named operations

operations are then defined in detail by writing an algorithm for each operation.

The reader may notice that tasks $K_{1,2}$, $K_{6,3}$, $K_{7,3}$, $K_{8,4}$ and $K_{9,4}$ are not indicated in Table 10.5. These tasks were ignored because they mean only that different departments accept certain documents that are created by already defined operations.

10.5.2. APPLICATION MODEL

Figure 10.8 shows the application model of the Internal Clinic, which consists of three parts. Corresponding to the second step of the fifth phase, we transformed the content of the entity table into the first two parts of the model. These two parts, Reports and Decision Support, are

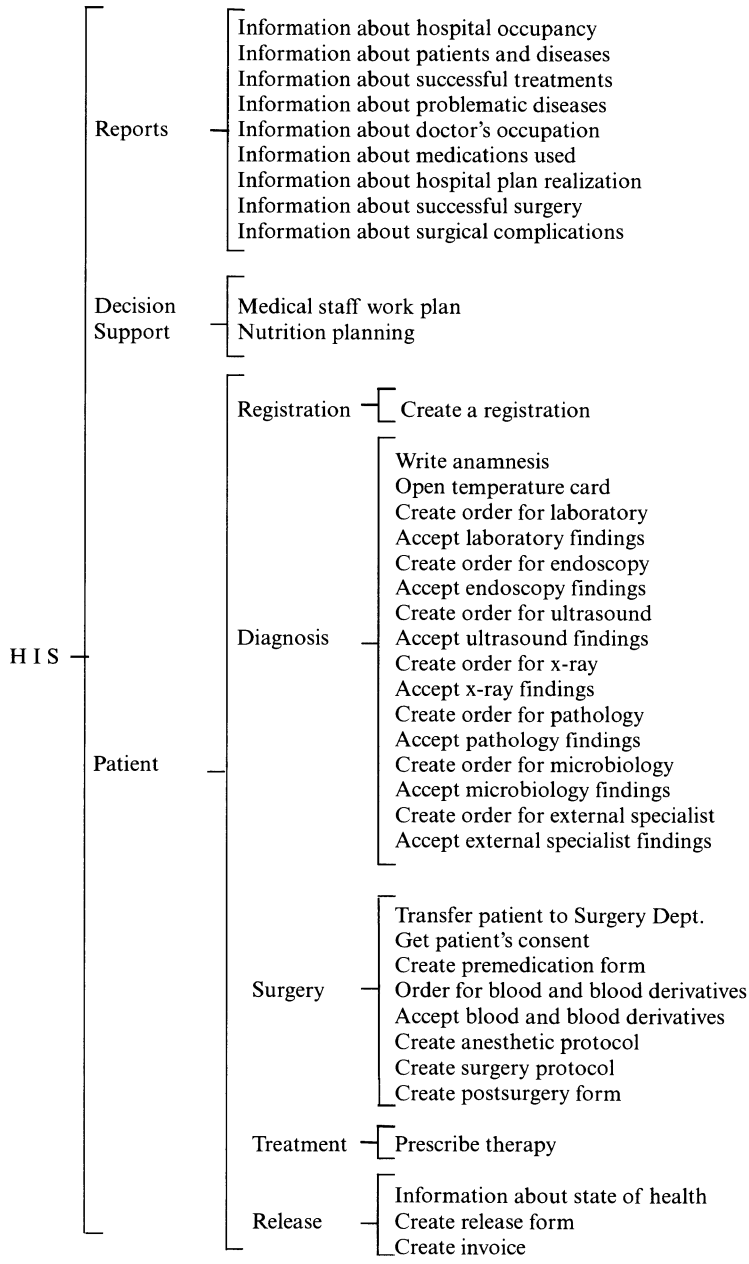


FIGURE 10.8. Application model of Internal Clinic

essential to the management's decision making. The Reports part includes nine vital analyses, such as information about the hospital, patients and their diseases, treatments, doctors and medicine. The Decision Support part deals with solving two problems: creating a work plan for the medical staff and nutrition planning.

The third part of the application model was developed from the activity table. This part represents the business process "Patient." This business process has five work processes: Registration, Diagnosis, Surgery, Treatment and Release; see Table 10.4.

The first work process consists of one activity, which deals with the way of accepting a patient by registering different important information about him or her. The second work process contains 16 activities, which represent all events and actions connected to the diagnosis of the disease.

The third work process includes eight activities, which introduces all activities related to the surgery. The fourth work process has one activity and deals with prescribing any needed therapy. The fifth work process includes the three activities linked to patient release.

CHAPTER 11

Clinic for Small Animals

This chapter introduces the third application: the use of TAD methodology to develop an information system in a clinic for small animals. The clinic had no information system. Its work involved three nonconnected small computer applications in different locations.

The first application was located in the Reception Office. This application registered patients and created a bill for treatment at the end of the treatment. To do so, the receptionist had to insert data in the application about various kinds of treatment for each patient.

The second application was located in the X-ray unit. This application stored data about X-ray pictures and printed reports about them. The third application was in the pharmacy. This application dealt only with registering received medicine, nutrition and other medical materials. It also printed a report about this.

11.1 Problem Definition

To implement the first phase of TAD methodology, we organized several interviews with the users. First we interviewed the top management and then continued with other users.

The purpose of the first interview with the top management was to understand how the clinic is organized, agree that the management should prepare a plan of the clinic's strategic goals and objectives, and create a plan of interviews with employees.

At the second meeting we discussed the strategic plan of the organization and also defined the vital analyses that are related to the strategic plan. Furthermore, we discussed the clinic's organizational scheme. We found the clinic has 10 specialized infirmaries in addition to the Reception Office, Laboratory, X-ray Department, Surgery, Pharmacy and

T. Damij, *Tabular Application Development for Information Systems*

Table 11.1. Entity table of the clinic

Requirements	Analysis \ Entity	Top Management	Reception Office	Infirmary	Hospital	X-ray	Laboratory	Pharmacy
Reports	1. Daily financial report	*	*					
	2. Analysis of income & outcome of every unit	*		*	*	*	*	*
	3. Analysis of hospital occupancy	*			*			
	4. Information about medicine & nutrition	*			*			*
	5. Analysis of nutrition expenses	*			*			*
	6. Information about diseases	*		*	*	*	*	*
	7. Analysis about patients grouped by breed and sex	*		*	*	*	*	
	8. Analysis of medications used	*		*	*			*
	9. Information about surgeries	*		*	*			
	10. Analysis of specialist's occupation	*		*	*	*	*	*
Decision Support	11. Medical staff work plan	*		*	*	*		

Hospital. These are the Internal, Dental, Trauma, Infections, Cancer, Skin, Cardiac, Opthamalic, Endocrine and Neurological infirmaries.

We continued our work according to the defined plan of interviews with the management of the above-listed units and infirmaries. The result of these interviews was the development of Table 11.1; this is the entity table of the clinic. It has to be emphasized that Table 11.1 represents a very reduced origin table. The columns of this table represent the structure of the clinic. The rows of the table show a list of the identified vital analyses related to the strategic plan and goals and to the requirement of the representatives.

The reader may notice that in the columns of the table we defined only one Infirmary to represent all 10 infirmaries, because of the space limitation in the book and also because these infirmaries have similar requirements and use similar documentation.

11.2 Systems Functioning

11.2.1. Activity Table Development

Corresponding to the first step of the second phase, we continued with interviewing the management at unit level. In each unit or infirmary we

organized at least two interviews with the responsible veterinarian until everything was clear concerning his or her unit.

The result of these interviews was the activity table of the clinic, which in its real dimension has 98 activities and 29 entities. Table 11.2 shows the activity table of the clinic in a reduced form. This table includes only 24 activities and 13 entities. The first nine entities are internal and the last four are external. In spite of the reduction, it creates a clear and understandable picture of the functioning of the system.

The first activity—"Reception of patient"—deals with patient registration. The Reception Office obtains patient data from its owner and directs it to the Internal or Trauma infirmary. If the patient has agreed to a date for treatment in another specific infirmary, then the Reception directs it to this particular infirmary. For this reason, we write S_{13} in square(1,13), T_{13} in square(1,1), S_1 in square(1,1) and T_1 in squares (1,2) and (1,8).

The second activity, "Create patient's bill," means that at the end of the treatment process the Reception Office prints a treatment bill. This is why we write S_1 in square(2,1) and T_1 in square(2,13). In addition, we write P_1 in square(1,1) and U_1 in square(2,1) to indicate that the first activity is a predecessor to the second one concerning the Reception Office column in the activity table.

The third activity, "Create patient protocol," means that the infirmary (Internal or Trauma) creates a general protocol for each patient. To indicate this we write S_2 in square(3,2). A specialized protocol is added to the general one if the patient is directed to any other infirmary.

The fourth activity—"Create a diagnostic order"—means that the infirmary creates an order and sends it to Endoscopy, Ultrasound, X-ray, Laboratory, Microbiology, or Pathology. For this reason, we write S_2 in square(4,2) and T_2 in squares (4,3), (4,4), (4,5), (4,6), (4,10) and (4,11).

The fifth activity, "Accept finding," means that the infirmary accepts the findings from the above-mentioned entities. We write S_{10} and S_{11} in squares (5,10), (5,11), and T_{10}, T_{11} in square (5,2) to register that the infirmary accepts the findings of Microbiology and/or Pathology. These two entities are external. For this reason, accepting their findings is indicated in the row of the fifth activity.

Concerning the Infirmary column, we can see that the first activity is a predecessor to the third one and the third activity is a predecessor to the fourth one. For this reason, we write P_1 in square(1,2), U_1 in square(3,2), P_3 in square(3,2) and U_3 in square(4,2). Furthermore, we find U_9, U_{10}, U_{11} and U_{12} in square(5,2). This indicates acceptance of the findings of the internal entities Endoscopy, Ultrasound, X-ray and Laboratory. This means that the fifth activity is a successor to the ninth, tenth, eleventh and twelfth activities.

Table 11.2. Activity table of the clinic

Activity \ Entity	1. Reception Office	2. Infirmary	3. Endoscopy	4. Ultrasound	5. X-ray	6. Laboratory	7. Hospital	8. Surgery	9. Pharmacy	10. Microbiology	11. Pathology	12. Supplier	13. Owner
1. Reception of patient	T_{13},S_1 P_1	T_1 P_1						T_1 P_1					S_{13}
2. Create patient's bill	S_1 $U_1U_6U_{18}U_{21}$												T_1
3. Create patient protocol		S_2 U_1,P_3											
4. Create a diagnostic order		S_2 U_3,P_4	T_2 P_4	T_2 P_4	T_2 P_4	T_2 P_4				T_2	T_2		
5. Accept findings		$T_{10}T_{11}$ P_5 $U_4U_9U_{10}U_{11}$,U_{12}								S_{10}	S_{11}		
6. Prescribe therapy	T_2 P_6	S_2 U_5,P_6											
7. Direct patient to hospital		S_2 U_5					T_2 P_7						
8. Order for medicine		S_2 U_6							T_2 P_8				
9. Create Endoscopy findings		T_3 P_9	S_3 U_4,U_{14}				T_3 P_9						
10. Create Ultrasound findings		T_4 P_{10}		S_4 U_4,U_{14}			T_4 P_{10}						
11. Create X-ray findings		T_5 P_{11}			S_5 U_4,U_{14}		T_5 P_{11}						
12. Create Laboratory findings		T_6 P_{12}				S_6 U_4,U_{14}	T_6 P_{12}						
13. Open temperature form							S_7 $P_{13}U_7U_{21}$						
14. Create diagnostic order			T_7 P_{14}	T_7 P_{14}	T_7 P_{14}	T_7 P_{14}	S_7 P_{14},U_{13}			T_7	T_7		

Table 11.2. (Continued) Activity table of the clinic

Activity \ Entity	1. Reception Office	2. Infirmary	3. Endoscopy	4. Ultrasound	5. X-ray	6. Laboratory	7. Hospital	8. Surgery	9. Pharmacy	10. Microbiology	11. Pathology	12. Supplier	13. Owner
15. Accept findings							$T_{10}T_{11}$ U_9,P_{15} $U_{10}U_{11}$ U_{12},U_{14}			S_{10}	S_{11}		
16. Prescribe therapy							S_7 P_{16},U_{15}						
17. Direct to surgery							S_7 P_{17},U_{15}	T_7 P_{17}					
18. Release the patient	T_7 P_{18}						S_7 U_{16}						
19. Order for medicine & nutrition							S_7 U_{16}		T_7 P_{19}				
20. Create surgery protocol								S_8 P_{20} U_1,U_{17}					
21. Direct to hospital or release	T_8 P_{21}						T_8 P_{21}	S_8 U_{20}					T_8
22. Issue medicine and nutrition		T_9					T_9		S_9 P_{22} U_8U_{19}				T_9
23. Create order for medicine & nutrition									S_9 U_{22},P_{23}			T_9	
24. Accept medicine & nutrition									T_{12} U_{23}			S_{12}	

11.2.2. TASK TABLE DEVELOPMENT

In accordance with the second step of the second phase, we created the task table of the clinic. This table was developed simultaneously as we developed the activity table of the clinic.

After every interview with the particular veterinarian responsible for a determined infirmary, we extended the activity table by the newly identified activities and also described them in the task table.

Table 11.3 shows the task table of the clinic, which actually represents only part of the real task table. Table 11.3 shows only the first 10 activities of the activity table (Table 11.2). The complete table contains 98 total activities.

The first activity includes three tasks; these are $K_{1,1}$, $K_{1,2}$ and $K_{1,8}$, which deal with the reception of the patient in an infirmary or for sur-

Table 11.3. Task table of the clinic

Activity	Characteristic Task Code	Description	Time	Condition	Input/ Output
1. Reception of patient	$K_{1,1}$	Reception Office obtains patient's data from its owner			Patient's data
	$K_{1,2}$	Reception Office sends patient's data to suitable infirmary			Patient's data
	$K_{1,8}$	Surgery receives patient's data from Reception Office			Patient's data
2. Create patient's bill	$K_{2,1}$	After completing the process of treatment, the Reception Office prints treatment bill, which owner pays			Bill
3. Create patient protocol	$K_{3,2}$	Infirmary creates a general protocol for each patient			GProtocol
4. Create a diagnostic order	$K_{4,2}$	Infirmary creates different diagnostic orders for Endoscopy, Ultrasound, X-ray, Laboratory, Microbiology or Pathology			Diagnostic Order
5. Accept findings	$K_{5,2}$	Infirmary accepts findings from Endoscopy, Ultrasound, X-ray, Laboratory, Microbiology or Pathology			Findings
6. Prescribe therapy	$K_{6,2}$	Infirmary prescribes for every patient suitable therapy based on the received findings			GProtocol
7. Direct patient to the hospital	$K_{7,2}$	Infirmary points the patient to the hospital			GProtocol
	$K_{7,7}$	Hospital accepts the patient			GProtocol
8. Order for medicine	$K_{8,2}$	Infirmary orders the needed medicine and other medical material			Order form
	$K_{8,9}$	Pharmacy accepts order from infirmary			Order form
9. Create Endoscopy findings	$K_{9,3}$	Endoscopy creates findings and sends them to infirmary or hospital			Findings
10. Create Ultrasound findings	$K_{10,4}$	Ultrasound creates findings and sends them to infirmary or hospital			Findings

gery. The second activity has one task: $K_{2,1}$. This task prints out the patient's treatment bill, which the owner then pays. The third activity also has one task, $K_{3,2}$, which creates the patient's general protocol.

All other tasks are described using the same concept.

11.2.3. WORK PROCESSES AND BUSINESS PROCESSES

After completing the process of interviews, and corresponding to the third step of the second phase, we organized a joint meeting with almost all representatives of the mentioned units and infirmaries. At this meeting we introduced the entity table and its relations to the strategic plan and goals. In addition, we introduced the activity and task tables to them to show the functioning of the clinic as a whole and also each task in it.

After discussing the content of the mentioned tables, we agreed on a new plan of interviews with the representatives. So each particular representative could again check the work in his or her unit.

After completing this detailed analysis, we grouped the activities into work processes and work processes into business processes. The representatives at the joint meeting agreed to our suggestion. Table 11.4 shows that all activities are grouped into nine work processes; these are Reception, Treatment, Endoscopy, Ultrasound, X-ray, Laboratory, Hospital, Surgery and Pharmacy. These work processes are grouped into two business processes: Patient and Medicine & Nutrition.

In the process of analyzing Table 11.4, we discovered the following problem: the eighth activity ("Order for medicine") and the nineteenth activity ("Order for medicine & nutrition") are predecessor activities to the twenty-second activity. To indicate this, we write P_8 in square(8,9), P_{19} in square(19,9) and U_8 and U_{19} in square(22,9).

When we tried to link the twenty-second activity to its successor, concerning entity Infirmary, we found that such a successor activity does not exist in the activity table. But the Infirmary has to have an activity that performs the reception of the issued medicine. This means that a new activity, "Accept medicine," had to be defined in the framework of work process "Treatment" and indicated as a successor activity to the twenty-second activity concerning the entity Infirmary.

The same problem was found regarding defining a linkage between the twenty-second activity and its successor concerning the entity Hospital. For this reason, a new activity, "Accept medicine & nutrition," had to be defined in the framework of work process "Hospital" and indicated as a successor to the twenty-second activity concerning the entity Hospital.

The newly found activities are defined in the original table. Because of space limitation and to show the problems, we left the activity table as it is.

Table 11.4. Activity table of the clinic

Business Process	Work Process	Activity \ Entity	1. Reception Office	2. Infirmary	3. Endoscopy	4. Ultrasound	5. X-ray	6. Laboratory	7. Hospital	8. Surgery	9. Pharmacy	10. Microbiology	11. Pathology	12. Supplier	13. Owner
Patient	Reception	1. Reception of Patient	T_{13},S_1 P_1	T_1 P_1						T_1 P_1					S_{13}
		2. Create patient's bill	S_1 U_1U_6 $U_{18}U_{21}$												T_1
	Treatment	3. Create patient protocol		S_2 U_1,P_3											
		4. Create a diagnostic order		S_2 U_3,P_4	T_2 P_4	T_2 P_4	T_2 P_4	T_2 P_4				T_2	T_2		
		5. Accept findings		$T_{10}T_{11}$ P_5 $U_4U_9U_{10}$ U_{11},U_{12}								S_{10}	S_{11}		
		6. Prescribe therapy	T_2 P_6	S_2 U_5,P_6											
		7. Direct patient to hospital		S_2 U_5					T_2 P_7						
		8. Order for medicine		S_2 U_6							T_2 P_8				
	Endoscopy	9. Create Endoscopy findings		T_3 P_9	S_3 U_4,U_{14}				T_3 P_9						
	Ultrasound	10. Create Ultrasound findings		T_4 P_{10}		S_4 U_4,U_{14}			T_4 P_{10}						
	X-ray	11. Create X-ray findings		T_5 P_{11}			S_5 U_4,U_4		T_5 P_{11}						
	Laboratory	12. Create Laboratory findings		T_6 P_{12}				S_6 U_4,U_{14}	T_6 P_{12}						

Table 11.4. (Continued) Activity table of the clinic

Business Process	Work Process	Activity \ Entity	1. Reception Office	2. Infirmary	3. Endoscopy	4. Ultrasound	5. X-ray	6. Laboratory	7. Hospital	8. Surgery	9. Pharmacy	10. Microbiology	11. Pathology	12. Supplier	13. Owner
Patient	Hospital	13. Open temperature form							S_7 $P_{13}U_7U_{21}$						
		14. Create diagnostic order			T_7 P_{14}	T_7 P_{14}	T_7 P_{14}	T_7 P_{14}	S_7 P_{14},U_{13}			T_7	T_7		
		15. Accept findings							$T_{10}T_{11}$ U_9,P_{15} $U_{10}U_{11}$ U_{12},U_{14}			S_{10}	S_{11}		
		16. Prescribe therapy							S_7 P_{16},U_{15}						
		17. Direct to surgery							S_7 P_{17},U_{15}	T_7 P_{17}					
		18. Release the patient	T_7 P_{18}						S_7 U_{16}						
		19. Order for medicine & nutrition							S_1 U_{16}		T_7 P_{19}				
	Surgery	20. Create surgery protocol								S_8 P_{20} U_1,U_{17}					
		21. Direct to hospital or release	T_8 P_{21}						T_8 P_{21}	S_8 U_{20}					T_8
Medicine & Nutrition	Pharmacy	22. Issue medicine and nutrition		T_9					T_9		S_9 P_{22} U_8U_{19}				T_9
		23. Create order for medicine & nutrition									S_9 U_{22},P_{23}			T_9	
		24. Accept medicine & nutrition									T_{12} U_{23}			S_{12}	

The above-discussed problems prove again that the activity table is a very precise tool that enables us to discover any deficit or mistake existing in the table.

11.3 Business Process Reengineering

11.3.1. Table Analysis and Reengineering

The first step of the third phase required the establishment of a project team to analyze carefully the entity, activity and task tables. The clinic's project team was established from three members: the analyst, a member of the top management and the responsible veterinarian of the Internal infirmary.

The result of the analysis of the entity, activity and task tables was that there was no need for changes in the structure or in the functioning of the organization, because the functioning was very satisfactory.

Nevertheless, the project team identified several problems that needed urgent solutions and informed top management about them. These problems were as follows:

- the pharmacy was without an employee and there was an urgent need for an employee to take care of the reception and issue of medicine, nutrition and other medical material;
- the medical staff was very occupied; there was a special need for more nurses;
- as mentioned before, there was a need to define two new activities. The first activity in the framework of work process "Treatment" is to perform the reception of the issued medicine. The second activity in the framework of work process "Hospital" is to ensure the reception of the issued medicine and nutrition.

11.3.2. Process Model

As mentioned earlier, the development of the process model may be skipped because the activity table is more visible and understandable than the process model.

Corresponding to the second step of the third phase, which describes the development of the process model of the system, we first developed the elementary-level DFD of the clinic and then continued by creating DFDs at activity, work-process and business-process levels. Finally, we developed the context diagram of the system.

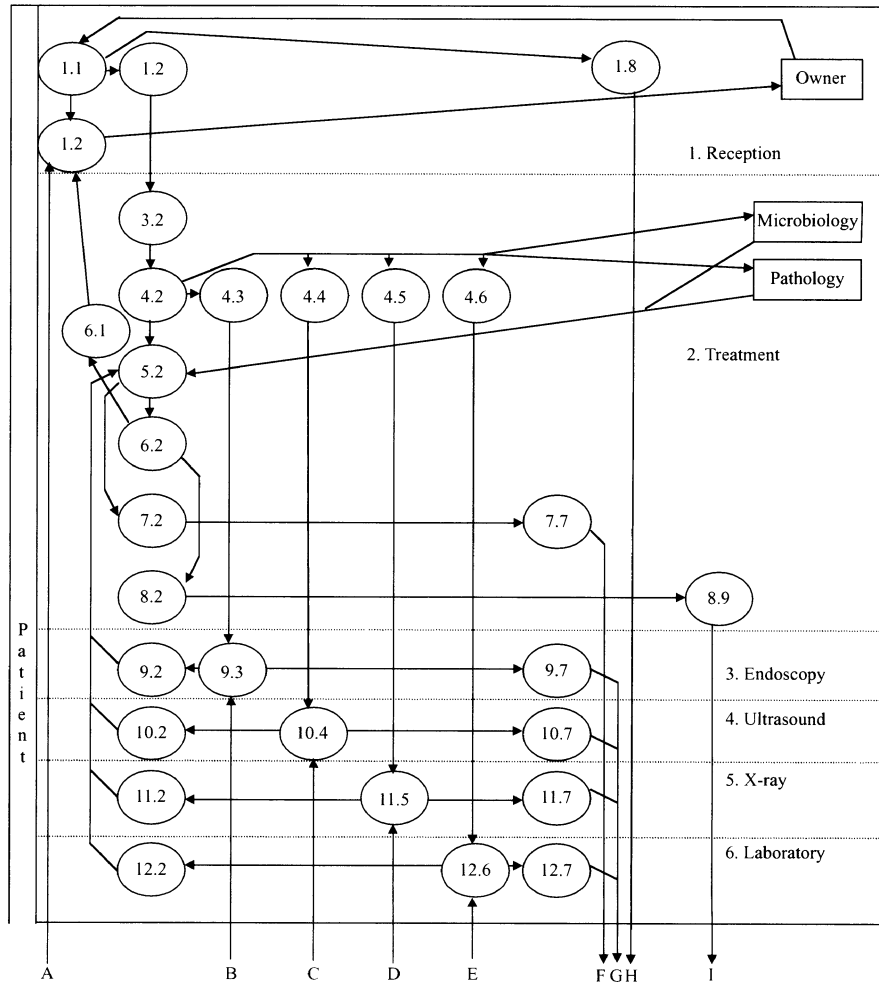


FIGURE 11.1. Elementary-level process model of the clinic

Figure 11.1 shows the elementary-level DFD of the clinic. This process model was developed in accordance with the information collected in the activity table (Table 11.4). This DFD consists of nine parts, which represent nine work processes.

The first part introduces the work process "Reception," which was created using the information collected in the first two rows of Table 11.4. The second part represents the work process "Treatment," which was developed in accordance with information stored in rows 3 to 8.

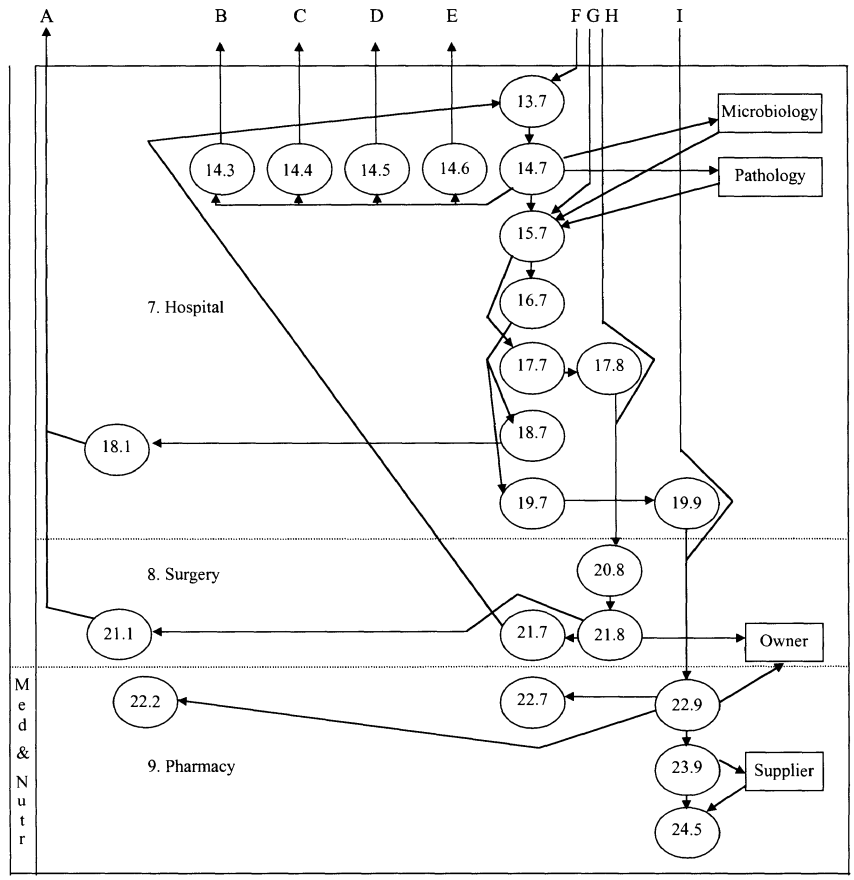


FIGURE 11.1. (Continued) Elementary-level process model of the clinic

The third, fourth, fifth and sixth parts model the work processes "Endoscopy," "Ultrasound," "X-ray" and "Laboratory." To develop these parts, we used information collected in rows 9 to 12.

The seventh part introduces the work process "Hospital." This part was created using information stored in rows 13 to 19. The eight and ninth parts represent the work processes "Surgery" and "Pharmacy." The eighth part was developed from information in rows 20 and 21. The ninth part was developed using information collected in the last three rows.

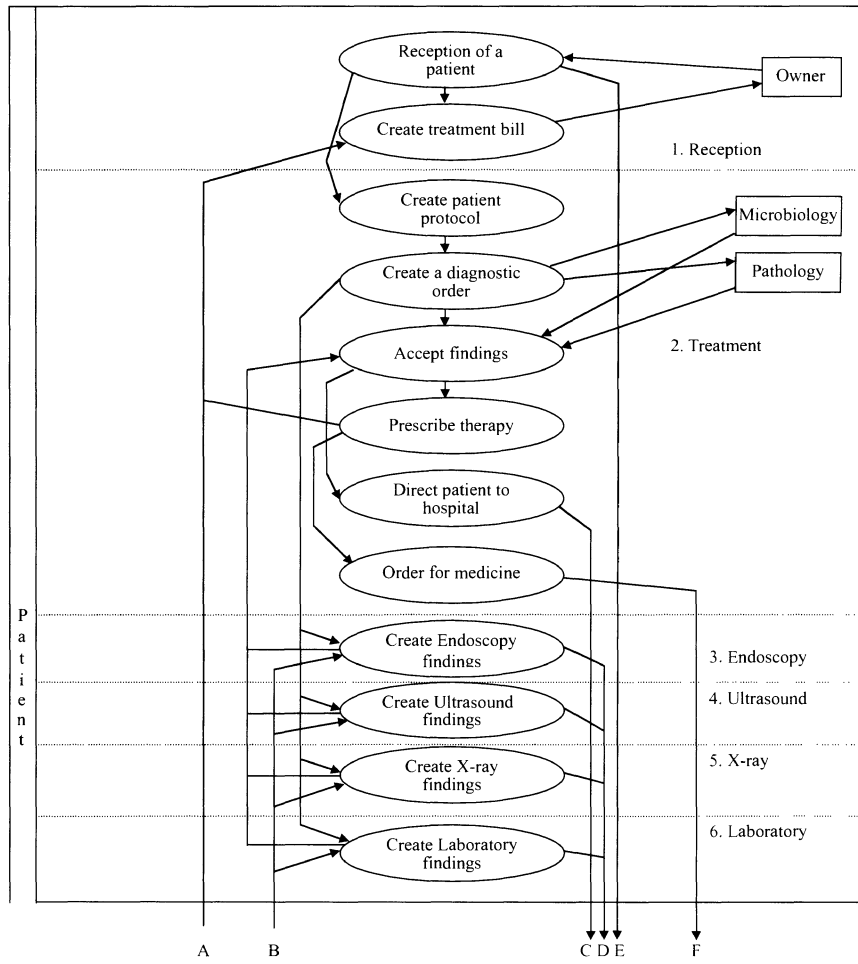


FIGURE 11.2. Activity-level process model of the clinic

Figure 11.2 shows the activity-level DFD. This DFD consists of 24 processes. Each process represents a certain activity of the activity table. Figure 11.3 describes the work-process–level DFD, which consists of nine processes. Every process in this DFD indicates a determined work process.

Figure 11.4 shows the business-process–level DFD, which has two processes. Each of these processes represents a determined business

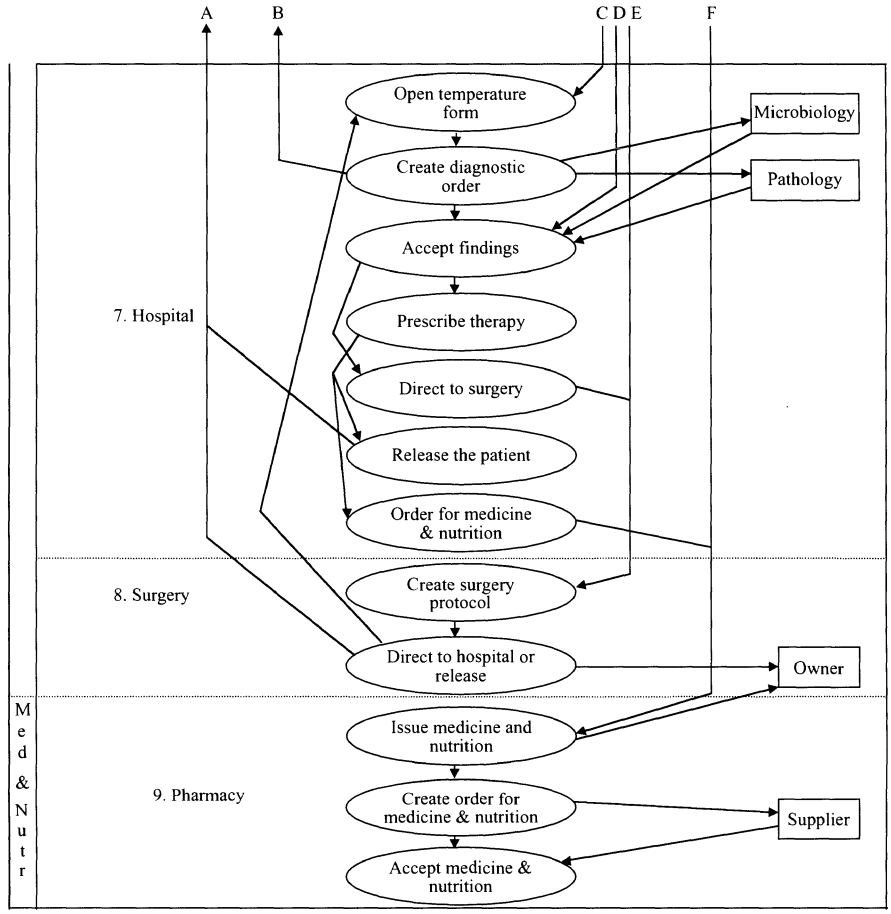


FIGURE 11.2. (Continued) Activity-level process model of the clinic

process. Finally, Figure 11.5 introduces the context diagram of the system.

11.4 Object Model

In accordance with the fourth phase of TAD methodology and using the procedure described in Section 7.3, we carefully analyzed the documents indicated in the Input/Output column of the task table.

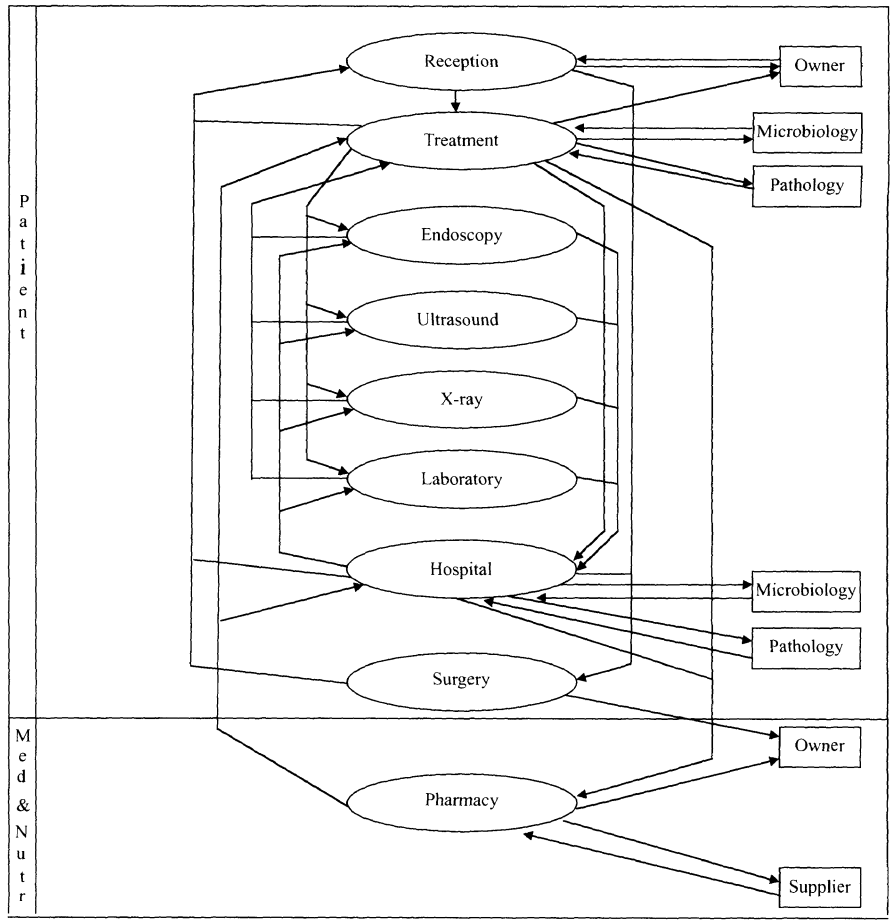


FIGURE 11.3. Work-process–level process model of the clinic

For each document analyzed, we identified several identity attributes of different object classes. The functional dependencies of other attributes on the identified identity attributes were defined. Furthermore, the associations existing between the identity attributes were determined. The relationships defined between the identity attributes and other attributes and also between the identity attributes themselves were then transformed into suitable structures of semantic network modeling. Finally, the result of this analysis was used to develop the initial object model.

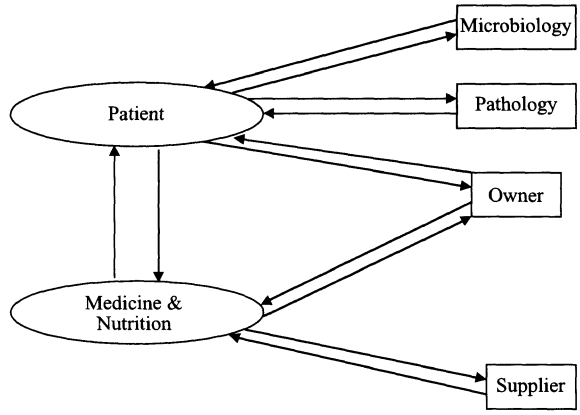


FIGURE 11.4. Business-process–level DFD of the clinic

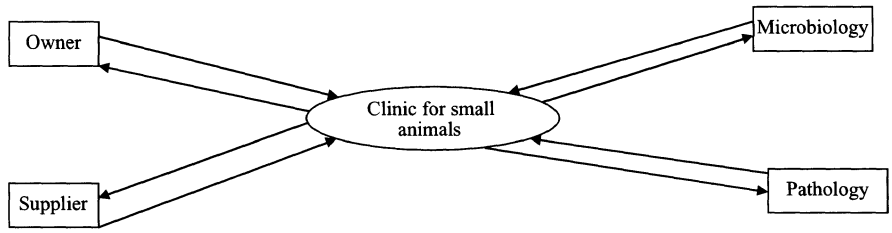


FIGURE 11.5. Context diagram

We continued our work by analyzing the content of the Description column in the task, activity and entity tables. Any newly discovered object class, attribute or association was added to the existing initial object model.

Figure 11.6 represents the initial object model of the clinic. This object model consists of 18 object classes and 3 association classes.

Corresponding to the last step of the fourth phase, we implemented inheritance between classes. For this purpose, we found six object classes that share similar characteristics: LabOrder, XrayOrder, UltraOrder, EndoOrder, MicroOrder and PathoOrder. For this reason, we created a superclass DiagOrder, which contains the shared characteristics. Figure 11.7 shows the object model of the clinic.

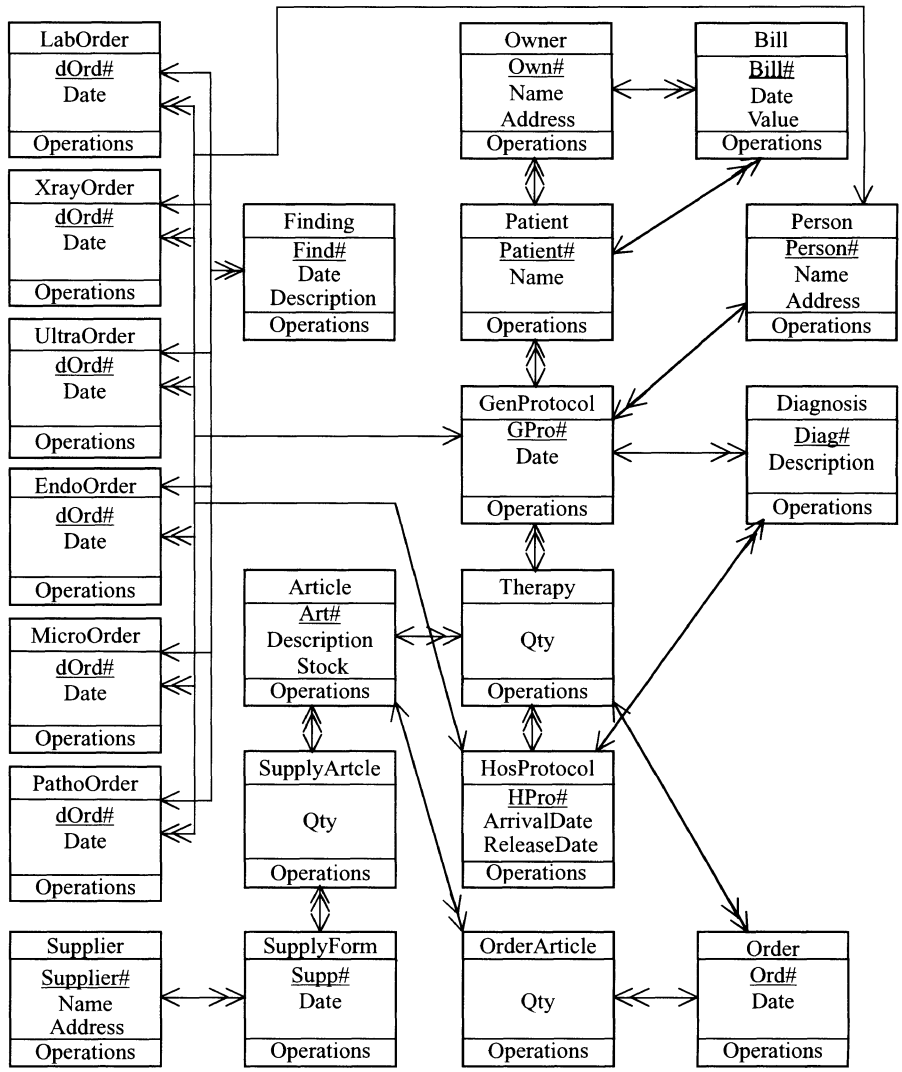


FIGURE 11.6. Initial object model of the clinic

11.5 Design

11.5.1. OPERATIONS

To define the operations of the object model (Figure 11.6), and corresponding to the first step of the fifth phase, we analyzed the task and activity tables (Tables 11.3 and 11.4). For each task defined in the task

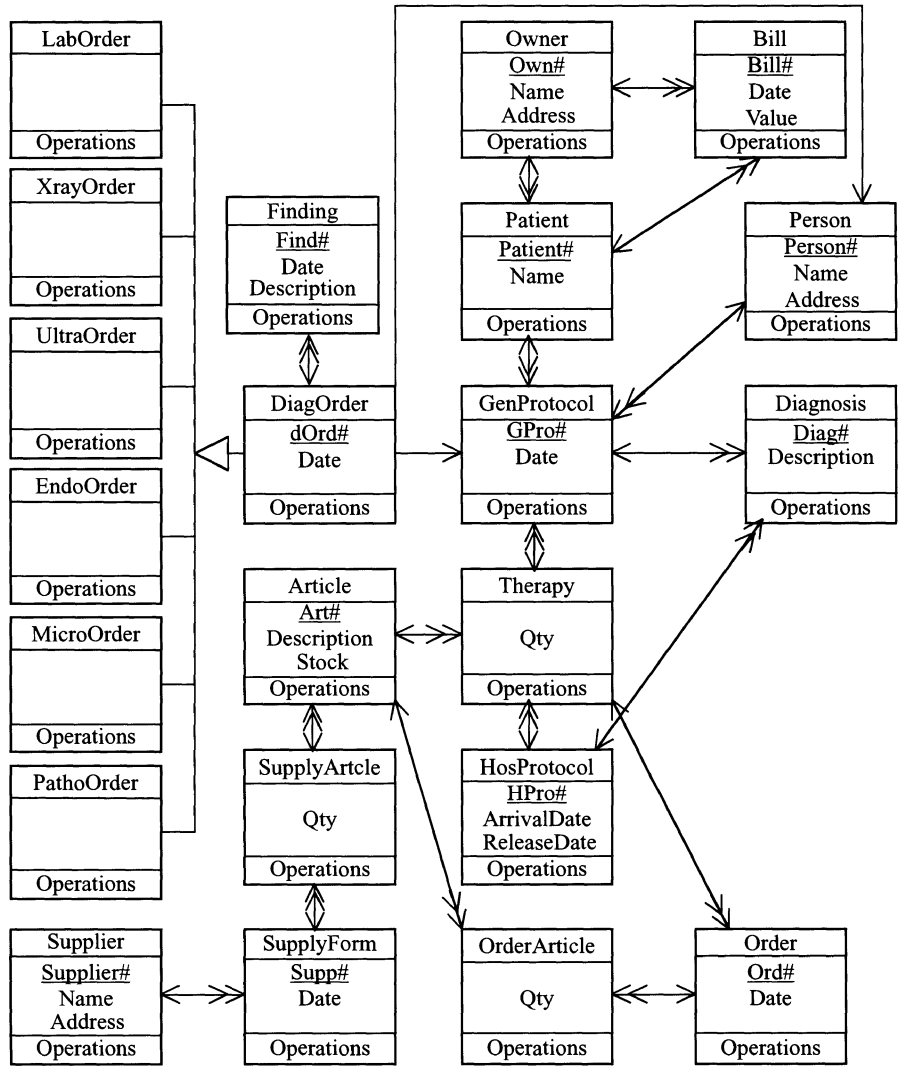


FIGURE 11.7. Object model of the clinic

table, we carefully analyzed in particular the contents of the Description column and also considered the task square in the activity table.

Using the results of such analysis, we developed the operation table. Table 11.5 shows the operation table of the clinic. The table contains 12 operations: AccOwnData, AccPatData, CreateBill, CreGprotocol, CreDiagOrd, AccFinding, PresTherapy, PointHospital, CreHProtocol, CreateOrder, CreEndoFind and CreUltraFind.

Table 11.5. Operation table of the clinic

Task Code	Operation	Class
$K_{1,1}$	AccOwnData	Owner
	AccPatData	Patient
$K_{2,1}$	CreateBill	Bill
$K_{3,2}$	CreGProtocol	GenProtocol
$K_{4,2}$	CreDiagOrd	DiagOrder
$K_{5,2}$	AccFinding	Finding
$K_{6,2}$	PresTherapy	Therapy
$K_{7,2}$	PointHospital	GenProtocol
$K_{7,7}$	CreHProtocol	HosProtocol
$K_{8,2}$	CreateOrder	Order OrderArticle
$K_{9,3}$	CreEndoFind	Finding
$K_{10,4}$	CreUltraFind	Finding

The reader may notice that the operation table (Table 11.5) introduces only the tasks indicated in the task table (11.3).

After completing the operation table, we registered the defined operations in the object model. Figure 11.8 shows the object model of the clinic extended with the identified operations.

We had to complete the first step by writing an algorithm for every defined operation. To write an algorithm, we considered the Description column of the task table, the operation table and the object model. These algorithms may be developed using any object-oriented programming language.

11.5.2. APPLICATION MODEL

The second step of the fifth phase requires development of the application model. This model is developed using the information registered in the entity and activity tables. Figure 11.9 shows the application model of the clinic. This model consists of three parts.

The first part was derived from the entity table. For this reason, this part defines 10 vital analyses and one decision support problem in accordance with the contents of the entity table (Table 11.1). The second part was also derived from the entity table and represents a decision support problem.

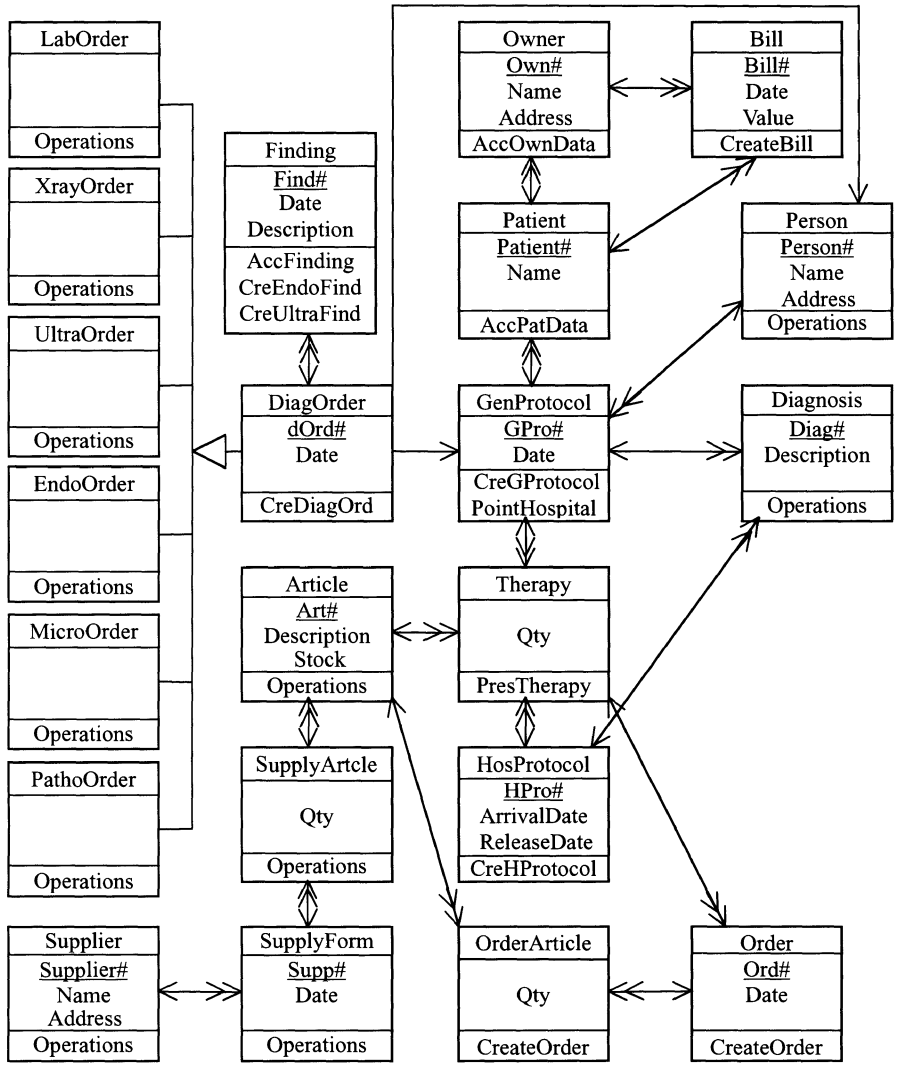


FIGURE 11.8. Object model of the clinic

The third and fourth parts were derived from the activity tables. The second part introduces the business process "Patient." This part consists of eight work processes and their activities: Reception, Treatment, Endoscopy, Ultrasound, X-ray, Laboratory, Hospital and Surgery. The fourth part represents the business process "Medicine & Nutrition." This part includes only one work process, Pharmacy.

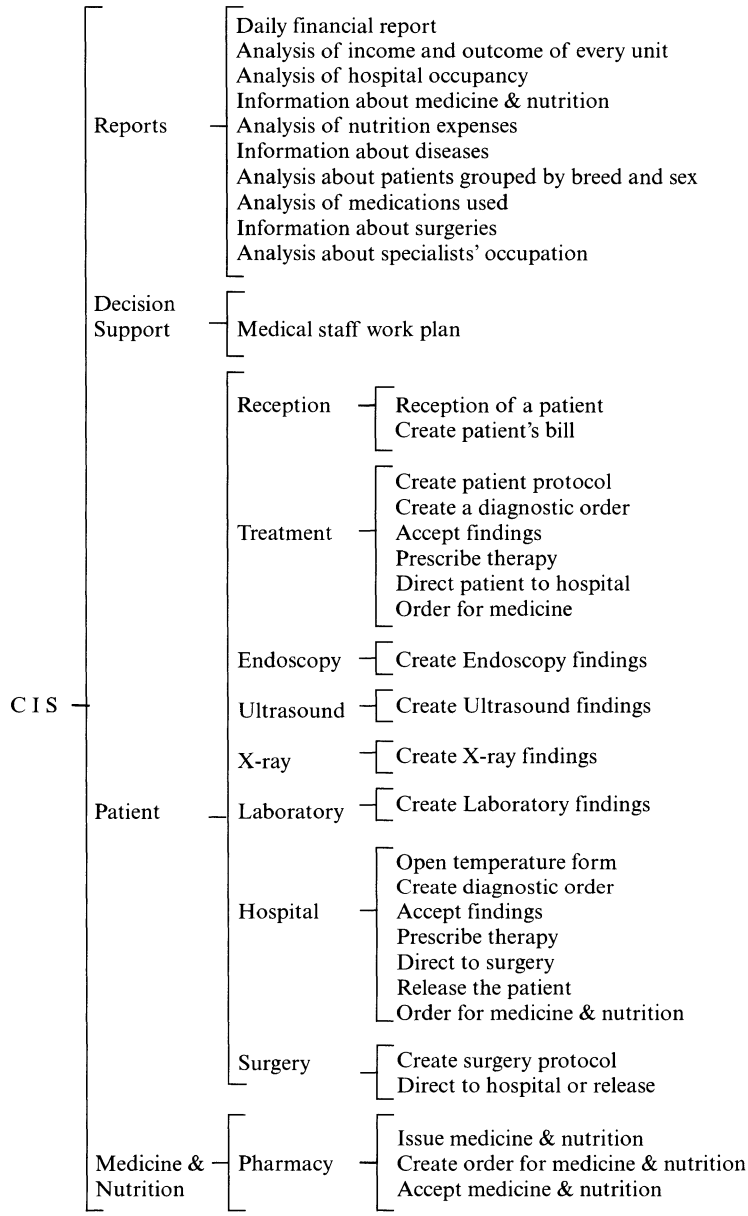


FIGURE 11.9. Application model of the clinic

CHAPTER 12

Payment Movement

The current chapter discusses the development of an information system, which deals with a bank's payment movement with customers. Payment movement is a very wide and complex process. For this reason, we cannot introduce it as a whole in this book. We will treat only a few of its primary functions, such as

- opening an account,
- withdrawing and depositing of a sum of money,
- making cheque payments,†
- making credit card payments and
- closing an account.

In addition to the above-listed functions, there are a number of other primary and supporting functions. Primary functions are made directly with customers, whereas supporting ones are performed internally in the bank.

The reader may notice that this application is very much reduced, showing only the above-mentioned functions in a brief manner and ignoring others because of the space limitations.

12.1 Problem Definition

To implement the first phase of TAD methodology, we organized different interviews with various employees in the bank. The purpose of these interviews was to develop the entity table. Before doing that, let us give a short description of each of the listed functions.

The first relation between the bank and a customer is usually opening an account. For this purpose, the customer has to fill in a form. After

†Note the British spelling of "cheque." It is used here to match the original application.

T. Damij, *Tabular Application Development for Information Systems*

Table 12.1. Entity table of payment movement

Requirements	Entity Analysis	Management	Controller	Cashier
Reports	1. Information about customers	*	*	*
	2. Information about accounts	*	*	*
	3. Information about interest rates	*		
	4. Information about operation of the bank	*		
	5. Information about cash transactions	*	*	
	6. Information about cheque transactions	*	*	
	7. Information about credit card transactions	*	*	
Decision Support Problem	8. Implementation of the bank's strategy	*		

checking the customer's personal data, the cashier opens the account and defines the conditions related to the opened account.

A transaction of cash withdrawal or deposit of cash or cheque(s) can be made directly in the bank or through an automatic teller machine. To perform this transaction, the customer has to have an account. Customers are informed about their transactions once a month.

Customers can make payments using cheques. Cheques can be used inside or outside the bank. Customers are also informed about the balance of their account and the cheques used monthly. Customers can also make various payments using their credit cards. The bank has to endorse or refuse every such transaction after checking the balance of the customer's account.

The customer can decide to close the account. In this case he or she has to fill out a special form and the cashier has to check the balance of the customer's account. If there is a positive balance, the bank pays it to the customer on closing the account. If the balance is negative, the customer must pay the sum due before the bank closes the account.

Table 12.1 shows the entity table of the bank concerning payment movement only. Table 12.1 introduces seven reports and one decision support problem.

12.2 Systems Functioning

12.2.1. Activity Table Development

Corresponding with the second phase of TAD methodology, we organized interviews with entities such as Management, Controller and

Cashier. The aim of these interviews was to create the activity and task tables.

Table 12.2 introduces the activity table of payment movement. The table includes only 20 activities and 5 entities. The first three entities are internal and the last two are external. Table 12.2 shows a reduced table of the real activity table of payment movement.

The first activity, "Accept an account opening form," indicates that the cashier accepts a completed form from a customer for opening an account. Therefore, we write S_4 in square (1,4) and T_4 in square (1,3).

The second activity, "Get approval for account opening," means that the cashier has to get approval from the controller to open an account. To mark this we write S_3 in square(2,3), T_3 in square(2,2), S_2 in square(2,2) and T_2 in square(2,3).

Considering the entity Cashier column, we found that the first activity is the predecessor to the second one. To indicate this we write P_1 in square(1,3) and U_1 in squares (2,3). Furthermore to indicate that the fourth activity, "Open customer's account," is the predecessor to the tenth, fourteenth and eighteenth activities, we write P_4 in square(4,3) and U_4 in square(10,3), square(14,3) and square(18,3).

The same concept is used to develop horizontal and vertical linkages between activities.

12.2.2. TASK TABLE DEVELOPMENT

In accordance with the second step of the second phase we tried to create the task table of payment movement (Table 12.3). This table was developed simultaneously with the activity table. Table 12.3 represents the task table of payment movement. It shows only the first seven activities and their tasks as they are indicated in the activity table.

The first activity has only one task, $K_{1,3}$. The second activity includes two tasks: $K_{2,2}$ and $K_{2,3}$, which are given in the second and third rows of the task table. The third activity, introduced in the fourth row of the task table, has one task: $K_{3,3}$. The fifth row of Table 12.3 shows the fourth activity, which has one task, $K_{4,3}$. The fifth, sixth and seventh activities are shown in the sixth, seventh and eighth rows of the table. Each of these activities has only one task; these are $K_{5,3}$, $K_{6,3}$ and $K_{7,3}$.

Other activities of Table 12.2 are not described in Table 12.3 because of space limitations.

12.2.3. WORK PROCESSES AND BUSINESS PROCESSES

As a result of the joint meeting with the manager of payment movement, the controller and the cashier, we extended the activity table by defining work processes and business processes. Table 12.4 shows the completed

Table 12.2. Activity table of payment movement

Activity \ Entity	1. Management	2. Controller	3. Cashier	4. Customer	5. Other Organization
1. Accept an account opening form			T_4 P_1	S_4	
2. Get approval for account opening		T_3,S_2	S_3,T_2 U_1,P_2		
3. Sign a contract			S_3,T_4 U_2,P_3	T_3,S_4	
4. Open customer's account			S_3 U_3,P_4	T_3	
5. Accept a withdrawal or deposit form			T_4 U_4,P_5	S_4	
6. Check account's balance			S_3 U_5,P_6		
7. Calculate bank's margin			S_3 U_6,P_7		
8. Perform withdrawal or deposit transaction			S_3 U_7,P_8	T_3	
9. Issue a confirmation			S_3 U_8	T_3	
10. Accept customer's cheque			T_4,T_5 U_4,P_{10}	S_4	S_5
11. Check cheque and account balance			S_3 U_{10},P_{11}		
12. Calculate bank's margin			S_3 U_{11},P_{12}		
13. Confirm transaction			S_3 U_{12}		
14. Accept credit card transaction			T_4,T_5 U_4,P_{14}	S_4	S_5
15. Check account's balance			S_3 U_{14},P_{15}		
16. Calculate bank's margin			S_3 U_{15},P_{16}		
17. Confirm transaction			S_3 U_{16}		
18. Accept account closing form			T_4 U_4,P_{18}	S_4	
19. Check account's balance			S_3 U_{18},P_{19}		
20. Close customer's account			S_3 U_{19}	T_3	

Table 12.3. Task table of payment movement

Activity	Characteristic Task Code	Description	Time	Condition	Input/Output
1. Accept an account opening form	$K_{1,3}$	Cashier accepts a completed form from a customer to open an account			Opening form
2. Get approval for account opening	$K_{2,3}$	Cashier sends an account opening form to controller for approval		Check customer's personal data	Opening form
	$K_{2,2}$	Controller approves opening an account and sends opening form back to cashier		Check customer's history	Opening form
3. Sign a contract	$K_{3,3}$	Cashier creates a contract and gives it to customer for signature			Contract
4. Open customer's account	$K_{4,3}$	Cashier opens a customer's account			Opening form
5. Accept a withdrawal or deposit form	$K_{5,3}$	Cashier accepts a withdrawal or deposit form from customer		Withdrawal or deposit form	Withdrawal or deposit form
6. Check account's balance	$K_{6,3}$	Cashier checks customer's account balance			Withdrawal or deposit form
7. Calculate bank's margin	$K_{7,3}$	Cashier calculates bank's margin		Check account's balance	Withdrawal or deposit form

activity table of payment movement. The table contains 20 activities, which are grouped into 5 work processes.

The first work process, "Account opening," contains four activities, which discuss the approach of opening an account in the bank. "Cash transaction" is the second work process; it includes five activities and deals with performing a withdrawal or deposit of a sum of money.

The third work process, "Cheque transaction," groups four activities and introduces the steps of performing financial transactions using cheques. The fourth work process, "Card transaction," has four activities and discusses the possibility of using credit cards to perform financial transactions. "Account closure" is the fifth work process; it implements the customer's wish to close his or her account.

These five work processes are grouped into one business process called "Payment Movement"; see Table 12.4.

Table 12.4. Activity table of payment movement

Business Process	Work Process	Entity / Activity	1. Management	2. Controller	3. Cashier	4. Customer	5. Other Organization
Payment Movement	Account opening	1. Accept an account opening form			T_4 P_1	S_4	
		2. Get approval for account opening		T_3,S_2	S_3,T_2 U_1,P_2		
		3. Sign a contract			S_3,T_4 U_2,P_3	T_3,S_4	
		4. Open customer's account			S_3 U_3,P_4	T_3	
	Cash transaction	5. Accept a withdrawal or deposit form			T_4 U_4,P_5	S_4	
		6. Check account's balance			S_3,U_5,P_6		
		7. Calculate bank's margin			S_3 U_6,P_7		
		8. Perform withdrawal or deposit transaction			S_3 U_7,P_8	T_3	
		9. Issue a confirmation			S_3 U_8	T_3	
	Cheque transaction	10. Accept customer's cheque			T_4,T_5 U_4,P_{10}	S_4	S_5
		11. Check cheque and account balance			S_3 U_{10},P_{11}		
		12. Calculate bank's margin			S_3 U_{11},P_{12}		
		13. Confirm transaction			S_3 U_{12}		
	Card transaction	14. Accept card transaction			T_4,T_5 U_4,P_{14}	S_4 S_5	
		15. Check account's balance			S_3 U_{14},P_{15}		
		16. Calculate bank's margin			S_3 U_{15},P_{16}		
		17. Confirm transaction			S_3 U_{16}		
	Account closure	18. Accept account closing form			T_4 U_4,P_{18}	S_4	
		19. Check account's balance			S_3 U_{18},P_{19}		
		20. Close customer's account			S_3 U_{19}	T_3	

12.3 Business Process Reengineering

This application discusses only a small segment of payment movement in the bank. For this reason, we could not use the first step of the third phase, which discusses the possibility of carrying out BPR. Thus we continued with the second step of the third phase, which develops the process model of the system.

12.3.1. Process Model

Figure 12.1 shows the process model of payment movement, which was created corresponding to the procedure described in Section 6.2. The diagram shown represents the elementary-level DFD of payment movement, which consists of five parts.

The first part represents the work process "Account opening," which is connected to the first four activities of the activity table (Table 12.4). The second part introduces the work process "Cash transaction." This part contains five processes related to five activities represented in rows 5 to 9 of the activity table.

The third part shows the work process "Cheque transaction," which has four processes. These processes are connected to four activities given in rows 10 to 13 of Table 12.4. The fourth and fifth parts show two work processes; these are "Card transaction" and "Account closure." The fourth part includes four processes, which represent the four activities defined in rows 14 to 17. The fifth part contains three processes, which are related to the last three activities of Table 12.4.

Figure 12.2 represents the activity-level DFD of payment movement. This diagram includes 20 processes related to the 20 activities of Table 12.4.

The reader may notice that the activity-level DFD of payment movement is very similar to the elementary-level DFD. This is because almost each activity of the activity table has only one task.

Figure 12.3 shows the work-process–level DFD. This diagram includes five processes: "Account opening," "Cash transaction," "Cheque transaction," "Card transaction" and "Account closure." These processes are defined by grouping appropriate activity-level processes of the activity-level DFD.

Figure 12.4 introduces the business-process–level DFD of payment movement. This DFD is also the context diagram of the system because the system has only one business process: "Payment Movement."

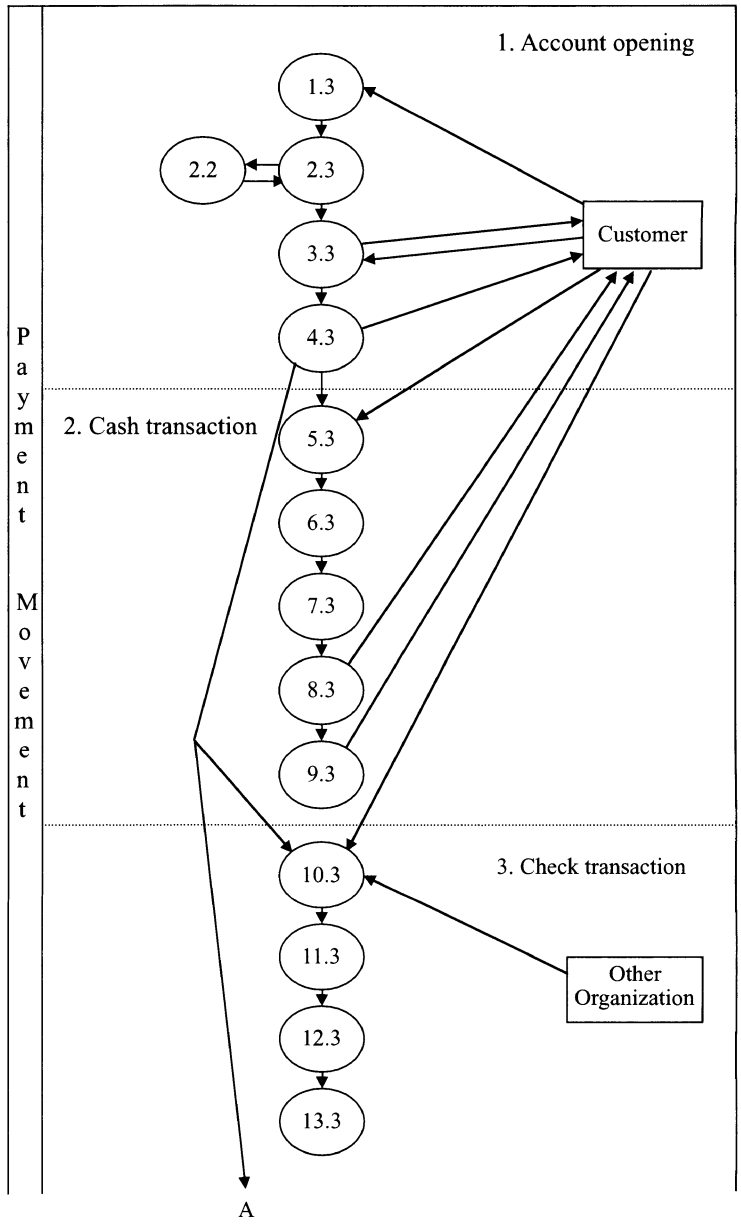


FIGURE 12.1. Elementary-level DFD of payment movement

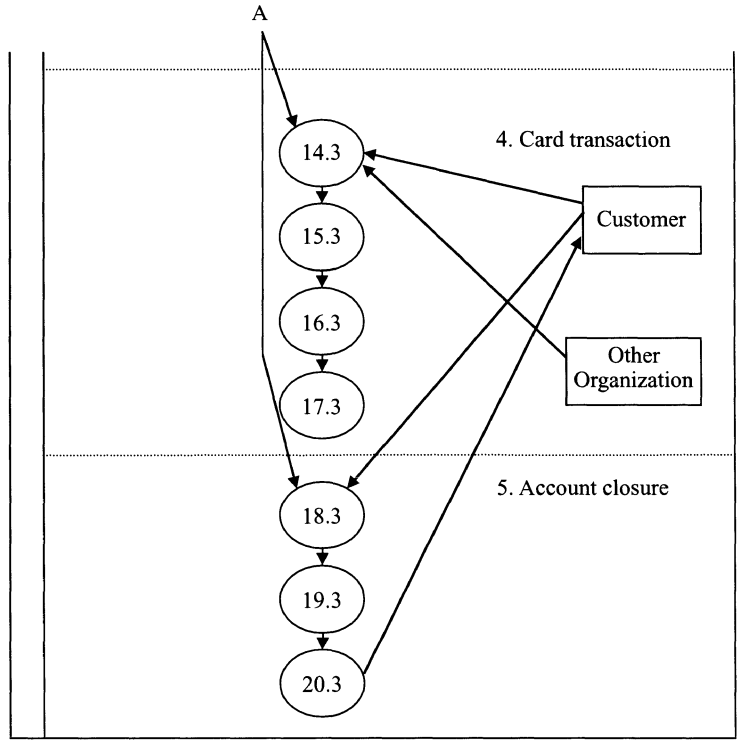


FIGURE 12.1. (Continued) Elementary-level DFD of payment movement

12.4 Object Model

Figure 12.5 shows the initial object model of payment movement, which was developed in accordance with the steps of the fourth phase given in Section 7.3.

This initial object model consists of nine object classes and one association class. The object classes are Bank, Customer, Contract, Account, Cashier, Card, CashTran, ChequeTran and Organization. The association class is CardTran.

Using the last step of the fourth phase, we tried to implement the inheritance between the classes of the initial object model. We found that classes. CashTran, ChequeTran and CardTran share more characteristics. For this reason, a superclass Transaction was created to contain these common characteristics. Figure 12.6 shows the final object model of payment movement.

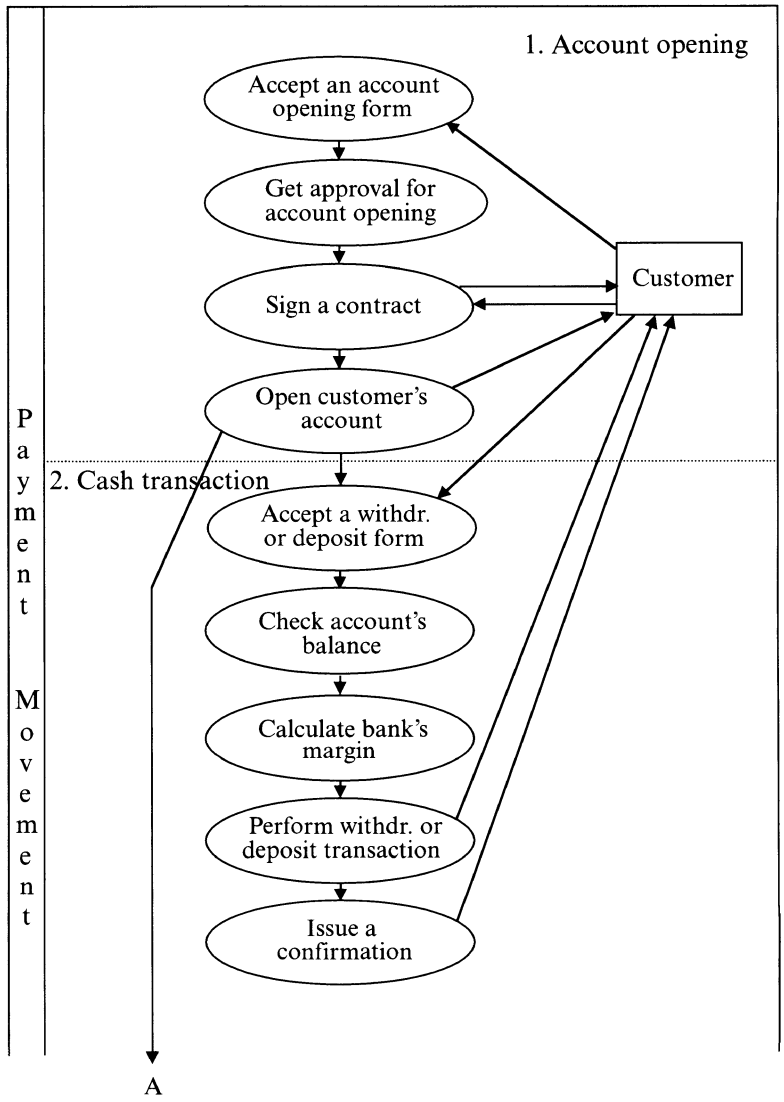


FIGURE 12.2. Activity-level DFD of payment movement

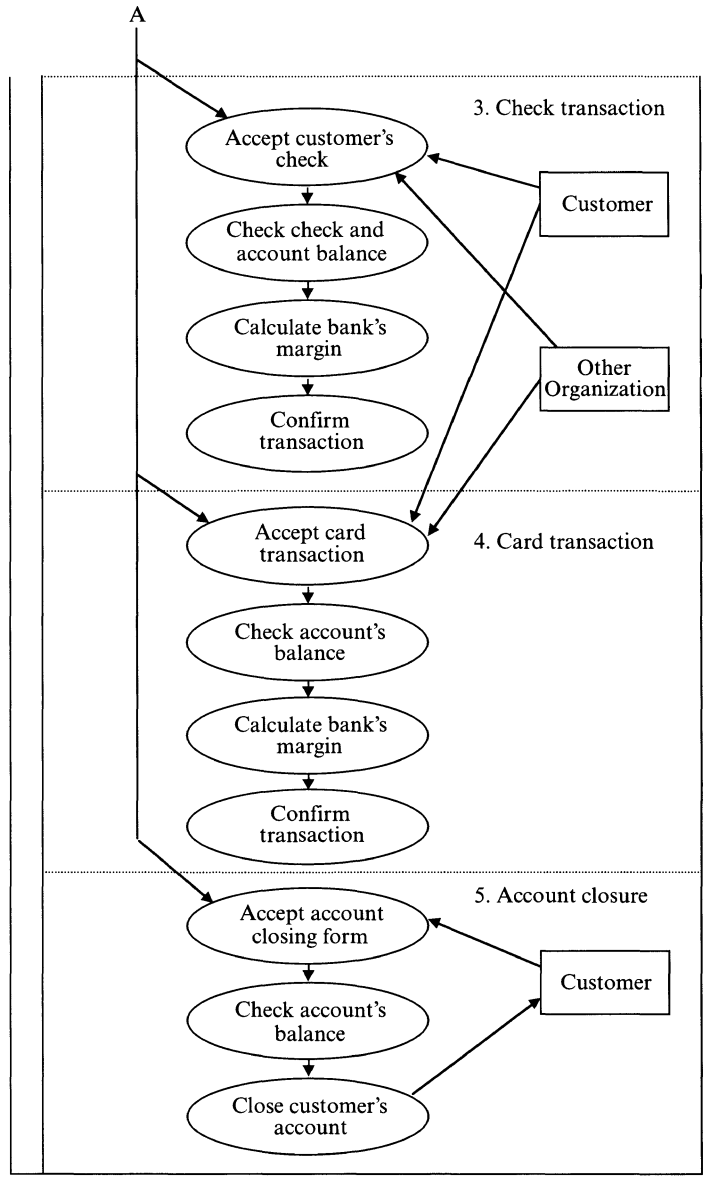


FIGURE 12.2. (Continued) Activity-level DFD of payment movement

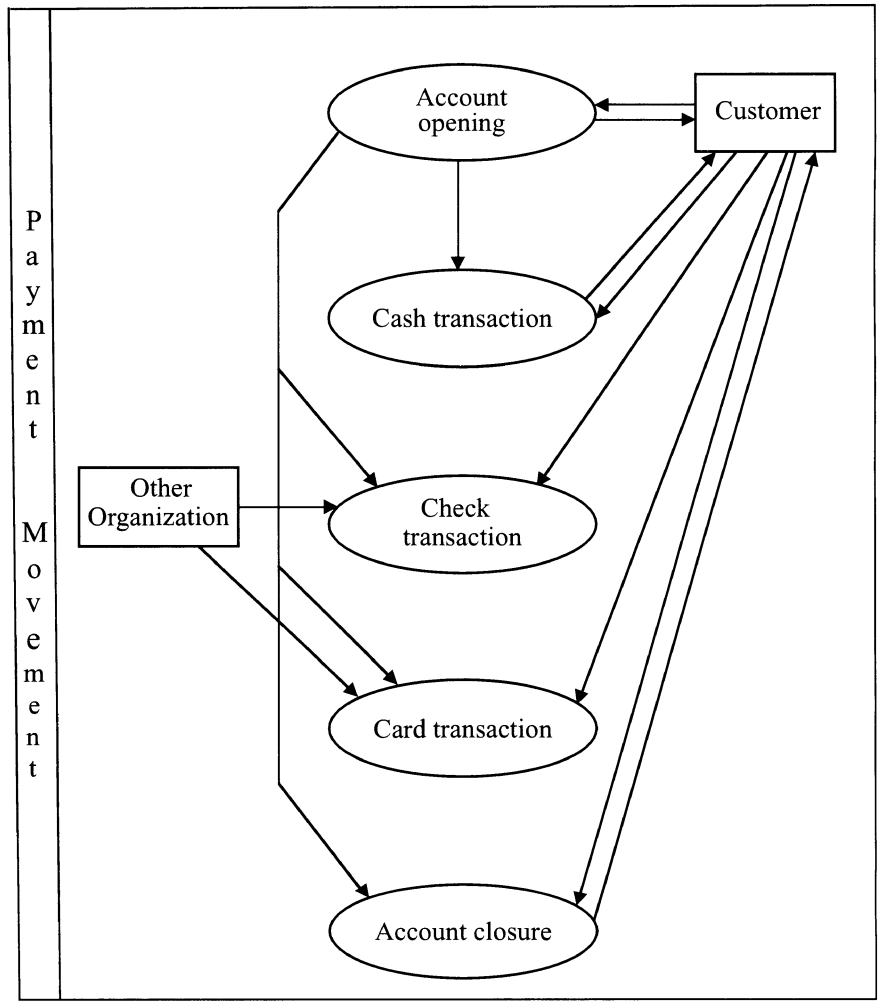


FIGURE 12.3. Work-process–level DFD

12.5 Design

12.5.1. OPERATIONS

To define a linkage between the functioning and the data structure of the system, which is prescribed by the first step of the fifth phase, we create the operation table. This table enables us to connect the activity and the task tables (Table 12.4 and 12.3) with the object model (Figure 12.6).

Table 12.5 represents only those operations that are extracted from the descriptions of the tasks listed in the task table. The operation table

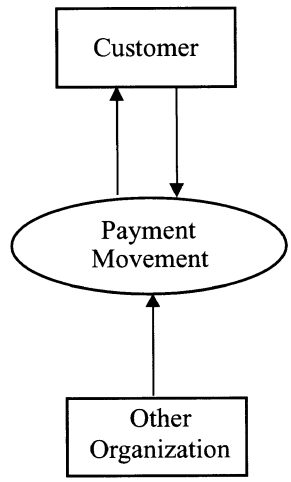


FIGURE 12.4. Business-process–level DFD (context diagram)

Table 12.5. Operation table of payment movement

Task Code	Operation	Class
$K_{1,3}$	CreateCashier	Cashier
	CreateCust	Customer
$K_{2,3}$	CheckCust	Controller
$K_{2,2}$	Approval	Controller
$K_{3,3}$	CreContract	Contract
$K_{4,3}$	OpenAccount	Account
$K_{5,3}$	AcceptTran	Transaction
$K_{6,3}$	CheckTran	Transaction
$K_{7,3}$	CalculateProv	Account

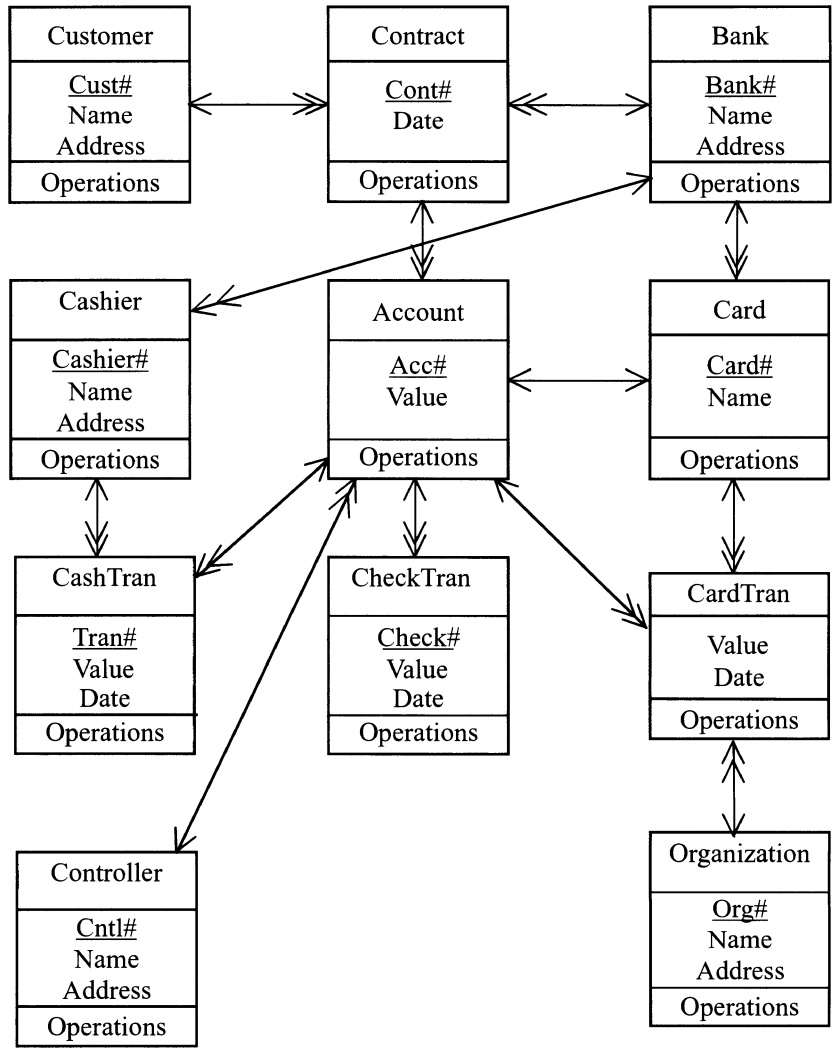


FIGURE 12.5. Initial object model of payment movement

shows nine operations connected with eight tasks and seven classes. The operations CreateCashier, CreateCust, CheckCust, Approval, CreContract and OpenAccount are connected with the method of opening an account. The operations AcceptTran, CheckTran and CalculateProv are related to performing a transaction.

Figure 12.7 introduces the object model of payment movement, which includes only the operations defined in the operation table (Table 12.5).

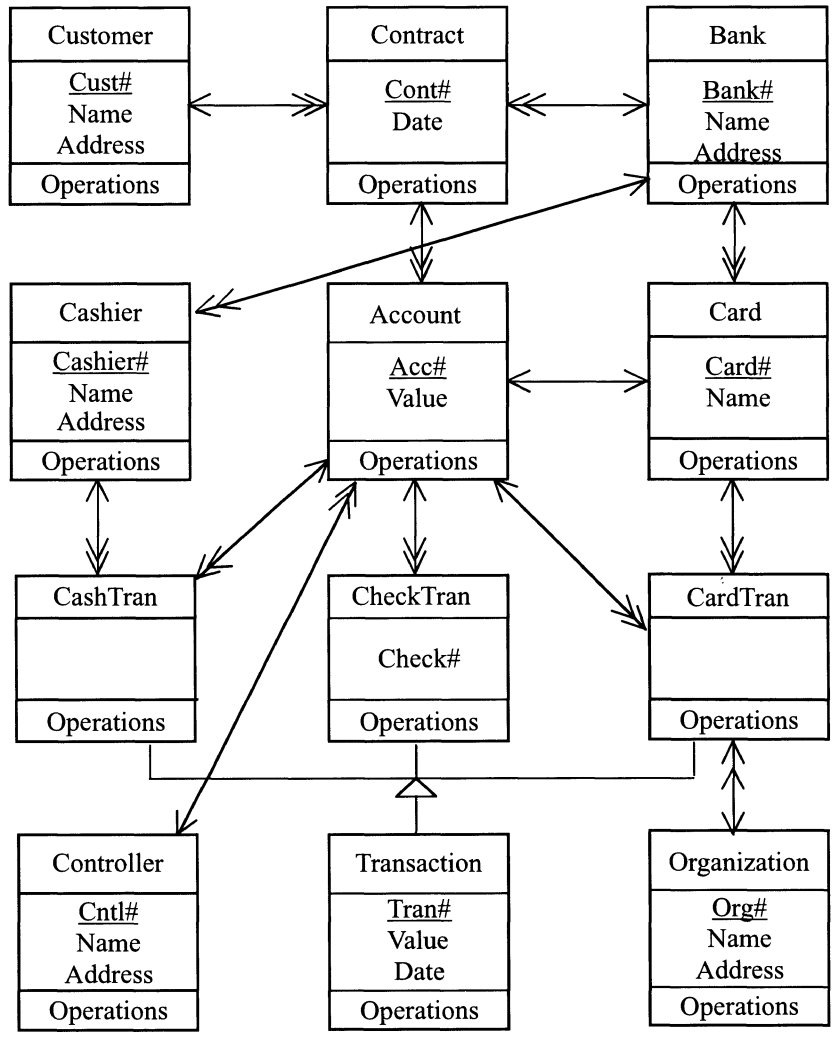


FIGURE 12.6. Object model of payment movement

Corresponding to the first step of the fifth phase, we have to write algorithms for the operations identified in the operation table. To do this, we usually use an object-oriented programming language.

12.5.2. APPLICATION MODEL

The second step of the fifth phase prescribes the development of the system's application model. In accordance with this step, we developed the application model of payment movement.

Figure 12.8 shows this model, which consists of three parts. The first

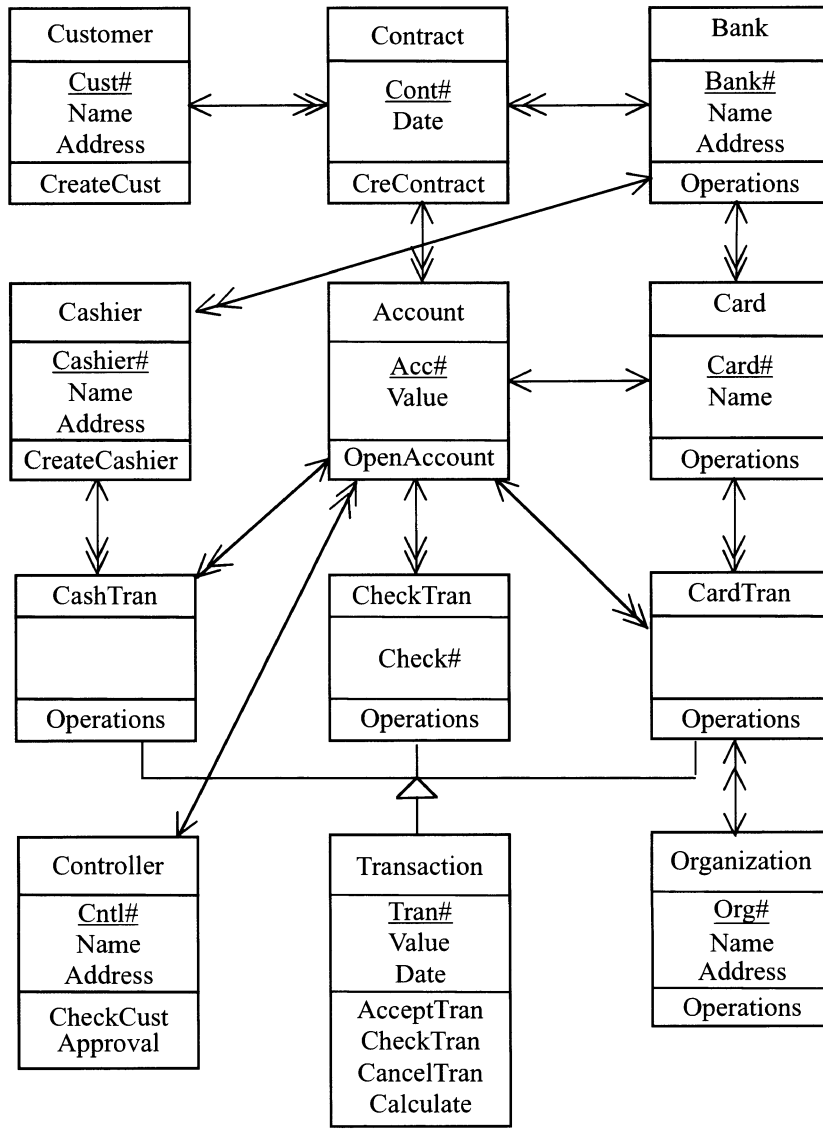


FIGURE 12.7. Object model of payment movement

two are derived from the entity table. The third part is developed from the activity table.

The first part introduces seven reports, which provide the management with information about customers' accounts and customers' transactions. The second part represents a decision support problem, which tries to link the functioning of the system with the strategy of the bank.

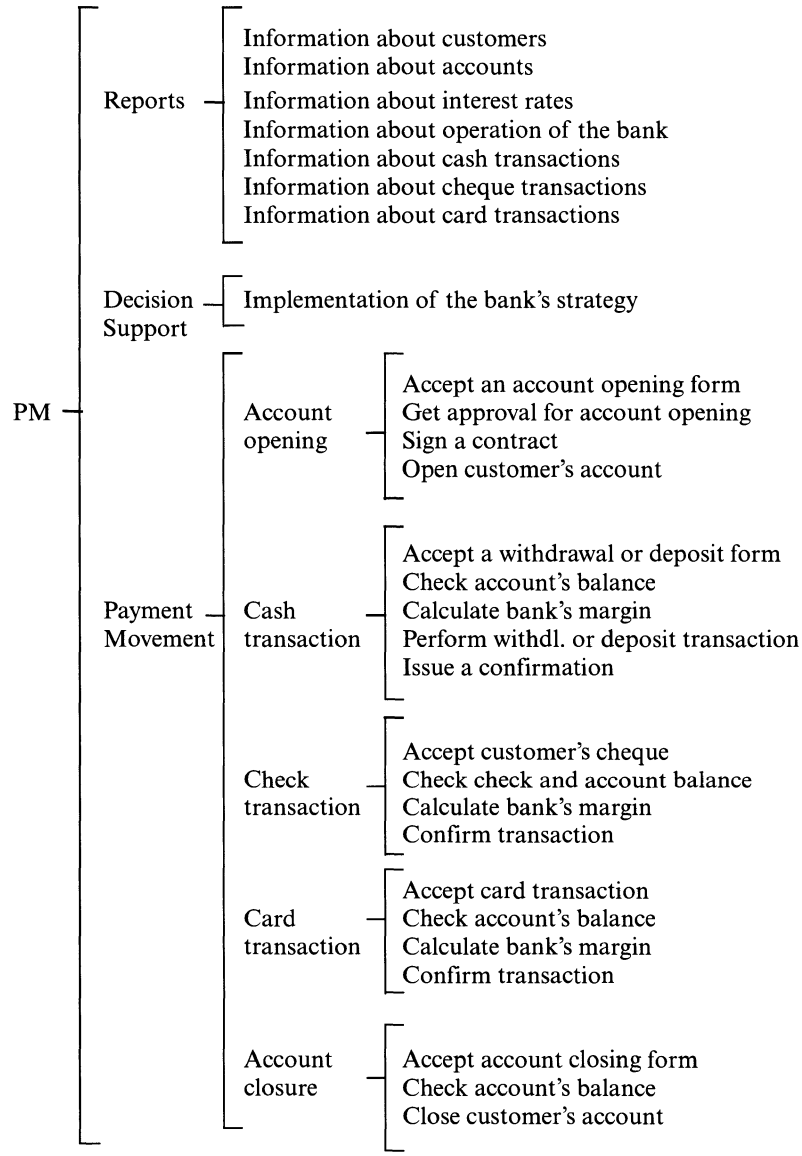


FIGURE 12.8. Application model of the payment movement

The third part shows the business process "Payment Movement," which consists of five work processes "Account opening," "Cash transaction," "Cheque transaction," "Card transaction" and "Account closure." Each of these work processes includes a number of activities.

PART IV

TAD CASE-TOOL

The fourth part of the book contains only one chapter; this is Chapter 14, which explains how to use TAD case-tool version 1.0.

CHAPTER 13

Student Office

The fifth application deals with the development of an information system for a student office.

The student office studied here has approximately 10 employees who take care of several thousand students. There are both full-time and part-time students. In addition, there are different study programs. For this reason some of the employees are specialized in particular study programs for full-time and others for part-time study. The student office is led by the office manager, who ensures that the functioning of the office is in accordance with the rules of the faculty. The office manager is responsible to the vice-dean in charge of curricular matters.

13.1 Problem Definition

We started the process of interviews with the vice-dean and then continued with the office manager. We asked the Vice-Dean to describe his plan and goals about the functioning of the student office. Furthermore, he was asked to define the analyses and requirements that he expects to get from the student office manager.

After a meeting with the vice-dean, we interviewed the office manager and discussed with her the analyses required by the vice-dean. We also asked her to define her requirements and decision support problems that are important for successfully carrying out the work of the office.

The results of the interviews are collected in Table 13.1. This table represents the entity table of the student office and contains two entities, six analyses, and one decision support problem. The required analyses are mainly related to information about enrollment, student graduation, and the professors' schedules. As Table 13.1 shows, the defined decision support problem deals with preparing the study schedule for each semester.

T. Damij, *Tabular Application Development for Information Systems*

Table 13.1. Entity table of student office

Requirements	Analysis \ Entity	Vice-Dean	Office Manager
Reports	1. Information about enrolled students	*	*
	2. Analysis of students' success at the end of first study year	*	*
	3. Analysis of students' success sorted by subjects and professors	*	*
	4. Information about graduation topics and defenses	*	*
	5. Analysis of student generations	*	*
	6. Information about professors	*	*
Decision Support Problem	7. Preparing a study schedule at the beginning of the study year	*	*

13.2 Systems Functioning

13.2.1. Activity Table Development

To create the activity table of the student office, we first interviewed the office manager. She explained the kind of work they deal with and introduced the employees of the office and their responsibilities. At this meeting we registered in the activity table every event of her daily work.

We then continued our interviews with each employee separately and registered their work and the connections among them.

Table 13.2 represents the activity table of the student office. This table contains 18 activities and 7 entities. The first three entities—Vice-Dean, Office Manager and Employee—are internal and the last four—Student, Professor, Other Department and University—are external. The entity Employee in Table 13.2 represents 10 employees who deal with almost the same activities.

The first activity, "Get list of first-year students," comes from the university to the office manager. For this reason, we write S_7 in square (1,7) and T_7 in square (1,2). The office manager gives a copy of it to the vice-dean and also to each employee. This is why we write S_2 in square (1,2) and T_2 in squares (1,1) and (1,3).

Considering the entity Employee, we find that the first activity, "Get list of first-year students," is the predecessor to the second and fourth activities. Therefore, we write P_1 in square(1,3) and U_1 in squares (2,3) and (4,3). Furthermore, the third activity, "Check study programs," is the predecessor to the fourth, and for this reason, we write P_3 in square(3,3) and U_3 in square(4,3).

Table 13.2. Activity table of student office

Activity \ Entity	1. Vice-Dean	2. Office Manager	3. Employee	4. Student	5. Professor	6. Other Dept.	7. University
1. Get list of first-year students	T_2	T_7,S_2	T_2 P_1				S_7
2. Inform students about enrollment			S_3 U_1	T_3			
3. Check study programs			S_3 P_3				
4. Enroll students			T_4 U_1,U_3,P_4	S_4			
5. Confirm enrollment			S_3 U_4	T_3			
6. Inform professors about enrollment	T_2	T_3,S_2	S_3 U_4		T_2		
7. Accept lecture schedule			T_6 U_3,P_7			S_6	
8. Enroll students for exams			T_4 U_3,U_7,P_8	S_4			
9. Create a list of enrolled students			S_3 U_8,P_9		T_3		
10. Announce the exam results			T_5,S_3 U_9,P_{10}	T_3	S_5		
11. Register exam results			S_3 U_{10},P_{11}				
12. Enroll students' practicals			T_3 U_3,P_{12}	S_3			
13. Register students' practicals			T_5,T_4 U_{12}	S_4	S_5		
14. Enroll theme of graduation work			T_5,T_4 $U_3U_{11}P_{14}$	S_4	S_5		
15. Confirm graduation work content			T_5 U_{14},P_{15}		S_5		
16. Review graduat. work technically			S_3 U_{15},P_{16}	T_3			
17. Accept graduation work			T_4 U_{16},P_{17}	S_4			
18. Defense of graduation work	T_3		S_3 U_{17}	T_3	T_3		

The same concept is used to develop horizontal and vertical linkages between other activities.

13.2.2. TASK TABLE DEVELOPMENT

Simultaneously with the activity table (Table 13.2), we also developed the task table of the student office (Table 13.3). The task table was

Table 13.3. Task table of student office

Activity	Characteristic Task Code	Description	Time	Condition	Input/Output
1. Get list of first-year students	$K_{1,2}$	The manager receives list of accepted students from the university and gives it to the vice-dean and employee			List
2. Inform students about enrollment	$K_{2,3}$	Employee informs students about their acceptance at the faculty		Check if student is on the list	List
3. Check study programs	$K_{3,3}$	Employee checks study programs			
4. Enroll students	$K_{4,3}$	Employee enrolls students in the first and higher years		For first-year students, check if student is on the list Check if the application is correct	List, Application form
5. Confirm enrollment	$K_{5,3}$	Employee gives student confirmation of enrollment if he or she requires it		Check student's card	Card
6. Inform professors about enrollment	$K_{6,3}$	Employee sends report about enrollment to the office manager		Check all enrollment applications	Application form
	$K_{6,2}$	The manager sends the report about enrollment to vice-dean and professors			

created step by step as a result of more interviews with the internal entities of the student office.

As the reader may notice, the activity table of the student office (Table 13.2) contains 18 activities. We present only a few of these activities in the task table, because the real task table is too large for presentation in this book. Table 13.3 shows the task table of the student office. This table represents only seven tasks, which belong to the first six activities defined in the activity table.

The first rows of Table 13.3 describe the first activity, which has three tasks. These are $K_{1,2}$, performed by the office manager and indicates the acceptance of the list of students from the university and sent to the vice-dean and employees; $K_{1,1}$ and $K_{1,3}$ indicate the acceptance of the list by the vice-dean and employee from the office manager. This information is already contained in task $K_{1,2}$. For this reason, we ignored the definition of the mentioned two tasks in special rows in the task table.

The next three rows describe the tasks $K_{2,3}$, $K_{3,3}$ and $K_{4,3}$, which belong to the second, third and fourth activities. These tasks deal with the enrollment of students in the first or higher years. The last three rows define the tasks $K_{5,3}$, $K_{6,3}$ and $K_{6,2}$, which belong to the fifth and sixth activities. These tasks deal with providing information about the result of enrollment.

Table 13.4. Activity table of student office

Business Process	Work Process	Entity / Activity	1. Vice-Dean	2. Office Manager	3. Employee	4. Student	5. Professor	6. Other Dept.	7. University
Study	1. Enrollment	1. Get list of first-year students	T_2	T_7,S_2	T_2 P_1				S_7
		2. Inform students about enrollment			S_3 U_1	T_3			
		3. Check study programs			S_3 P_3				
		4. Enroll students			T_4 U_1,U_3,P_4	S_4			
		5. Confirm enrollment			S_3 U_4	T_3			
		6. Inform professors about enrollment	T_2	T_3,S_2	S_3 U_4		T_2		
	2. Study	7. Accept lecture schedule			T_6 U_3,P_7			S_6	
		8. Enroll students for exams			T_4 U_3,U_7,P_8	S_4			
		9. Create a list of enrolled students			S_3 U_8,P_9		T_3		
		10. Announce the exam results			T_5,S_3 U_9,P_{10}	T_3	S_5		
		11. Register exam results			S_3 U_{10},P_{11}				
		12. Enroll students' practicals			T_3 U_3,P_{12}	S_3			
		13. Register students' practicals			T_4,S_3,t_5 U_{12}	S_4	T_3,S_5		
	3. Graduation	14. Enroll theme of graduation work			T_4,T_5 $U_3U_{11}P_{14}$	S_4	S_5		
		15. Confirm graduation work content			T_5 U_{14},P_{15}		S_5		
		16. Review graduat. work technically			S_3 U_{15},P_{16}	T_3			
		17. Accept graduation work			T_4 U_{16},P_{17}	S_4			
		18. Defense of graduation work	T_3		S_3 U_{17}	T_3	T_3		

13.2.3. WORK PROCESSES AND BUSINESS PROCESSES

When we finished this cycle of interviews, we organized a joint meeting with the office manager and three employees. This meeting was important in order to remove many misunderstandings that were registered in the activity table. The results of the interviews are given in Table 13.4.

Table 13.4 represents the activity table of the student office. This table has 18 activities, which are grouped into three work processes: "Enrollment," "Study" and "Graduation."

The first work process has six activities and deals with the enrollment of students in the first or higher study years. The second work process contains seven activities and controls the study process throughout the year. The third work process groups the last five activities and deals with graduation.

The three mentioned work processes are grouped into one business process called "Study"; see Table 13.4.

13.3 Business Process Reengineering

The purpose of the fifth application was only to develop an information system for the student office and did not include the possibility of carrying out BPR in the office.

For this reason, we skipped the first step of the third phase and continued our work at the second step, which deals with the development of the system's process model.

13.3.1. PROCESS MODEL

Figure 13.1 represents the process model of the student office, which was developed using the procedure described in Section 6.2. The created DFD introduces one business process, "Study." This elementary-level DFD consists of three parts.

The first part introduces the work process "Enrollment," which was obtained as a result of transforming the first six rows of the activity table (Table 13.4) into this part of the DFD. Figure 13.1 shows that this part contains eight elementary processes and three sources. The processes are obtained by transforming the indicated tasks of internal entities into elementary processes. The sources are obtained by transforming the non-empty squares of external entities.

The second part of the DFD shows the work process "Study." This part was created using the information collected in rows 7 to 13. It contains seven elementary processes and three sources.

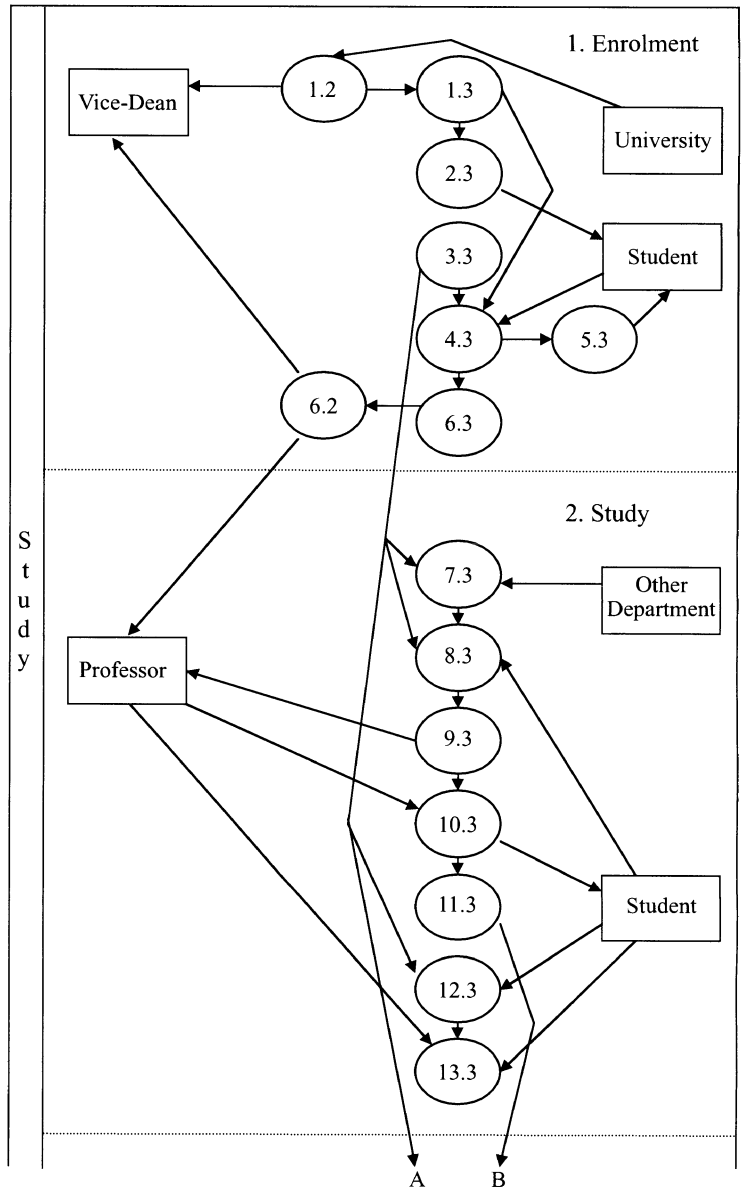


FIGURE 13.1. Elementary-level DFD of student office

The third part introduces the third work process, "Graduation." This part was developed by transforming the information collected in the last five rows of the activity table. The third part has five elementary processes and three sources.

In addition to the elementary-level DFD, we created DFDs at different

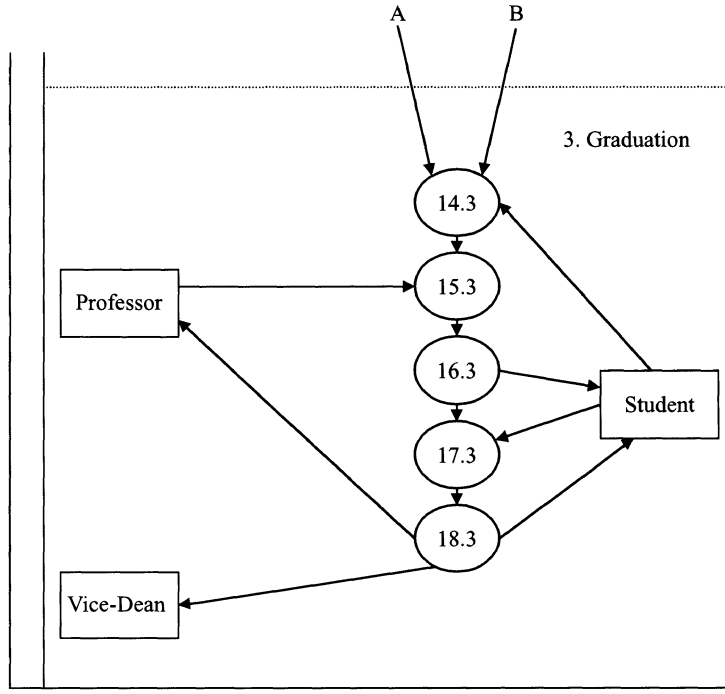


FIGURE 13.1. (Continued) Elementary-level DFD of student office

levels by grouping processes into suitable groups. Figure 13.2 shows the activity-level DFD, which contains 18 activity-level processes. Each of them was obtained by grouping elementary processes that belong to a certain activity into a single process.

Figure 13.3 represents the work-process–level DFD, which includes three processes: "Enrollment," "Study" and "Graduation." Each of them was created by grouping the activity-level processes that belong to a determined work process into a single process.

Figure 13.4 shows the context diagram of the system or business-process–level DFD. This DFD contains only one process, the business process "Study."

13.4 Object Model

Corresponding to the described steps of the fourth phase, we used the procedure introduced in Section 7.3 to develop the object model of the student office.

To do this we analyzed the documents defined in the Input/Output column of the task table and identified the identity attribute existing in

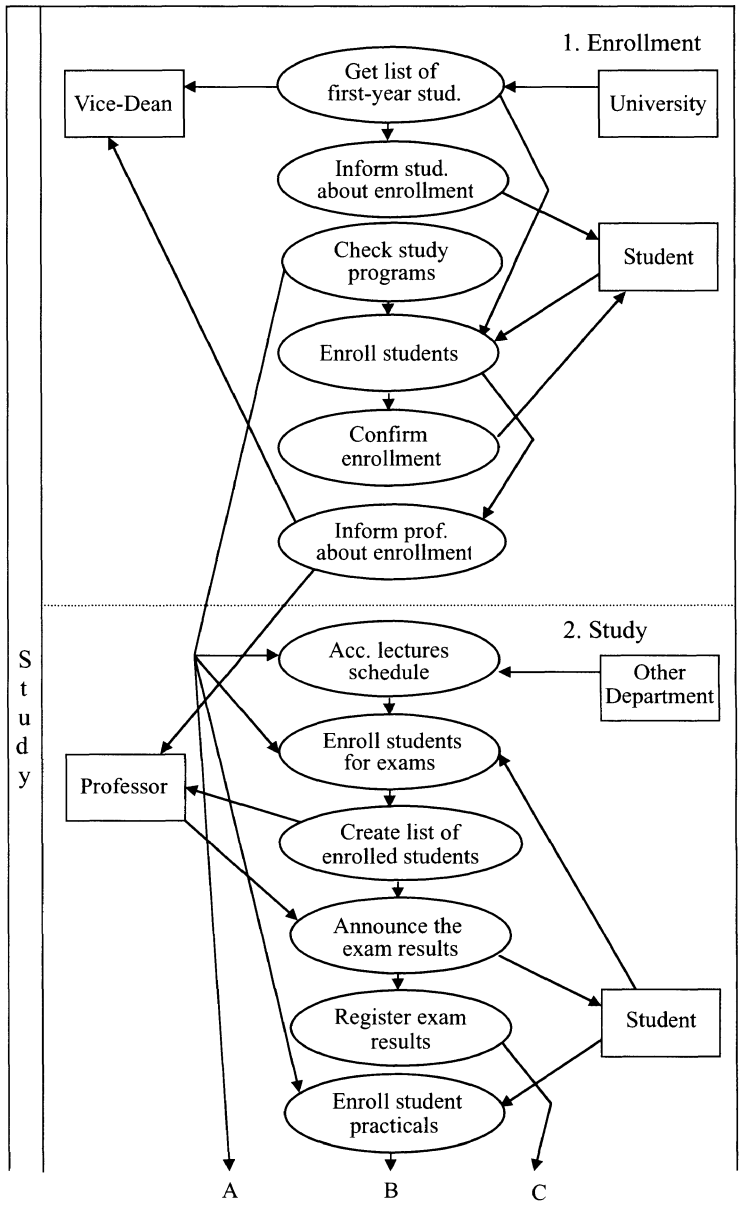


FIGURE 13.2. Activity-level DFD of student office

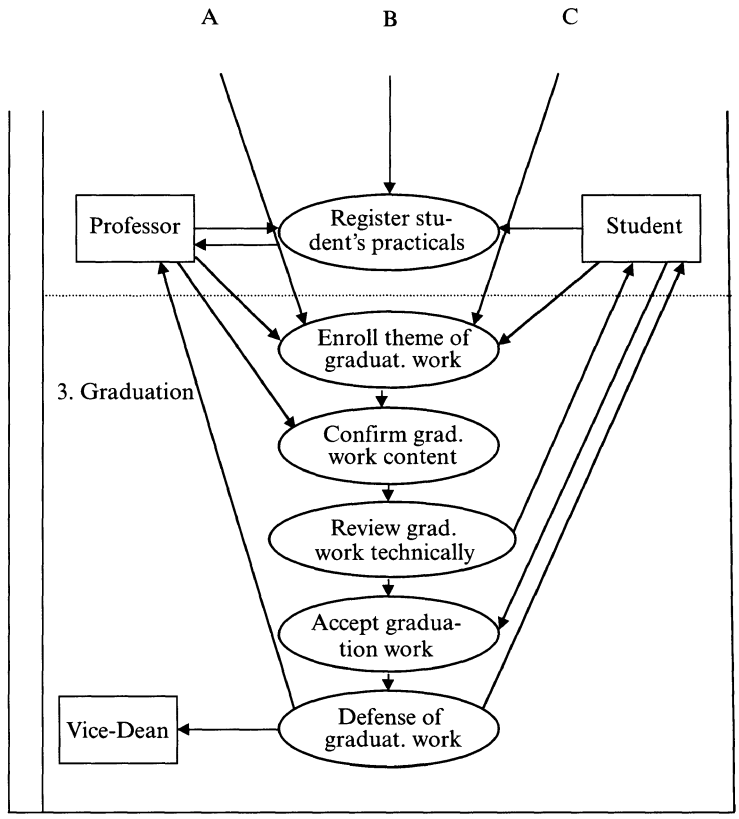


FIGURE 13.2. (Continued) Activity-level DFD of student office

the framework of each document. Furthermore, we defined the functional dependencies on identity attributes and also determined the associations existing among the identity attributes. At the end we transformed these relationships into convenient structures of semantic network modeling, which led us to the development of the initial object model of the student office.

In addition, we analyzed the three entity, activity and task tables of the student office, which were developed in the previous phases. We carefully analyzed each description defined in the Description column of the task table (Table 13.3).

The result of this work was the initial object model of the student office. Figure 13.5 shows this initial object model, which consists of eight object classes and five association classes. The object classes are Student, Town, Country, Program, Specialization, Subject, Payment and Professor. The association classes are StudSubject, Exam, SpecSubject, SpecStudent and ProfSubject.

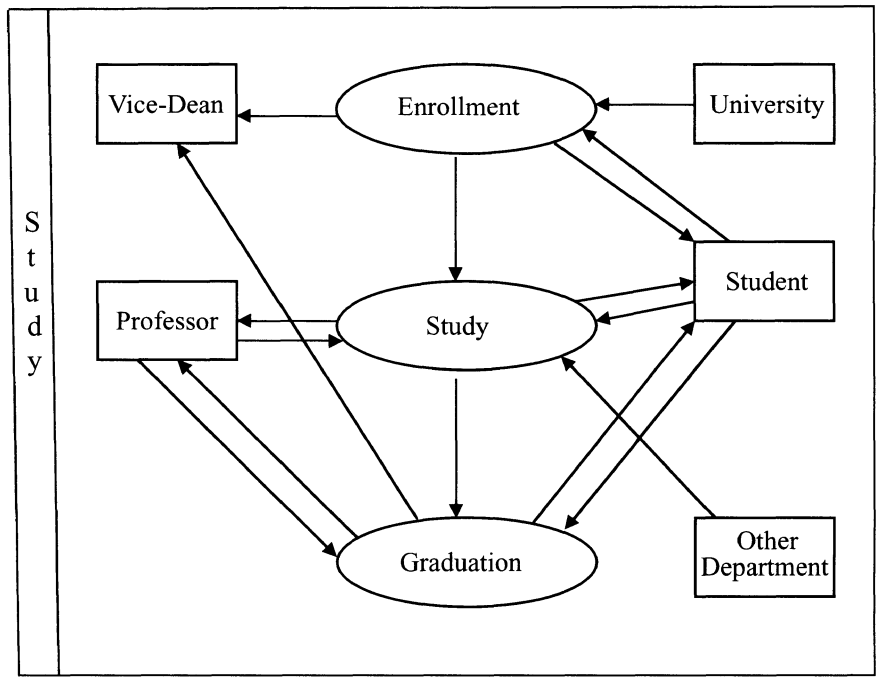


FIGURE 13.3. Work-process–level DFD

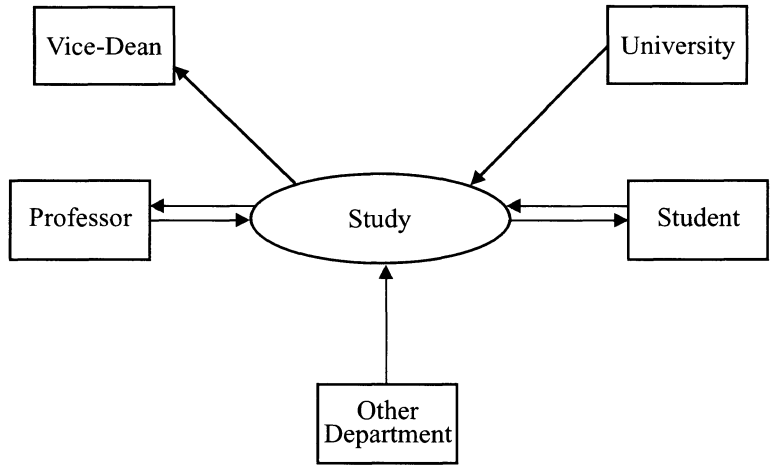


FIGURE 13.4. Business-process–level DFD (context diagram)

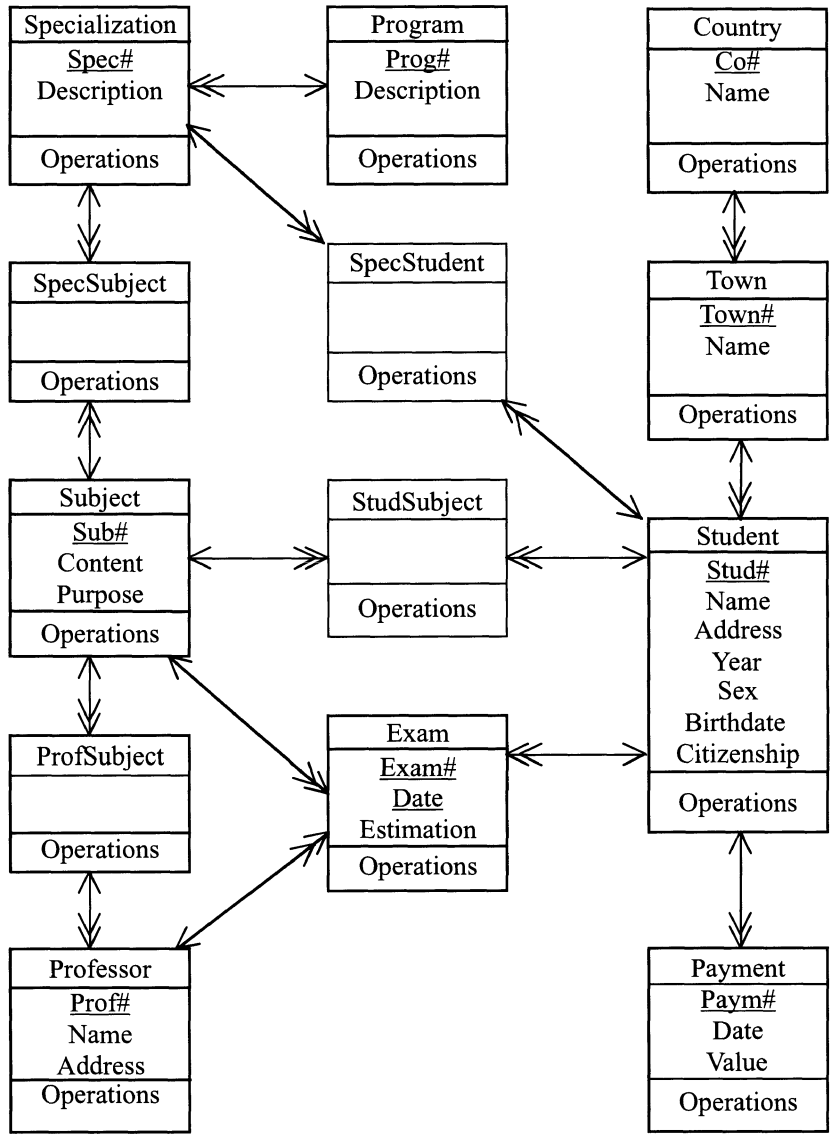


FIGURE 13.5. Initial object model of student office

Corresponding to the last step of the fourth phase, we analyzed the initial object model created, trying to implement the inheritance between classes. As a result of this analysis we found that class Student contains two types of students; these are full-time and part-time students. For this reason, we created two subclasses related to the class Student: PartTime and FullTime.

Figure 13.6 shows the completed object model of the student office.

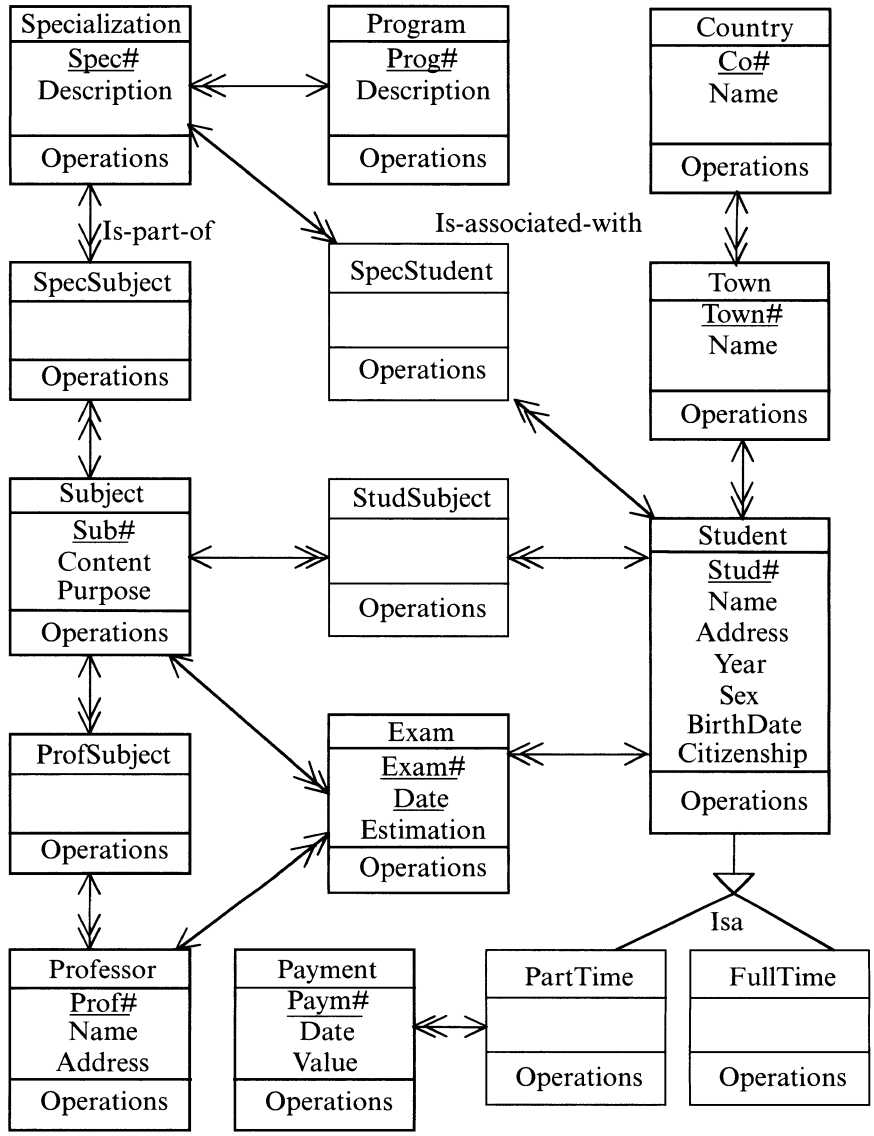


FIGURE 13.6. Object model of student office

13.5 Design

13.5.1. OPERATIONS

Corresponding to the procedure described in Section 8.1, we try to create a linkage between the functioning of the student office represented by

activity and task tables (Tables 13.3 and 13.4) and the static structure introduced by the object model of the student office (Figure 13.6).

For this purpose, we developed the operation table to identify the operations of the student office which define the mentioned linkage between the functioning of the system and the object model. Table 13.5 shows the operation table of the student office. The first column of the table lists the last four tasks as they are defined in the task table (Table 13.3).

The second column lists all operations extracted from the descriptions of the tasks in the task table: these are Enroll, RegisterPay, Confirm, EmpReport and ManagReport. The third column defines the classes of the object model with which the identified operations are related.

The reader may see that there are classes in the object model that do not include operations, because we include only operations related to the tasks described in the task table.

Figure 13.7 shows the object model extended with the operations listed in the second column of the operation table (Table 13.5).

The second task accomplished in the framework of the first step is writing algorithms that define in detail the operations identified. This task is usually done using an object-oriented programming language.

13.5.2. APPLICATION MODEL

As a result of using the second step of the fifth phase in the case of the student office, we created the application model shown in Figure 13.8. This application model has three parts. The first two were obtained from the entity table. The third part was developed from the information collected in the activity table.

The first part represents six reports, which analyze the enrollment of students, the students' success in the first year, the success of students related to professors and subjects, graduation themes and defenses, student, generations and the schedule of professors. The second part shows a decision support problem. This part deals with preparing a study schedule at the beginning of the study year.

The last part of the application model was derived from the activity

Table 13.5. Operation table of student office

Task Code	Operation	Class
$K_{4,3}$	Enroll	Student
	Enroll	SpecStudent
	RegisterPay	Payment
$K_{5,3}$	Confirm	Student
$K_{6,3}$	EmpReport	Student
$K_{6,2}$	ManagReport	Student

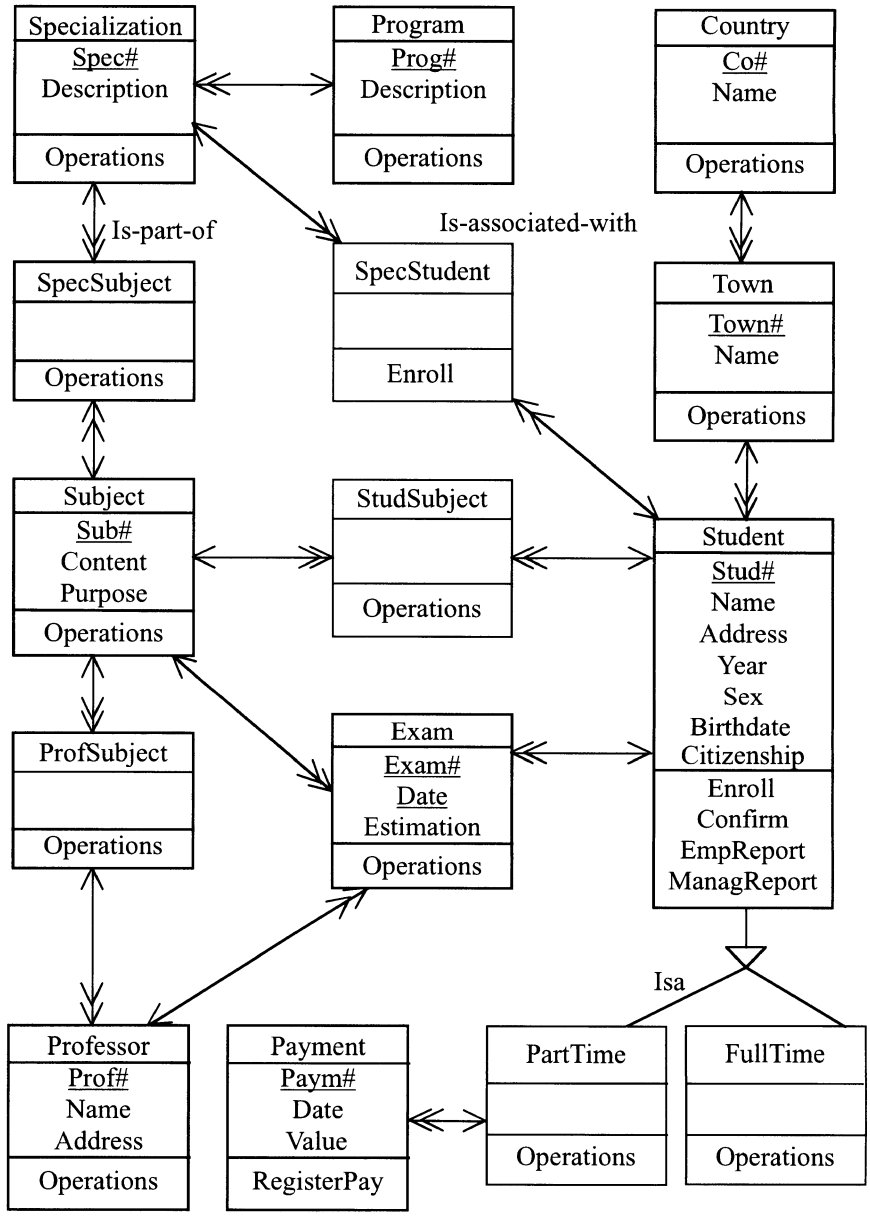


FIGURE 13.7. Object model of student office

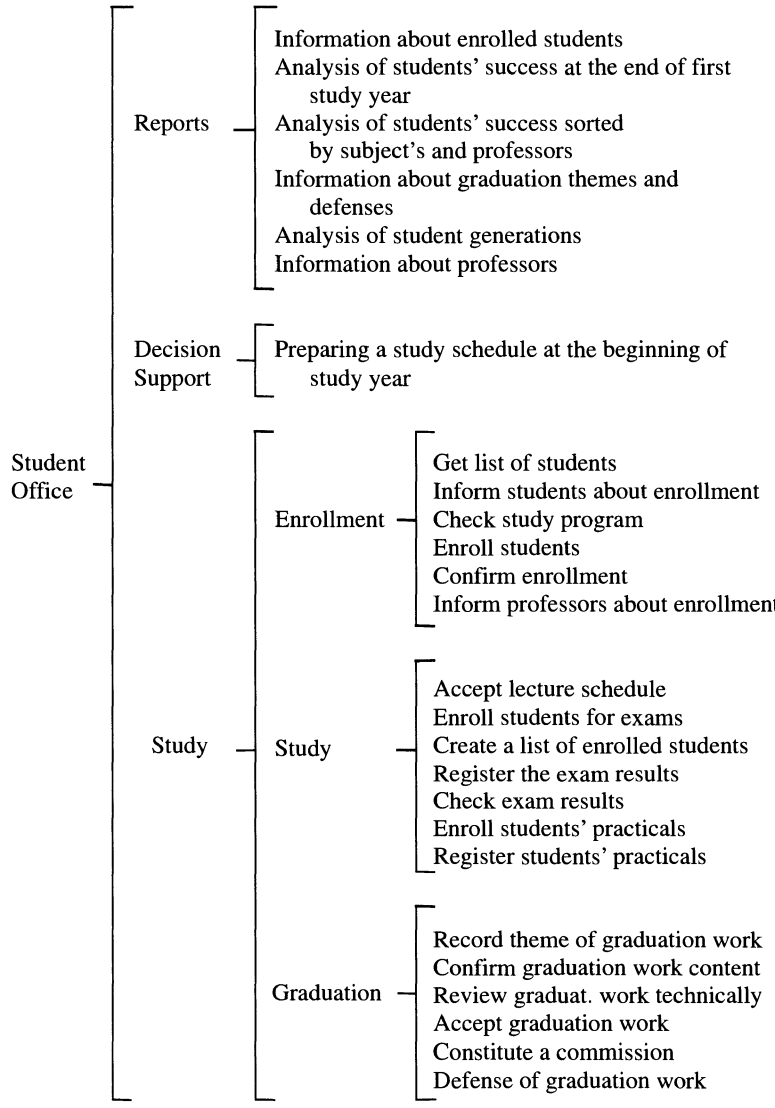


FIGURE 13.8. The application model of the student office

table. This part represents the business process "Study," which has three work processes; these are "Enrollment," "Study" and "Graduation."

The first work process deals with implementing the procedure for enrollment of students in the first and higher study years. The second work process controls the whole study process by registering information about the students' obligations, exams and practicals. The third activity has six tasks and deals with defining the procedure of student graduation.

CHAPTER 14

TAD Case-Tool

This final chapter introduces the TAD case-tool. This software implements most of the algorithms of TAD methodology described in the previous chapters. The purpose of developing such a case-tool is to support the use of the methodology and to enable the reader to test the ideas and concepts of the methodology explained in different phases and steps.

The reader should understand that the software introduced is the first version of the TAD case-tool. Therefore, it may still contain some bugs. For this reason, the author would be very grateful for any information about any bug(s) the reader discovers.

In addition, it has to be emphasized that this version of the TAD case-tool covers the first and second phases completely. Furthermore, it covers part of the third phase and the fifth phase. Meanwhile, the second part of the third phase, which is the process model, and the fourth phase, which is the object model, are not covered. To develop the mentioned two models, the reader may use any other software specialized for drawing such models.

14.1 Installation and Start

To begin the installation, insert the CD-ROM into your CD-ROM drive, use Windows Explorer to locate your CD-ROM drive, run the setup.exe file, and follow the installation instructions.

To start the TAD case-tool, click Start, click Programs, and run TAD application to see Figure 14.1 on screen.

T. Damij, *Tabular Application Development for Information Systems*

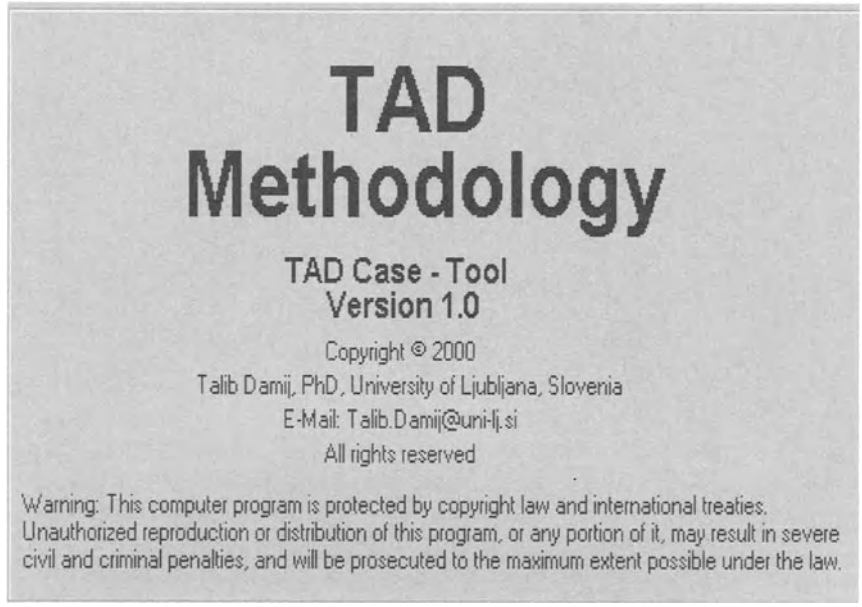


FIGURE 14.1. TAD case-tool

14.2 Application

To use the TAD case-tool we first have to define an application. Each application is defined by a file, which is a TAD type; for example, "TouristOrganization.tad."

To do so, click Figure 14.1 or press Enter to get TAD screen Figure 14.2. Click **File—New, File—Open,** or click one of these icons to create a new or open an existing application.

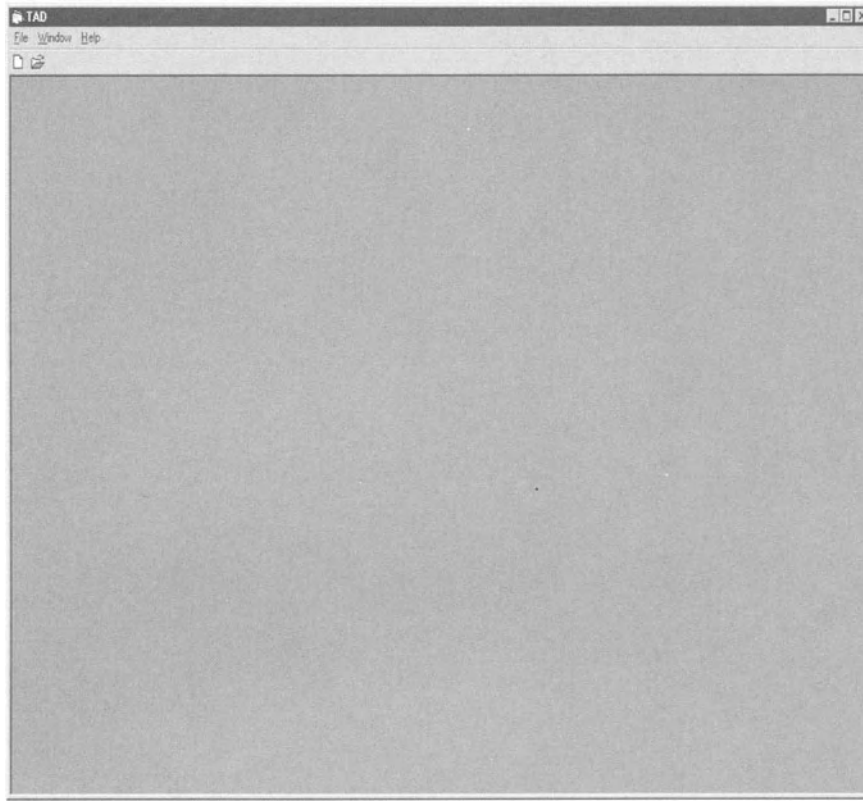


FIGURE 14.2. TAD screen

14.2.1. CREATING AN APPLICATION

To create a new application, click **File** from Figure 14.2 and then **New,** or click **New** icon and you will see Figure 14.3, "Create New Application."

In the Save in field (Figure 14.3) choose a folder for the new application. In the File name field enter the application name and click the **Save** icon to save the file and to get Figure 14.4, "Application Properties."

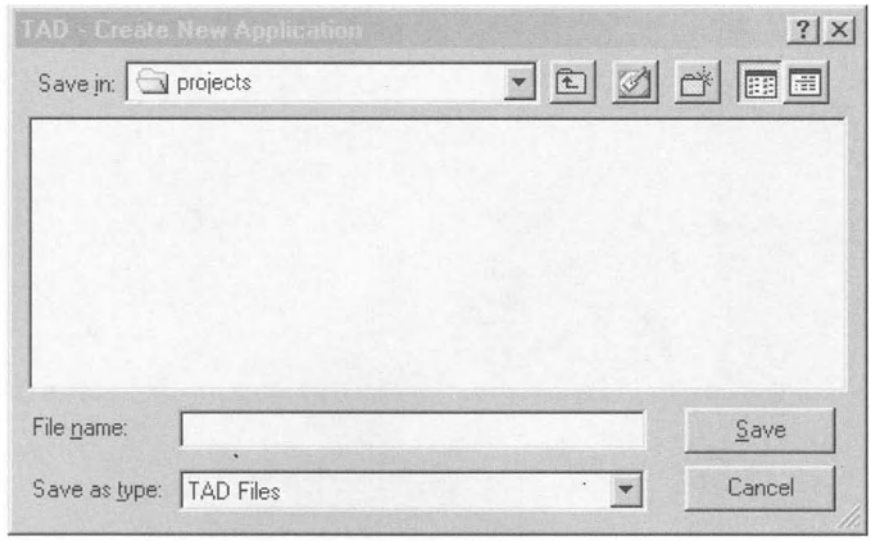


FIGURE 14.3. Creating an application

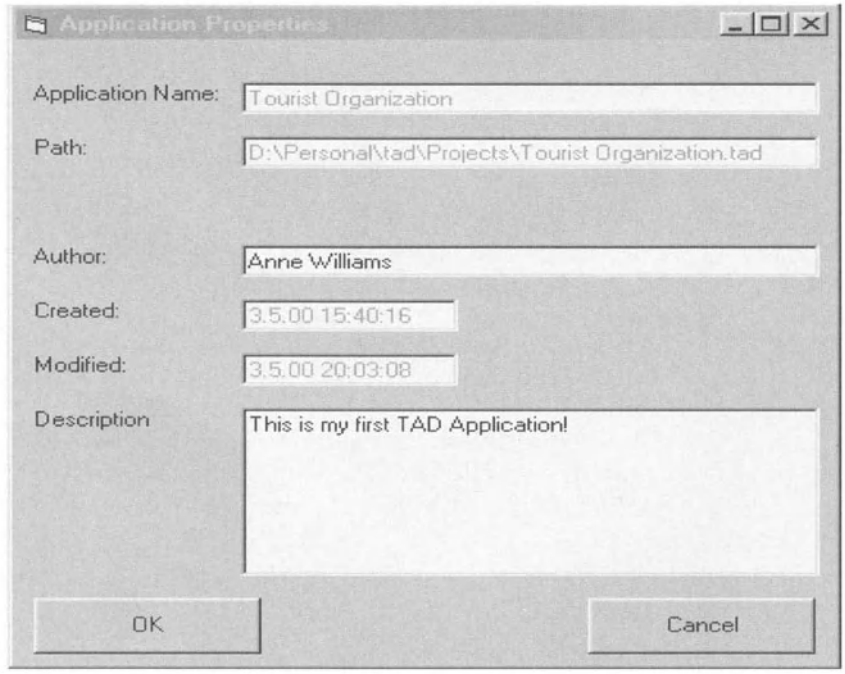


FIGURE 14.4. Application properties

FIGURE 14.4. Application properties

Figure 14.4 shows information about the application such as application name, path, created and modified dates.

In the Author field of Figure 14.4 enter the author name and in the Description field enter a short description of the application. Click the **OK** icon to save the entered data and to go to Figure 14.6 to start creating the entity table.

14.2.2. OPENING AN APPLICATION

To open an existing application, click **File** from Figure 14.2 and then click **Open,** or click the **Open** icon and you will get Figure 14.5, "Open Application."

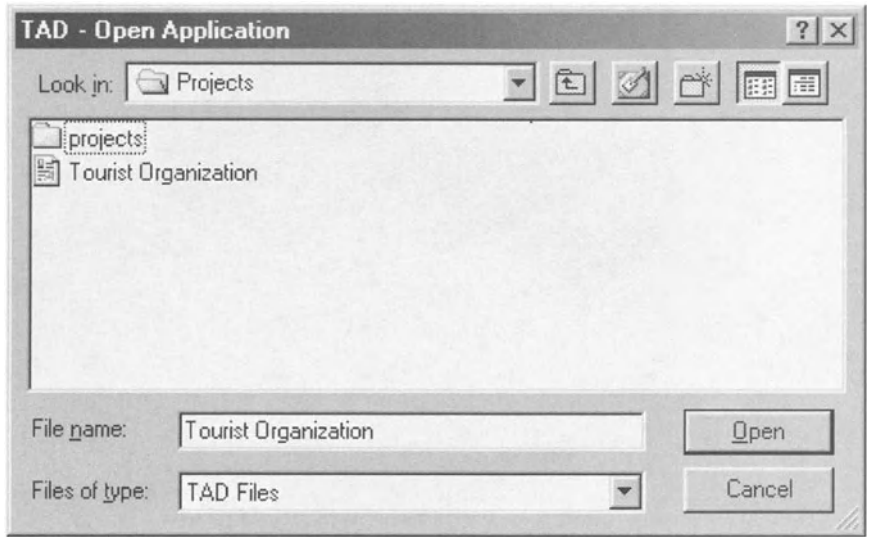


FIGURE 14.5. Opening an application

To open a particular application, choose a folder, select the desired application, click the **Open** icon to see Figure 14.4 and then click the **OK** icon to get Figure 14.6, which enables us to develop the entity, activity and task tables.

Let us emphasise that only one application can be opened at the time. So, before opening any other application, we have to close the opened one if there is an open application.

14.3 TAD Main Menu

Figure 14.6 shows TAD main menu, which is used to perform the described phases of TAD methodology.

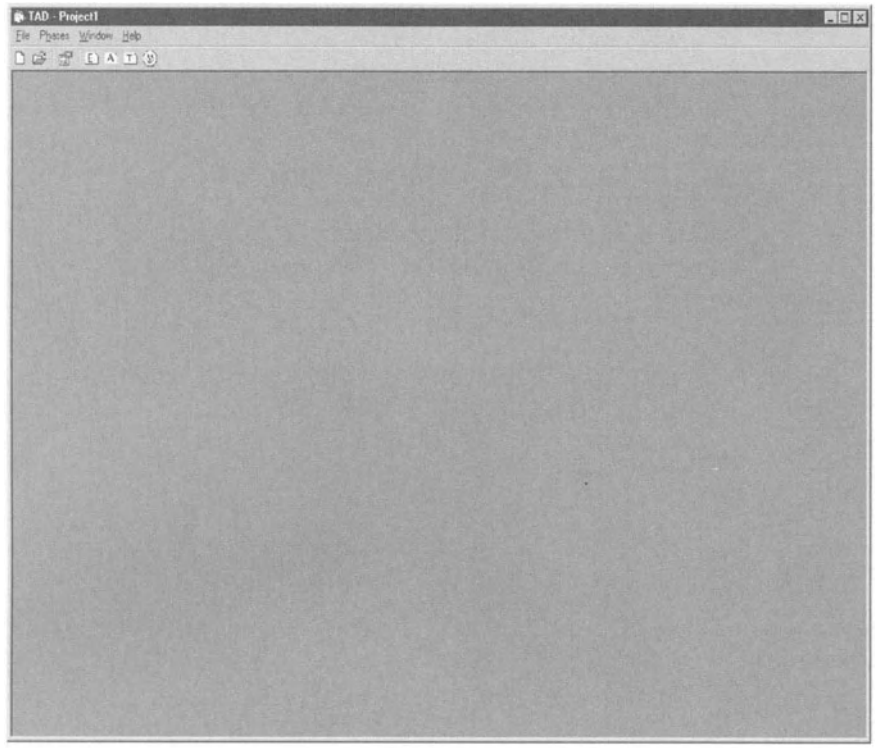
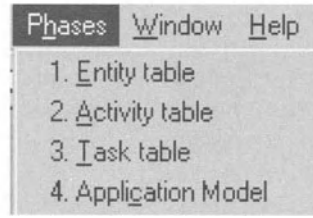


FIGURE 14.6. Main menu

Click **Phases** and you will get the following menu:



Select **Entity table, Activity table** or **Task table** to start or continue developing the entity, activity or task table. When all these tables are completed, select **Application Model** to create the application model.

Another possibility is to click one of the following: L A T icons to open the entity table, the activity table or the task table. Click the icon to create the application model.

To close the active table, click the opened icon again. It has to be emphasized that only one icon can be opened at a time. It is possible to

work with more than one table at the same time, which we will stress later.

We may also move the tables or change their size on the screen as we do with windows in the Windows environment.

14.4 Entity Table

To create a new or to open an existing entity table from the main menu, click **Phases** and then **Entity table** and you will see Figure 14.7. Another possibility is to click the icon to bring the entity table.

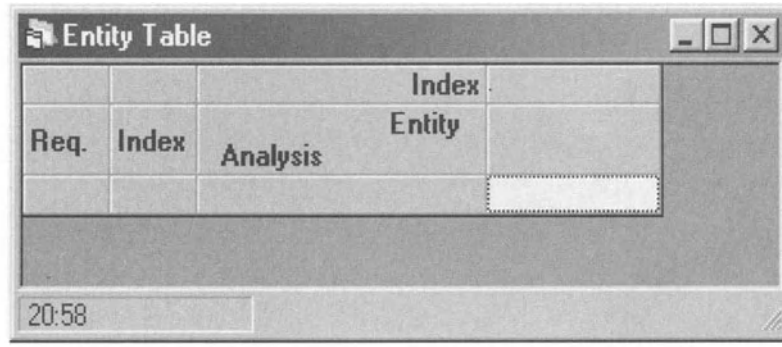


FIGURE 14.7. Entity table

14.4.1. NEW ENTITY

To create a new entity, click **Edit** from the main menu to get the Edit menu and then click **Entity.**

to see the following menu:

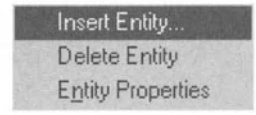


FIGURE 14.8. Entity menu

Another way to bring Figure 14.8 on-screen is to click the right mouse button. Now click **Insert Entity** and you will get Figure 14.9.

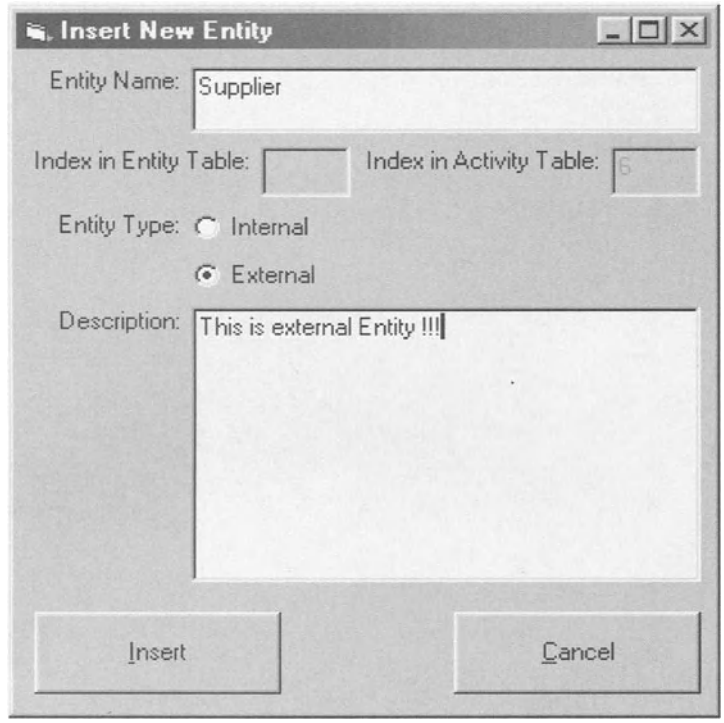


FIGURE 14.9. Inserting an entity

In the Entity Name field enter the entity name. The entity type is Internal by default, because the entity table contains only internal entities. In the Description field we may enter a description of the created entity. Click the **Insert** icon to create a new entity or **Cancel** to cancel this operation.

14.4.2. ENTITY PROPERTIES

To change the properties of an existing entity you have to locate the cursor in the column of the desired entity in the entity table. Click **Edit, Entity** and **Entity Properties** and you will get Figure 14.10, where you can change entity properties.

Another possibility is as follows: After positioning the cursor in the column of the desired entity in the entity table, click the right mouse button to bring Figure 14.8 and then click **Entity Properties** to get Figure 14.10.

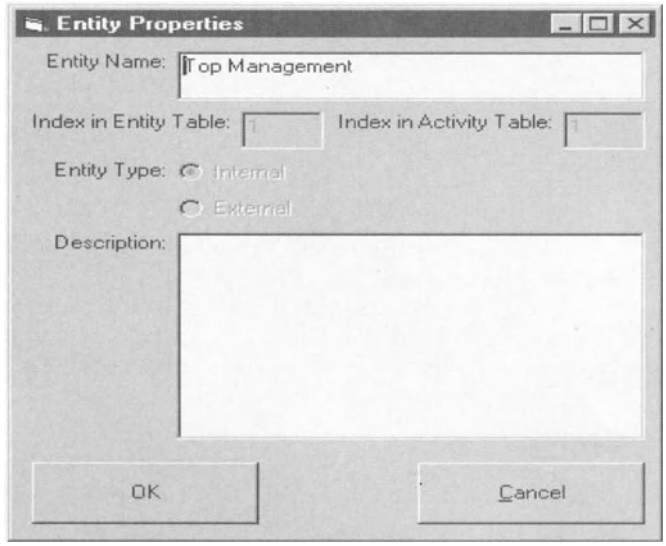
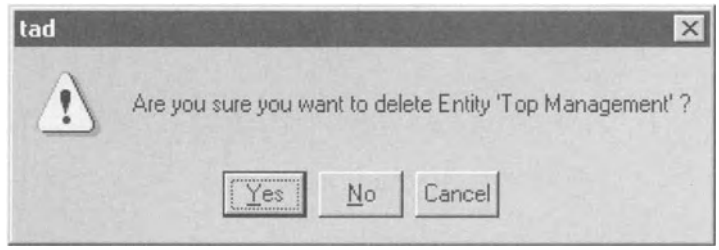


FIGURE 14.10. Entity properties

14.4.3. DELETE ENTITY

To delete an entity from the entity table, position the cursor in the column of the entity in the entity table and click **Edit, Entity** and **Delete Entity.** To complete the deletion of the entity, you have to confirm this operation.

We can click the right mouse button to get Figure 14.8 and now click **Delete Entity** and confirm the deletion operation.



14.4.4. NEW ANALYSIS

To create a new analysis, click **Edit** from the main menu to see the Edit menu.

Click **Analysis** to bring the menu of Figure 14.11, where we click **Insert Analysis** to get Figure 14.12.

FIGURE 14.11. Analysis menu

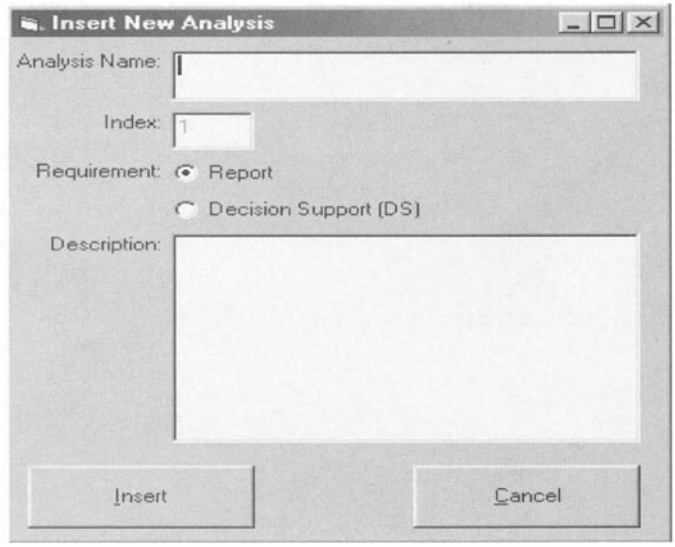


FIGURE 14.12. Inserting an analysis

In the Analysis Name field enter the name of the analysis. The Index field shows the index of the analysis in the entity table. In the Requirement check box choose the type of analysis; this may be a report or a decision support. Finally, in the Description field enter a short description of the analysis entered; this is an optional choice. At the end click the **Insert** button to save the entered data.

To create a new analysis, we may use another method. Click the right

mouse button in the Analysis column of the entity table to bring Figure 14.11 on-screen and choose **Insert Analysis** to see Figure 14.12.

14.4.5. ANALYSIS PROPERTIES

To change Analysis properties, click **Edit, Analysis** and **Analysis Properties** and you will see Figure 14.13, where, after making the needed changes, you click **OK** to save the changes.

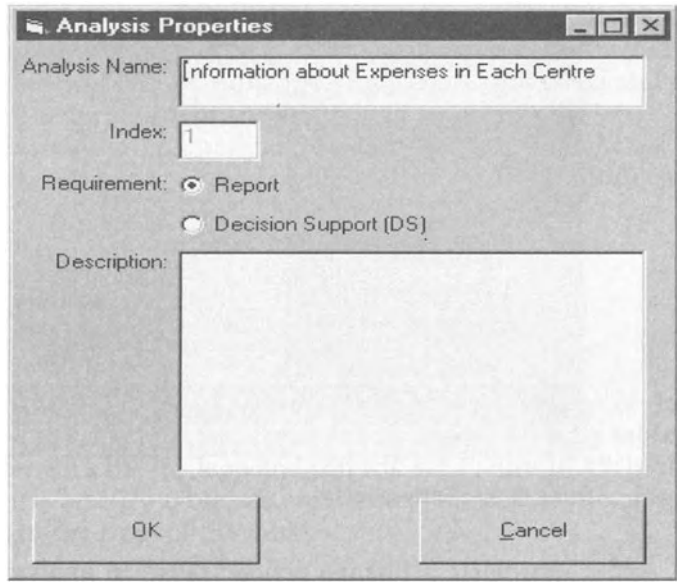
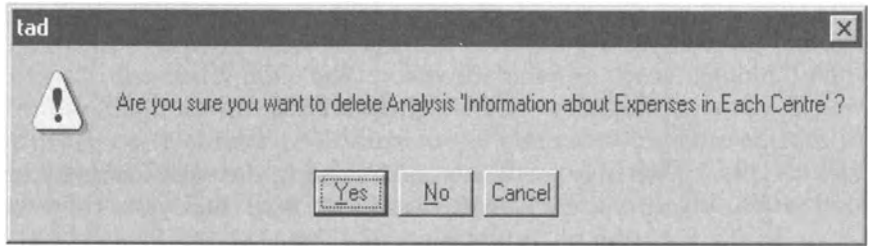


FIGURE 14.13. Analysis properties

14.4.6. DELETE ANALYSIS

To delete an analysis from the entity table, first locate the cursor in the row of the analysis you wish to delete, click **Edit** from the main menu to get the Edit menu, choose **Analysis** to get Figure 14.11, click **Delete Analysis** and confirm the deletion operation.

Or to bring Figure 14.11, click the right mouse button; after positioning the cursor in the row of the desired analysis, click **Delete Analysis** and confirm the deletion.



14.4.7. LINKAGE DEFINITION

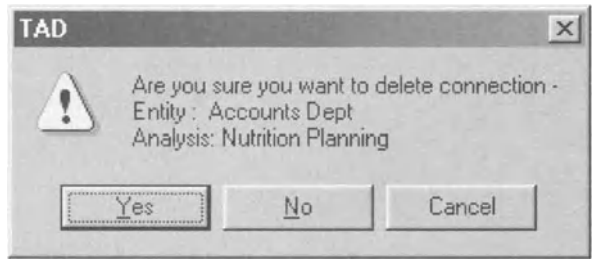
To link between analysis in row i and entity in column j, simply position the cursor in cell(i,j) and click the left mouse button twice and you will see an asterisk written in the cell.

Entity Table

Req.	Index	Analysis (Index / Entity)	1 Top Management	2 Purchasing Dept	3 Marketing Dept	4 Accounts Dept	5 Restaurant
Report	1	Information about Contracts	*		*	*	
	2	Information about Expenses in Each Centre	*	*	*	*	
	3	Information about Guests	*		*	*	*
	4	Information about Guests Complaints	*		*		*
	5	Information about Hotel Occupation	*	*	*	*	*
	6	Information about Suppliers	*	*		*	*
	7	Information about Turn-Over of Each Centre	*		*	*	
DS	8	Nutrition Planning	*	*			*

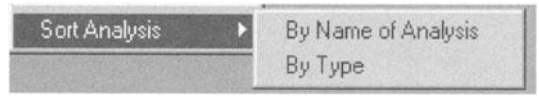
FIGURE 14.14. Entity table

To delete a linkage defined between analysis in row i and entity in column j, position the cursor in cell(i,j), click the left mouse button twice and confirm the deletion.



14.4.8. SORTING ANALYSES

We may sort analyses defined in the entity table by name or by type. To do so, click **Edit, Analysis, Sort Analysis,** then choose between **By Name of Analysis** and **By Type.**



14.4.9. CHANGING POSITION AND REFRESHING

Sometimes we need to change the position of a certain entity or a determined analysis in the entity table. To do so, click the fixed part of the table; this is the fixed part of a column of the entity or the fixed part of a row of the analysis; hold the left mouse button pressed and move the entity to a new column or the analysis to a new row.

It is important to emphasize that after making any change, particularly changes related to different tables, click **Edit** and **Refresh** to save the changes.

14.5 Activity Table

To start developing or to continue with the development of the activity table from the main menu, click **Phases,** choose **Activity table,** or from the toolbar click the icon and you will see Figure 14.15. In the framework of the activity table we create new entities in the columns and activities in the rows of the table.

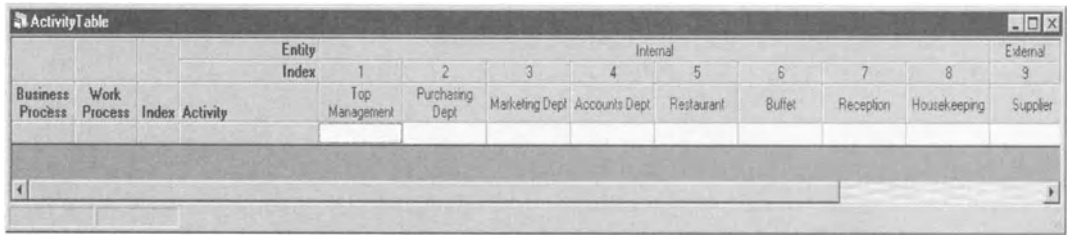


FIGURE 14.15. Activity table

14.5.1. ENTITY

When a new activity table is opened you will notice that the entities defined in the entity table are already included in the new table. In addition to these entities, we may define new entities. To create a new entity, we use a similar procedure to the one used to create entities in the entity table. To do so, click **Edit** to see Figure 14.16. Now click **Entity** and then **Insert Entity** to get Figure 14.9. Or click the right mouse button and **Insert Entity** to see Figure 14.9.

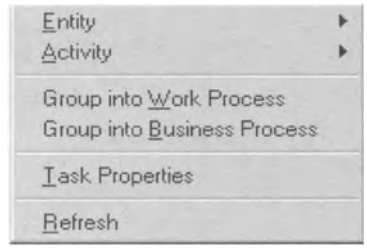


FIGURE 14.16. Edit menu

In the Entity Name field enter the entity name. In the Entity Type check box choose the entity type; this is either Internal or External. In the Description field enter a description of the entered entity if such a description is needed. Then click the **Insert** icon to create a new entity or **Cancel** to cancel the entity creation.

To change the properties of an entity, first position the cursor in the column of the entity in the activity table. Then click **Edit, Entity** and **Entity Properties** and you will see Figure 14.10, which enables us to change entity properties. We also can change entity properties using the following procedure: Position the cursor in the fixed part of the column of the entity; click the right mouse button and then click **Entity Properties** to get Figure 14.10.

To delete an entity from the activity table, position the cursor in the column of the entity, click **Edit, Entity** and **Delete Entity.** At the end confirm the deletion operation. Or position the cursor in the fixed part of the column of the entity, click the right mouse button, click **Delete Entity** and confirm this operation.

It has to be emphasized that the deletion of an entity from the activity table also deletes the entity from the entity table. After such an operation, we have to click **Edit** and **Refresh.**

To change the position of a certain entity in the table, click the fixed part of the column of the entity, hold the left mouse button pressed and move the entity to its new column. To change the position of an activity, click the fixed part of the row of the activity, hold the left mouse button pressed and move the activity to its new column.

14.5.2. ACTIVITY

To insert a new activity in the activity table, click **Edit** to get Figure 14.16. Now click **Activity,** and then **Insert Activity** to see Figure 14.17. Or click the right mouse button in the Activity column of the activity table and then click **Insert Activity.**

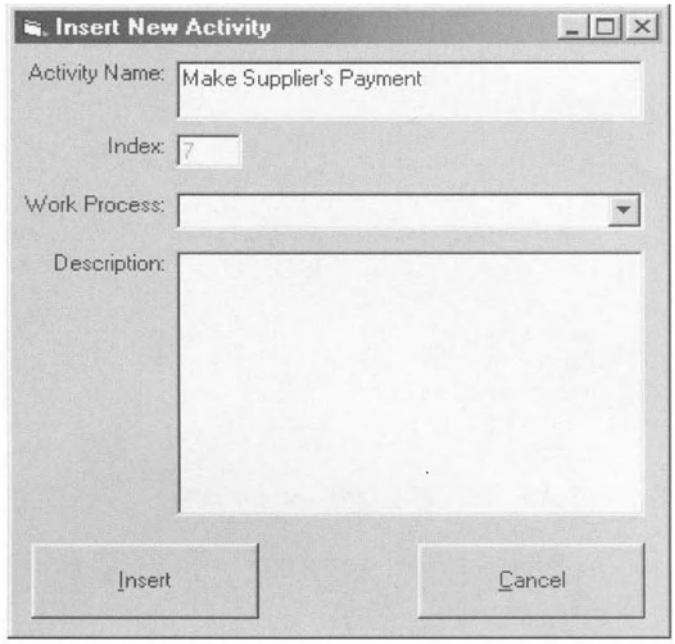


FIGURE 14.17. Inserting an activity

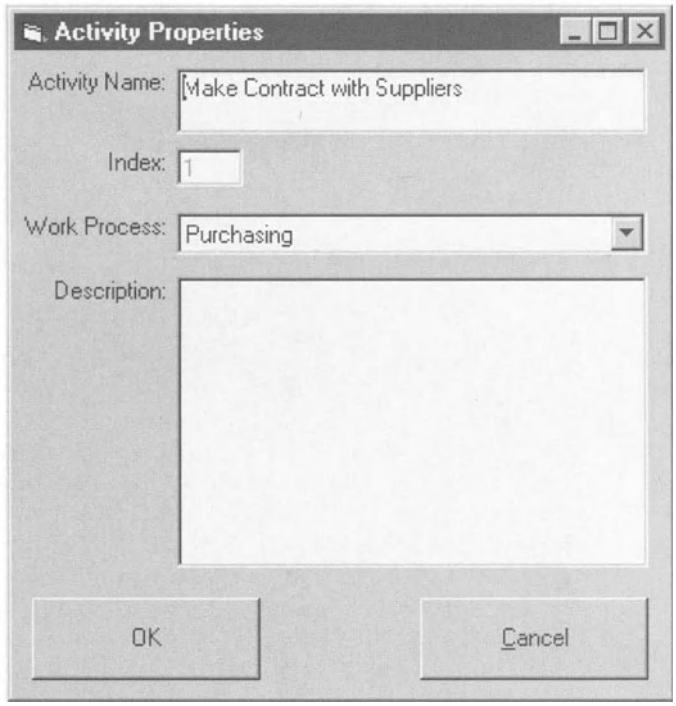


FIGURE 14.18. Activity properties

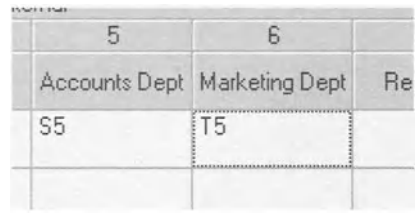
In the Activity Name field enter the activity name. The Index field is read-only and shows the index of the current activity in the activity table. The Work Process field will be stressed later. The Description field is optional; here we could enter a description of the activity. To complete the insertion of the activity in the activity table, click **Insert** button. To cancel this operation, click the **Cancel** button.

To change the properties of an activity, click **Edit, Activity** and **Activity Properties** or click the right mouse button in the Activity column and then **Activity Properties** to see Figure 14.18.

To delete an activity from the activity table, click **Edit, Activity** and **Delete Activity** or click the right mouse button in the Activity column and then choose **Delete Activity.** To complete this operation, we have to confirm the deletion of the activity.

14.5.3. HORIZONTAL LINKAGE

The horizontal linkage of an activity is defined by determining the source and target entities of the treated activity. To do so, position the cursor in the cell stated in the row of the activity and in the column of the source entity, click the Ctrl button and the left mouse button and you will see the arrow ⇔. Keep the left mouse button pressed and move to the cell of the target entity; release the left mouse button to see that a horizontal linkage is created by indicating the source and target entities with letters S and T, such as shown in the following figure.



To delete a defined horizontal linkage, position the cursor in the activity row inside the Activity column of the table, click **Edit, Activity** and **Horizontal Linkage List** and you will see Figure 14.19. Or position the cursor in the activity row in the framework of the Activity column of the activity table. Click the right mouse button, **Activity** and **Horizontal Linkage List** to get Figure 14.19. Now select the linkage which you wish to delete, click the **Delete** button, and confirm the deletion operation by clicking the **Yes** button in the following figure.

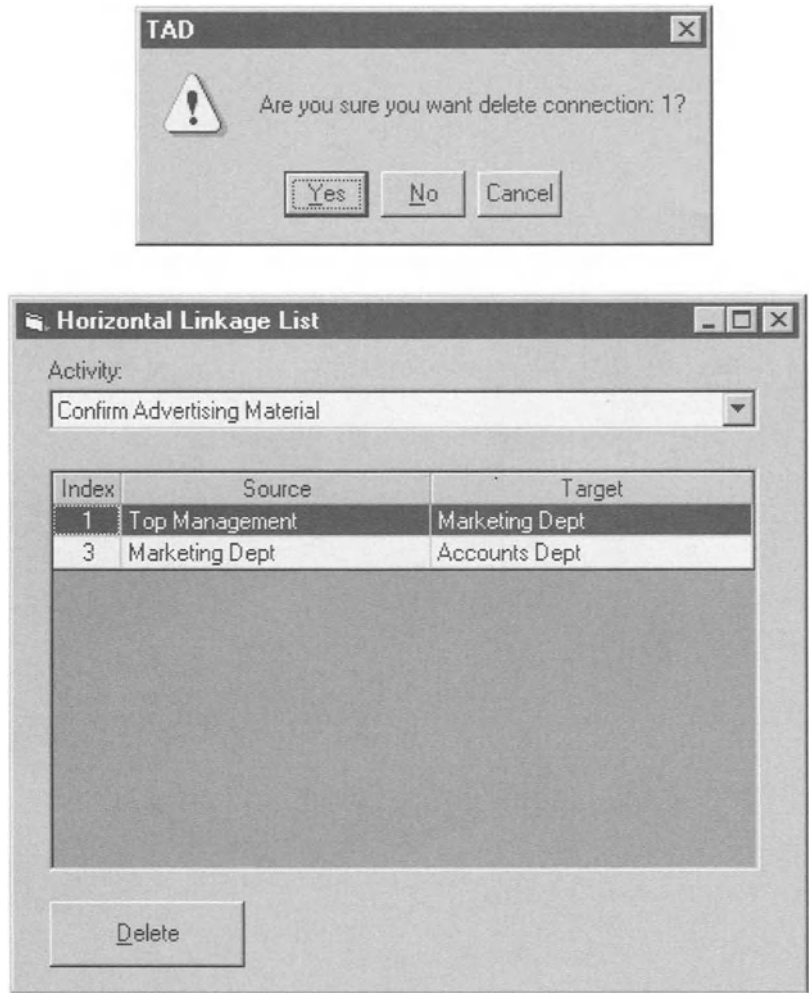


FIGURE 14.19. Horizontal linkage deletion

14.5.4. VERTICAL LINKAGE

A vertical linkage can be defined only in the column of an internal entity to define the sequence of tasks performed by the entity. To define such a linkage, position the cursor in the cell stated in the column of the entity and in the row of the predecessor activity. Click the Alt button and the left mouse button and you will see the arrow ⇳. Keep the left mouse button pressed and move to the cell of the successor; release the left mouse button to see that a vertical linkage is created by indicating the predecessor and successor activities with letters P and U; see the following figure.

	Index	1
	Activity	Top Management
	Make Contracts with Suppliers	T2 P1
2	Confirm Suppliers Contract	S1 U1

To delete a vertical linkage, position the cursor in the fixed part of the entity column; click **Edit, Entity** and **Vertical Linkage List** to get Figure 14.20. Or position the cursor in the fixed part of the entity column, click the right mouse button, **Entity** and **Vertical Linkage List** to see Figure 14.20. Now select the linkage you wish to delete, click **Delete** and confirm the deletion operation.

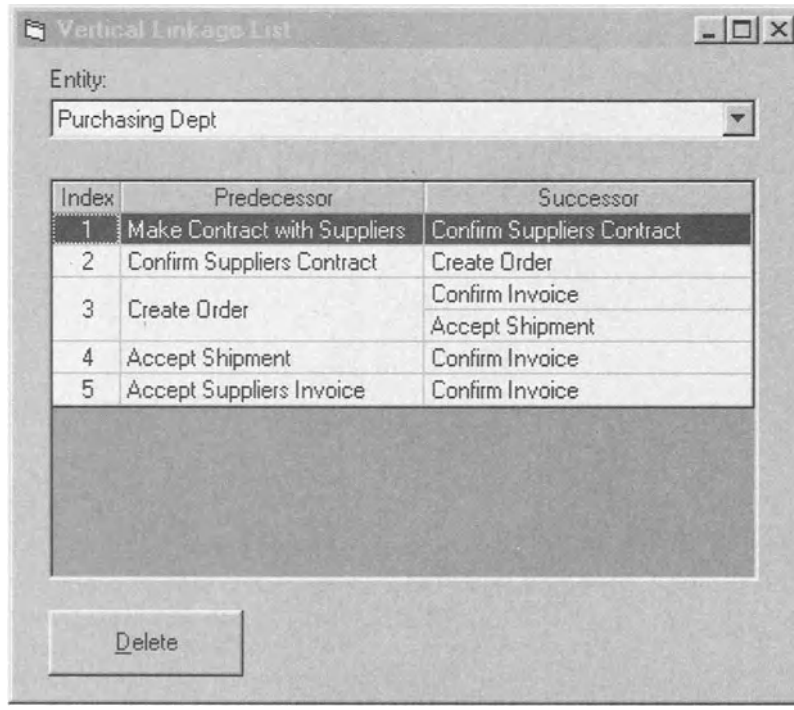


FIGURE 14.20. Vertical linkage deletion

14.6 Task Table

To create a new or open an existing task table, click **Phases** and **Task table,** or click the icon from the toolbar and you will see the task table (Figure 14.21), which already lists all tasks defined in the activity table.

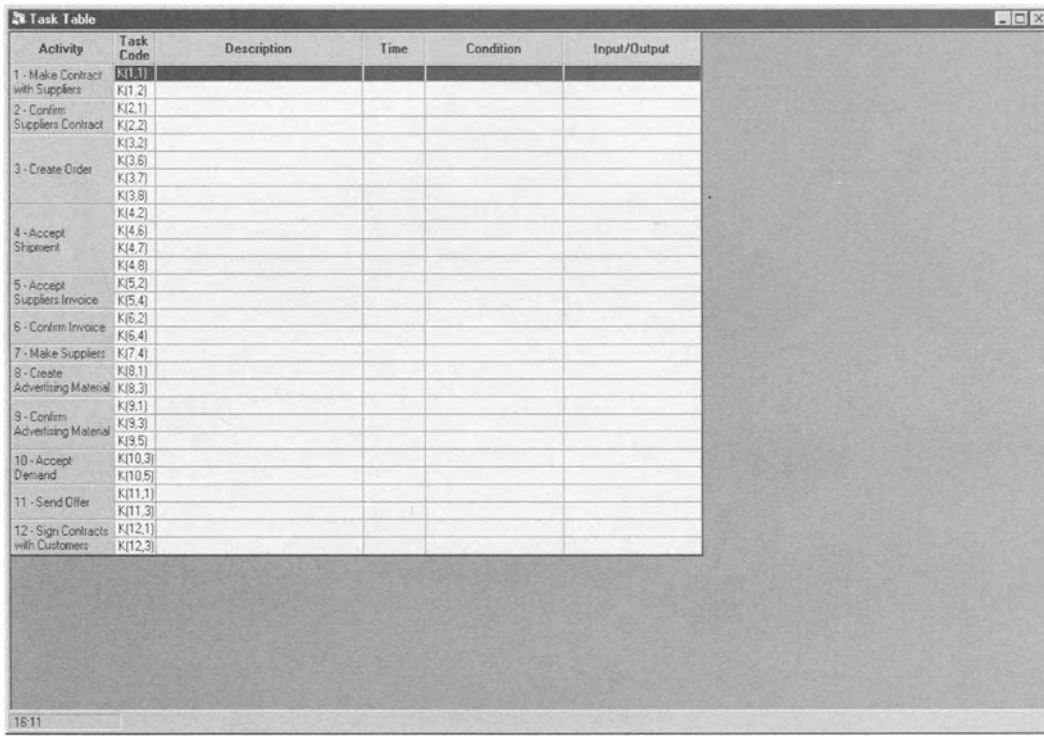


FIGURE 14.21. Task table

To describe the characteristics of a certain task defined in the task table position in its row, click **Edit** and **Edit Task Properties** to get Figure 14.22, which enables us to enter the task's characteristics. Or position the cursor in the task's row, click the right mouse button and choose **Edit Task Properties.**

In the Description field enter a precise task description. Field Time is optional. In the Condition field enter one or more conditions related to the task. In the Input/Output field enter inputs and outputs connected with the task. Click the **OK** button to save the entered data or the **Cancel** button to cancel the operation.

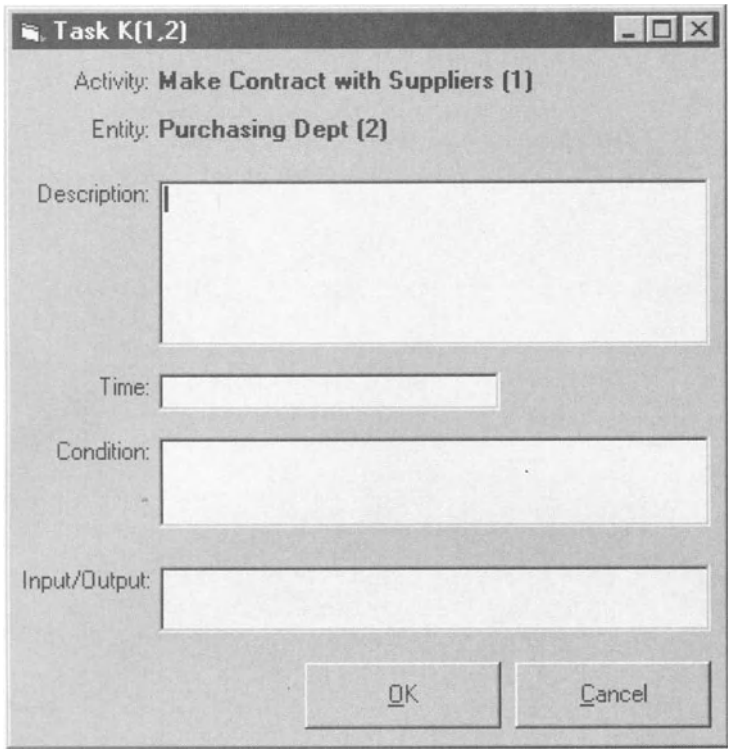


FIGURE 14.22. Task properties

Task characteristics can be entered directly from the activity table. To do so, position the cursor in any non-empty cell in the activity table, click the **Edit** button to see Figure 14.16, where we choose **Task Properties** to get Figure 14.22. Or after positioning the cursor in a certain cell, click the right mouse button and choose **Task Properties.**

14.7 Work Process

In accordance with TAD methodology, after defining all activities of the activity table, we continue with grouping them into work processes. To

do so, open the activity table, click **Edit** to get Figure 14.16 and choose **Group into Work Process** to see Figure 14.23.

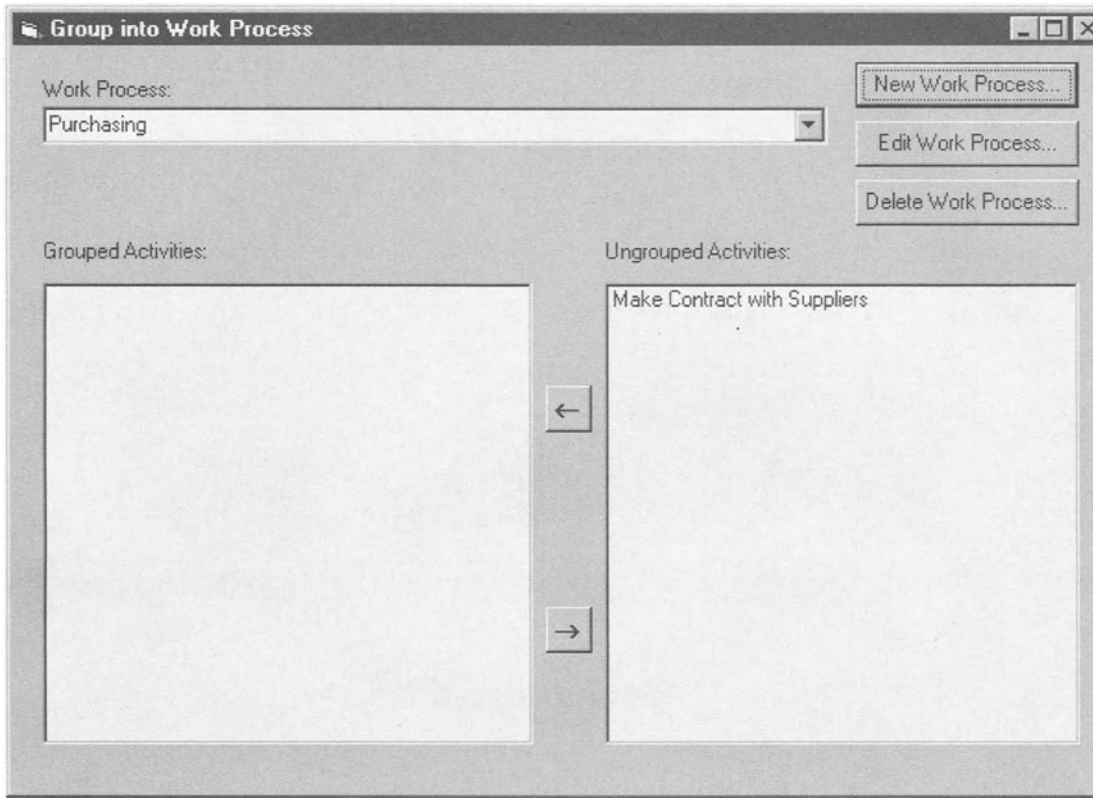


FIGURE 14.23. Grouping into work processes

The reader may notice that Figure 14.23 lists all ungrouped activities of the activity table in the window "Ungrouped Activities." To group one or more of these activities into a work process, we have to define a new work process by clicking **New Work Process** to see Figure 14.24.

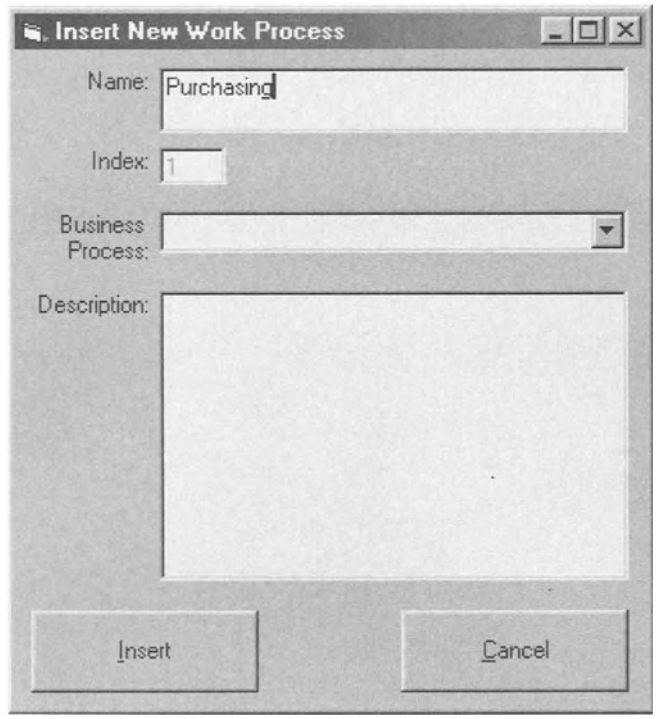


FIGURE 14.24. Inserting work process

In the Name field enter the work process name. Field Index is read-only and shows the index of the work process in the activity table. The Business Process field shows the current work process is defined inside this business process. Because business processes are not yet defined, this field stays empty. In the Description field, which is optional, enter a description of the work process. Click the **Insert** button to insert the work process or **Cancel** to cancel that.

After defining work processes we may group the ungrouped activities into different work processes. To do this, select the work process from the combo box in Figure 14.23, select one or more ungrouped activities and then click the left arrow to bring the selected activities to the "Grouped Activities" window. Grouping activities into work processes is also seen in the activity table immediately.

We use the right arrow to bring one or more selected activities from the "Grouped Activities" window back to the "Ungrouped Activities" window.

To change data of a certain work process, select the work process from the combo box in Figure 14.23, click **Edit Work Process** and you will see Figure 14.25, where you can change its data. To save changes, click the **OK** button or click the **Cancel** button to cancel this operation.

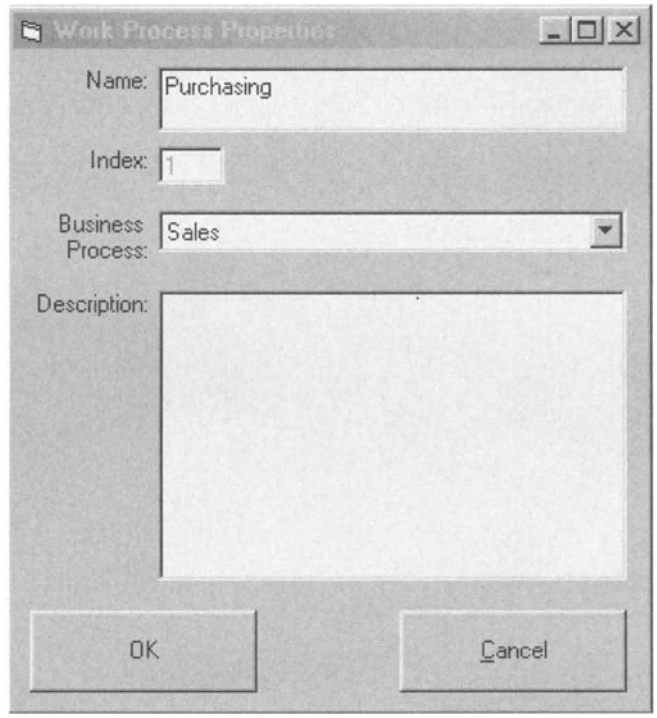


FIGURE 14.25. Work process properties

To delete a work process, select it from the combo box in Figure 14.23, click the **Delete Work Process** button and confirm the deletion operation.

14.8 Business Process

After grouping the activities into a set of work processes, we go on with grouping the defined work processes into business processes. To achieve this, open the activity table, click **Edit** to get Figure 14.16, and choose **Group into Business Process** and you will see Figure 14.26.

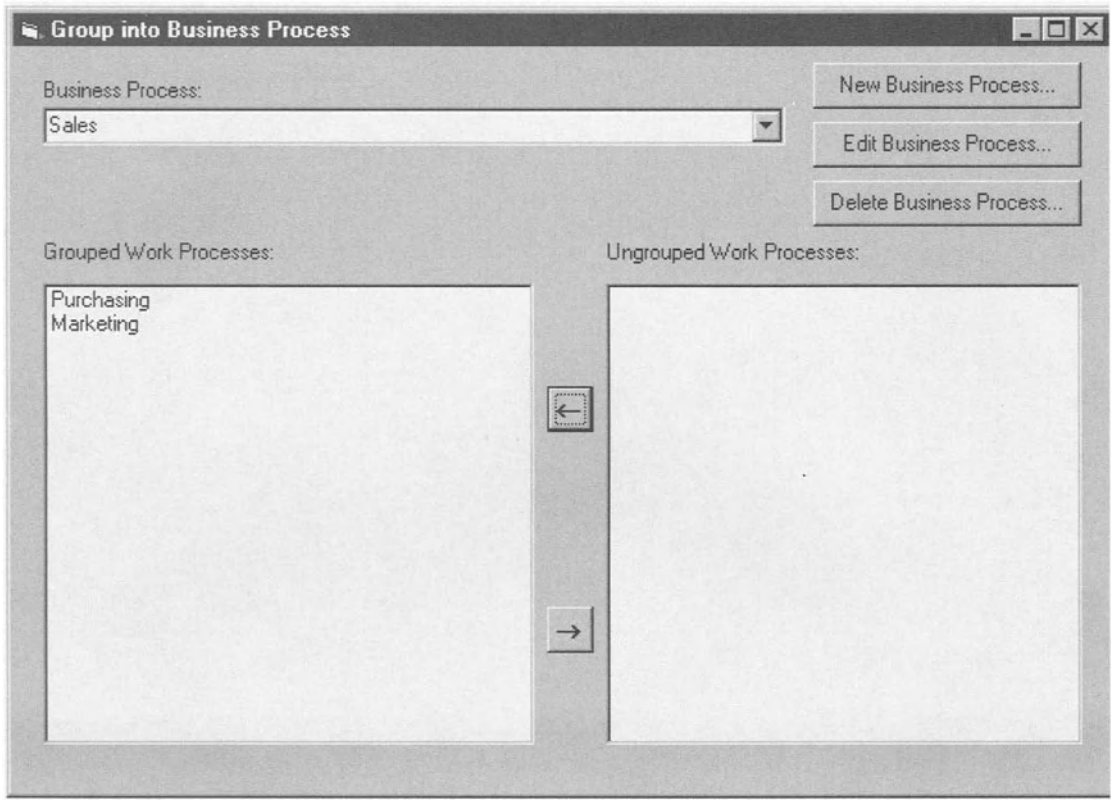


FIGURE 14.26. Grouping into business processes

To group one or more ungrouped work processes, we first define a new business process or select an existing one. To create a new business process, click **New Work Process** to bring on-screen Figure 14.27, which enables us to enter its data.

In the Name field enter the business process name. Field Index is read-only and shows the business process position in the activity table. In the Description field you may enter a description of the business process. Finally, click **Insert** to insert a business process or **Cancel** to cancel the insertion.

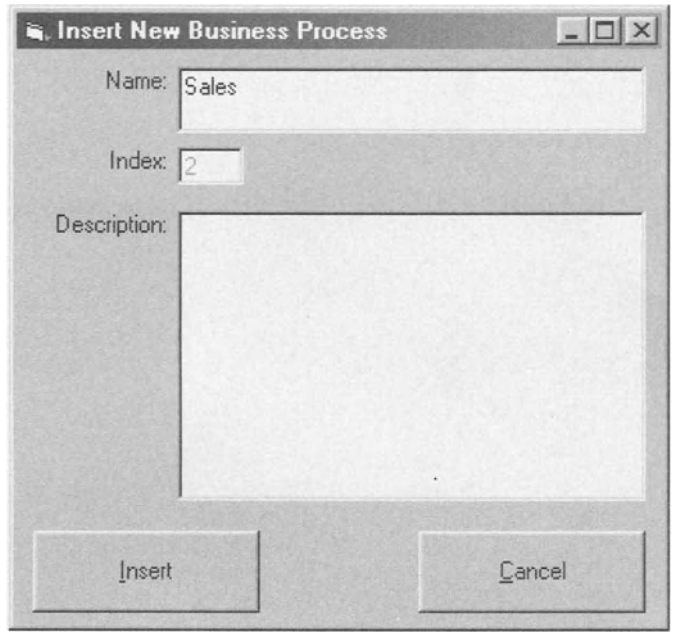


FIGURE 14.27. Inserting business process

To group a set of ungrouped work processes into a business process, select the business process from the combo box, select one or more ungrouped work processes and then click the left arrow to bring the selected work processes to the "Grouped Work Processes" window. Grouping work processes into business processes is also indicated in the activity table immediately.

We can use the right arrow to bring one or more work processes from the "Grouped Work Processes" window to the "Ungrouped Work Processes" window after selecting them.

To edit the data of a certain business process, select the work process from the combo box of Figure 14.26 and click **Edit Business Process** to get Figure 14.28, where you can change business process data. To save the changes, click the **OK** button or click the **Cancel** button to cancel this operation.

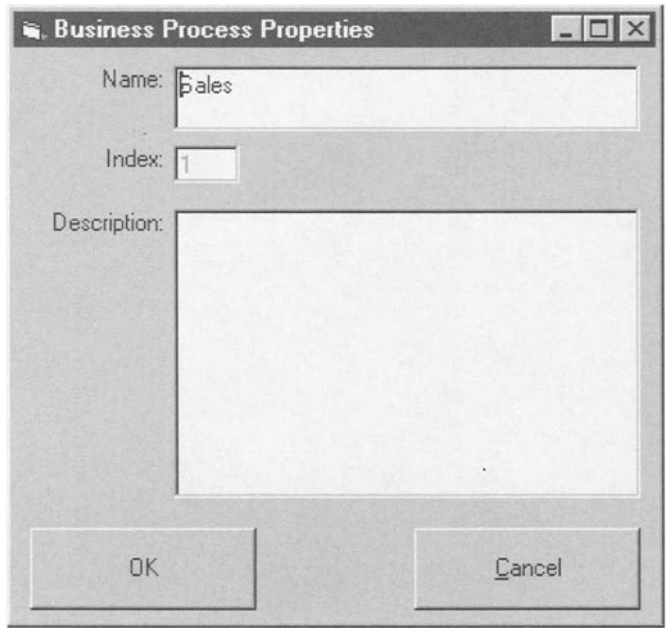


FIGURE 14.28. Business process properties

To delete a business process, select it from the combo box in Figure 14.26, click the **Delete Business Process** button and confirm the deletion operation.

The final form of the activity table can look like the table shown in Figure 14.29.

14.9 Application Model

After completing the development of the tables, we can create the application model from the information in the entity and activity tables. To do so, click **Phases** from the main menu (Figure 14.6) and then choose **Application Model,** or click from the toolbar and you will see the application model, which is shown in Figure 14.30.

Sometimes we need to see two or more tables on the screen at the same time. To do so, open the tables you need by clicking the following icons: click **Window** from the main menu, and then choose **Tile Horizontally** to organize the tables horizontally, **Tile Vertically** to organize them vertically, or **Cascade** to cascade them. The following two figures show examples of two tables on-screen at once.

ActivityTable

			Entity	Internal								External	
			Index	1	2	3	4	5	6	7	8	9	10
Business Process	Work Process	Index	Activity	Top Management	Purchasing Dept	Information Office	Buffet	Accounts Dept	Marketing Dept	Reception	Restaurant	Supplier	
Sales	Purchasing	1	Make Contracts with Suppliers	T2 P1	T9, S2 P1							S9	
		2	Confirm Suppliers Contract	S1 U1	T1, S2 U1, P2	T2						T2	
		3	Create an Order		S2 U2, P3	T2, T4, T8	S4 P3				S8 P3	T2, T4, T8	
		4	Accept Shipment			T9	T9 U3				T9 U3	S9	
		5	Accept Suppliers Invoice	T5	T5 U3, P5	T5		T9, S5				S9	
		6	Confirm Invoice		S2 U5			T2 P6					
		7	Make Suppliers Payment			T5		S5 U6				T5	
	Marketing	8	Create Advertising Material	T6					S6				
		9	Confirm Advertising Material										
		10	Accept Demand										
		11	Send Offer										
		12	Sign Contracts with Customers										

FIGURE 14.29. Activity table

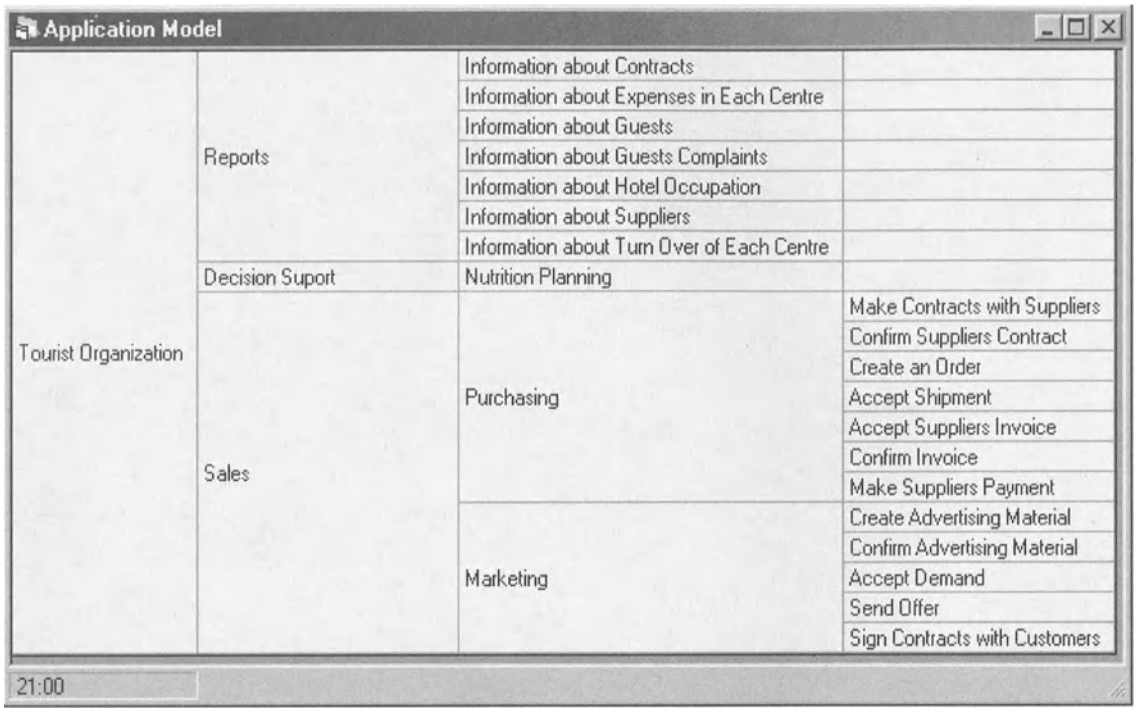
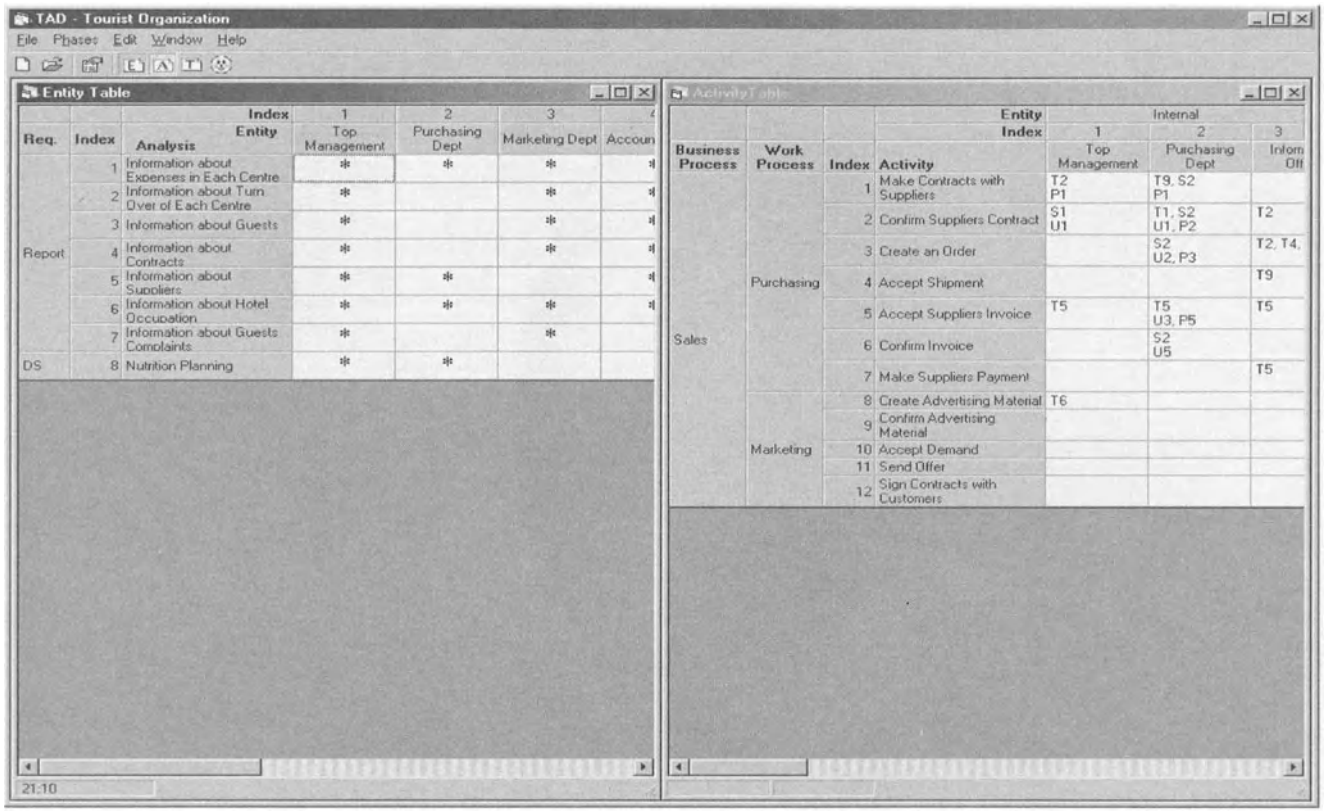
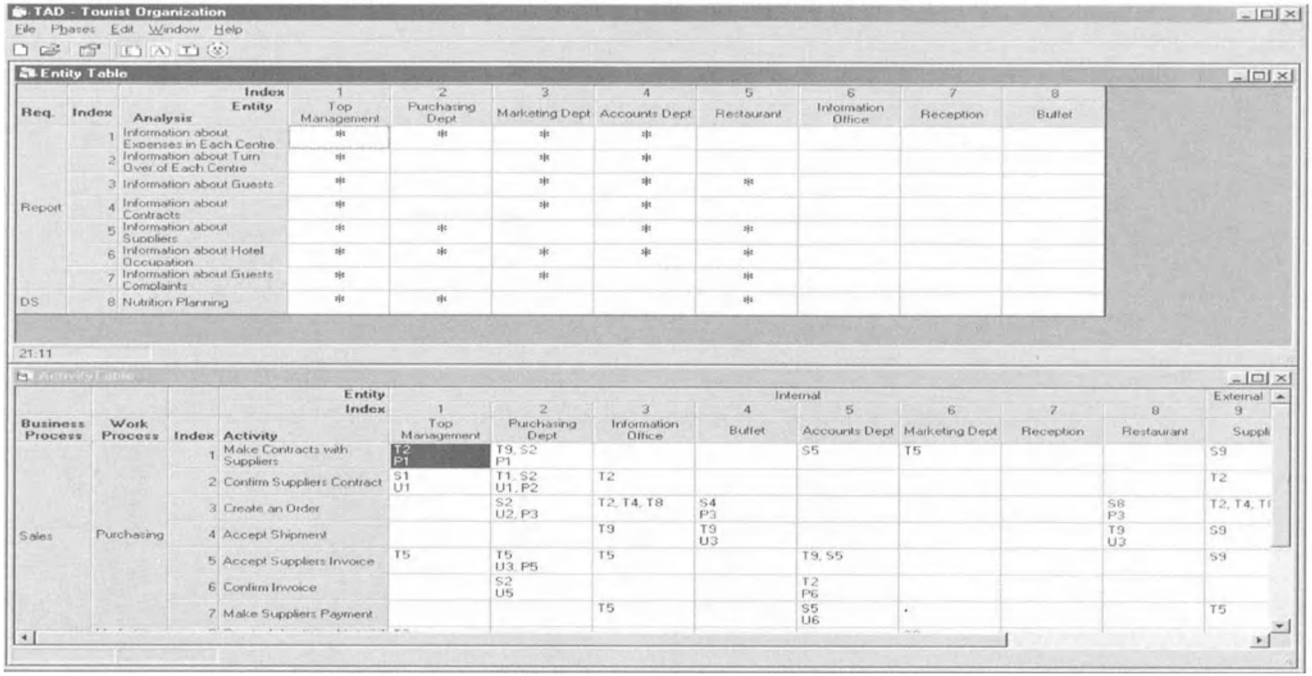


FIGURE 14.30. Application model


TAD - Tourist Organization
File Phases Edit Window Help
Entity Table
Req. | Index | Analysis | Index / Entity | 1 Top Management | 2 Purchasing Dept | 3 Marketing Dept | Accoun
Report | 1 | Information about Expenses in Each Centre | * | * | * | *
2 | Information about Turn Over of Each Centre | * | | * | *
3 | Information about Guests | * | | * | *
4 | Information about Contracts | * | | * | *
5 | Information about Suppliers | * | * | | *
6 | Information about Hotel Occupation | * | * | * | *
7 | Information about Guests Complaints | * | | * | |
DS | 8 | Nutrition Planning | * | * | |
21:10
ActivityTable
Business Process | Work Process | Index | Activity | Entity / Index | Internal 1 Top Management | 2 Purchasing Dept | 3 Inform Off
Sales | Purchasing | 1 | Make Contracts with Suppliers | T2 P1 | T9, S2 P1 |
2 | Confirm Suppliers Contract | S1 U1 | T1, S2 U1, P2 | T2
3 | Create an Order | | S2 U2, P3 | T2, T4,
4 | Accept Shipment | | | T9
5 | Accept Suppliers Invoice | T5 | T5 U3, P5 | T5
6 | Confirm Invoice | | S2 U5 |
7 | Make Suppliers Payment | | | T5
Marketing | 8 | Create Advertising Material | T6 | |
9 | Confirm Advertising Material | | |
10 | Accept Demand | | |
11 | Send Offer | | |
12 | Sign Contracts with Customers | | |



TAD - Tourist Organization

File Phases Edit Window Help

Entity Table

Req.	Index	Analysis	1 Top Management	2 Purchasing Dept	3 Marketing Dept	4 Accounts Dept	5 Restaurant	6 Information Office	7 Reception	8 Buffet
Report	1	Information about Expenses in Each Centre	*	*	*	*				
	2	Information about Turn Over of Each Centre	*		*	*				
	3	Information about Guests	*		*	*	*			
	4	Information about Contracts	*		*	*				
	5	Information about Suppliers	*	*		*	*			
	6	Information about Hotel Occupation	*	*	*	*	*			
	7	Information about Guests Complaints	*		*		*			
DS	8	Nutrition Planning	*	*			*			

21:11

Activity Table

Business Process	Work Process	Index	Activity	1 Top Management (Internal)	2 Purchasing Dept (Internal)	3 Information Office (Internal)	4 Buffet (Internal)	5 Accounts Dept (Internal)	6 Marketing Dept (Internal)	7 Reception (Internal)	8 Restaurant (Internal)	9 Suppli (External)
Sales	Purchasing	1	Make Contracts with Suppliers	T2 P1	T9, S2 P1			S5	T5			S9
		2	Confirm Suppliers Contract	S1 U1	T1, S2 U1, P2	T2						T2
		3	Create an Order		S2 U2, P3	T2, T4, T8	S4 P3				S8 P3	T2, T4, T[illegible]
		4	Accept Shipment			T9	T9 U3				T9 U3	S9
		5	Accept Suppliers Invoice	T5	T5 U3, P5	T5		T9, S5				S9
		6	Confirm Invoice		S2 U5			T2 P6				
		7	Make Suppliers Payment			T5		S5 U6	.			T5

References

Alter S., *Information Systems, A Management Perspective,* Benjamin/Cummings Publishing Co., Menlo Park, California, 1996.

Avison D. E., and Fitzgerald G., *Information Systems Development: Methodologies, Techniques and Tools,* McGraw-Hill Companies, London, 1996.

Booh G., *Object-Oriented Analysis and Design with Applications,* Benjamin-Cummings Publishing Co., Menlo Park, California, 1994.

Burch J. G., and Grudnitski G., *Information Systems, Theory and Practice,* John Wiley & Sons, New York, 1989.

Connor D., *Information System Specification and Road Map,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

Damij T., *An Approach for Information Systems Development,* International Conference on Information Systems Development—ISD '94, Methods & Tools, Theory & Practice, University of Maribor, Slovenia, 1994.

Damij T., *Tabular Based Approach for Systems Development,* International Conference on Organization and Information Systems, University of Maribor, Slovenia, 1995.

Damij T., "Development of a Hospital Information System Using the TAD Method," *Journal of the American Medical Informatics Association,* **5,** no. 2, Mar/Apr 1998.

Damij T., "An Object-Oriented Methodology for Information Systems Development and Business Process Reengineering," *Journal of Object-Oriented Programming,* 13, no. 4, July/August 2000.

Daniels A., and Yeates D., *Basic Systems Analysis,* Pitman Publishing, London, 1990.

Hammer M., and Champy J., *Reengineering the Corporation, A Manifesto for Business Revolution,* HarperBusiness, New York, 1993.

Haryszkiewycz I. T., *Introduction to Systems Analysis and Design.* Prentice-Hall, New York, 1988.

Jacobson I., *Object-Oriented Software Engineering, A Use Case Driven Approach,* Addison-Wesley Publishing Company, Wokingham, England, 1995.

Jacobson I., Ericsson M., and Jacobson A., *The Object Advantage, Business Process Reengineering with Object Technology,* Addison-Wesley Publishing Company, Workingham, England, 1995.

Jacobson I., Booch G., and Rumbaugh J., *The Unified Software Development Process,* Addison-Wesley, Reading, Massachusetts, 1999.

Khoshafian S., and Abnous R., *Object Orientation,* John Wiley & Sons, Inc., New York, 1995.

Langerfors B., Verrijn-Stuart A. A., and Bracchi G., *Trends in Information Systems,* Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1986.

Martin C., and Powell P., *Information Systems, A Management Perspective,* McGraw-Hill Company, London, 1992.

Martin J., *Strategic Data-Planning Methodologies,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

Martin J., *Principles of Object-Oriented Analysis and Design,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993.

Martin J., and Odell J. J., *Object-Oriented Analysis & Design,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1992.

Martin J., and Odell J. J., *Object-Oriented Methods, A Foundation,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995.

Olle T. W., Sol H. G., and Verrijn-Stuart A. A., *Information Systems Design Methodologies: A Comparative Review,* Elsevier Science Publishers B.V., North-Holland, 1982.

Rubin K. S., and Goldberg A., "Object Behavior Analysis," *Communications of the ACM,* **9:**48–62, 1992.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., and Lorensen W., *Object-Oriented Modeling and Design,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991.

Rumbaugh J., Jacobson I., and Booch G., *The Unified Modeling Language Reference Manual,* Addison-Wesley, Reading, Massachusetts, 1999.

Sanders G. L., *Data Modeling,* Boyd & Frase Publishing Company, Danvers, Massachusetts, 1995.

Short K., and Dodd X., Information Engineering with Objects, *ObjectMagazine,* **4:**61–64, 1993.

Vetter M., *Database Design Methodology,* Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

Watson H. G., *Business Systems Engineering: Managing Breakthrough Changes for Productivity and Profit,* John Wiley & Sons, Inc., New York, 1994.

Wrycza S., and Zupancic J., *Information Systems Development,* ISD'96. Methods & Tools, Theory & Practice. Proceedings of the Fifth International Conference, University of Gdansk, Poland, 1996.

Index