AMCIS 1995 Proceedings

Americas Conference on Information Systems (AMCIS)

8-25-1995

Object-Oriented Analysis: A Decision-Driven Approach

Ta-Tao Chuang
Texas Tech University

Surya B. Yadav
Texas Tech University

Object-Oriented Analysis: A Decision-Driven Approach

Ta-Tao Chuang
Surya B. Yadav, Ph.D.
ISQS, College of Business Administration
Texas Tech University

I. Introduction

Recently, many object-oriented analysis and design approaches (OOADs) have been proposed. This research boom may be attributed to the success of applying object-oriented programming (OOP) in embedded systems and systems software. However, object-oriented analysis (OOA) does not seem as successful as object-oriented design (OOD) or OOP [3].

Whereas the extant OOADs claim to perform systems analysis, this goal is seldom fulfilled [3]. Systems analysis consists of two kinds of activities: requirement analysis (problem analysis) and requirement specification (product description) [1]. During the requirement analysis, analysts aim to understand the problem and identify all possible constraints on the problem's solution through observations, interviews, and discussions with experts in the problem domain. The requirement analysis activity analyzes the requirement space of a problem domain. Here, the requirement space is defined as the range of all possible user needs and constraints in a problem domain. Requirement specification, on the other hand, is intended to resolve conflicting views, to eliminate inconsistencies and ambiguities, and to document some particular requirement which describes the expected behavior of the future system. As Hoydalsvik et al [3] indicate, the extant OOADs are target-system oriented. A target-system oriented OOA aims to construct an "object"-oriented system and represents the requirement in a way more consistent with the design issues than with the users' perception of the problem domain. In other words, it concentrates on a solution and not on understanding the problem.

Finding objects and classes is the prevalent trend in the pure OOA. However, as Rubin et al [5] note, there are several problems in searching for objects: 1) The availability of a written requirement specification is usually assumed. Assuming a narrative specification is accessible, an OOAD searches for nouns as objects and for verbs as methods. This approach ignores that a written specification is barely available; even if it is available, ambiguities of text, synonyms, and homonyms are not unusual, 2) there is a strong bias toward the tangible aspects of a problem, and 3) it tends to incorporate all tangible objects of the analysis results.

In order to address these shortcomings, an OOA approach should include a systematic procedure to understand the problem and the organization before finding the objects.

Decision making is a major activity of an organization [6]. This article proposes a decision-driven OOA approach, which consists of a set of well-organized guidelines and procedures, focuses on the understanding of organizations through the analysis of decision making, and helps derive requirement specification in the form of object models. In particular, this article aims to address the following issues:

- What decision making model is more appropriate for understanding the organization?
- What aspects of decision making should be captured for understanding the organization?
- What steps should an OOA approach have?
- What mechanisms can help verify and validate the process of OOA?

We will briefly review several OOADs in the next section. The proposed approach will be discussed in the following section.

II. A Brief Review of Current OOADs

OOADs can be classified into three categories: combinative, adaptive, and pure approaches [4]. A combinative OOAD is a partial life-cycle approach focusing on design and implementation phases. DFD, ER model, and JSD are usually employed to precede OOD. An adaptive approach applies existing techniques to the analysis phase in object-oriented ways [4]. A typical approach is to include operations in entities in ER model, or apply DFD to the object level. A pure approach incorporates features of OOP, such as objects, classes, and inheritance, into the analysis phase. These features are believed to be the necessary conditions for so-called object-orientedness.

The OBA approach by Rubin et al [5] differs from approaches above in that it starts with the identification of service, differentiates analysis from design and does not assume that a written specification is available [3]. However, its step 0--setting the analysis context seems to be more like a project planning. Its step 1--understanding the problem seems to be too narrow to understand the organization.

The proposed approach adopts bottom-up specification, which is similar to OBA. However, the approach provides a broader perspective for understanding the problem.

III. Decision-Driven Object-Oriented Analysis

The DDOOA is based on the information exchange perspective of decision making activities of an organization. Decision processes manipulate objects and the results of manipulation is conveyed as information among decision processes. Thus, by analyzing the information flows among decision processes, an analyst can derive the attributes and methods of the objects involved and construct object model. However, decision making does not exist by itself. Decisions are made to formulate goals or to propose alternatives for accomplishing goals. The effectiveness of decisions must be measured against their relevance to the fulfillment of goals. In order to evaluate and legitimize the decisions, a hierarchical structure of organizational goals is built by using ends-means analysis [6]. Besides, resources must be employed to propose or carry out the alternatives. By resources, we mean both participants in the decision processes and things about which information is exchanged. Thus, resource analysis should precede the analysis of decisions.

Based on the above rationale, the DDOOA method is shown in Figure 1. DDOOA consists of two phases: top-down understanding of the organization and bottom-up specification of the organizational requirements. The understanding phase aims to understand the hierarchical structure of goals, the utilization of resources, and the decision making activities of an organization or a part of it. Hence, this phase has three steps: goal analysis, resource analysis, and decision analysis. A concept of "decision module" is used to thread these steps together. The hierarchical chart of goals, the refined decision modules, and the communication locuses are the main outputs of this phase.

The specification phase is intended to synthesize and document the results of the understanding phase. The phase consists of three steps: message & role analysis, object analysis, and class analysis. Starting with the analysis of the communication locus, the message & role analysis identifies the messages and roles related to a particular actor to form an Actor-Role-Message Table. By analyzing Actor-Role-Message Tables, the object analysis consolidates the attributes and methods into Object-Property-Tables, which are used as the basis for establishing preliminary object models. A preliminary model represents a particular decision module or a part of a user view. To integrate the preliminary models, the vertical and horizontal integration techniques are used in the class analysis to establish external views and an internal class hierarchy. While the process seems to be linear, the nature of iteration must be stressed. Iteration may occur between steps and among activities within a step.

A significant implication of DDOOA is that the results of each step in the specification phase reflect the output of the understanding phase. Especially, the structure of classes/objects reflects, at least implicitly, the structure of organizational goals.

Following the definition of system analysis [1], we regard the goal hierarchy as part of the requirement space directed to improve the effectiveness of the organization. Thus, the class hierarchy representing a particular requirement specification must be organized in a way that reflects the goal hierarchy. Given the goal hierarchy, resources are allocated and utilized to achieve these goals. The collaboration of various resources for a particular decision task will be represented by a preliminary object model. Finally, Actor-Role-Message Tables resulted from the message & role analysis capture the information exchange activities involved in the decision process.

To document the outputs from each step, the following tools are used: Goal Hierarchy Chart, Decision Module, Communication Chart[2], Actor-Role-Message Table, Object-Property Table, Preliminary Object Model, and Object&Class Model.

This approach is different from others because it starts with the analysis of organizational goals. Besides, it separates the understanding issues from the specification issues involved in determining information requirements. The separation should help derive a more reliable and effective object model, which, in turn, should help develop an effective information system.

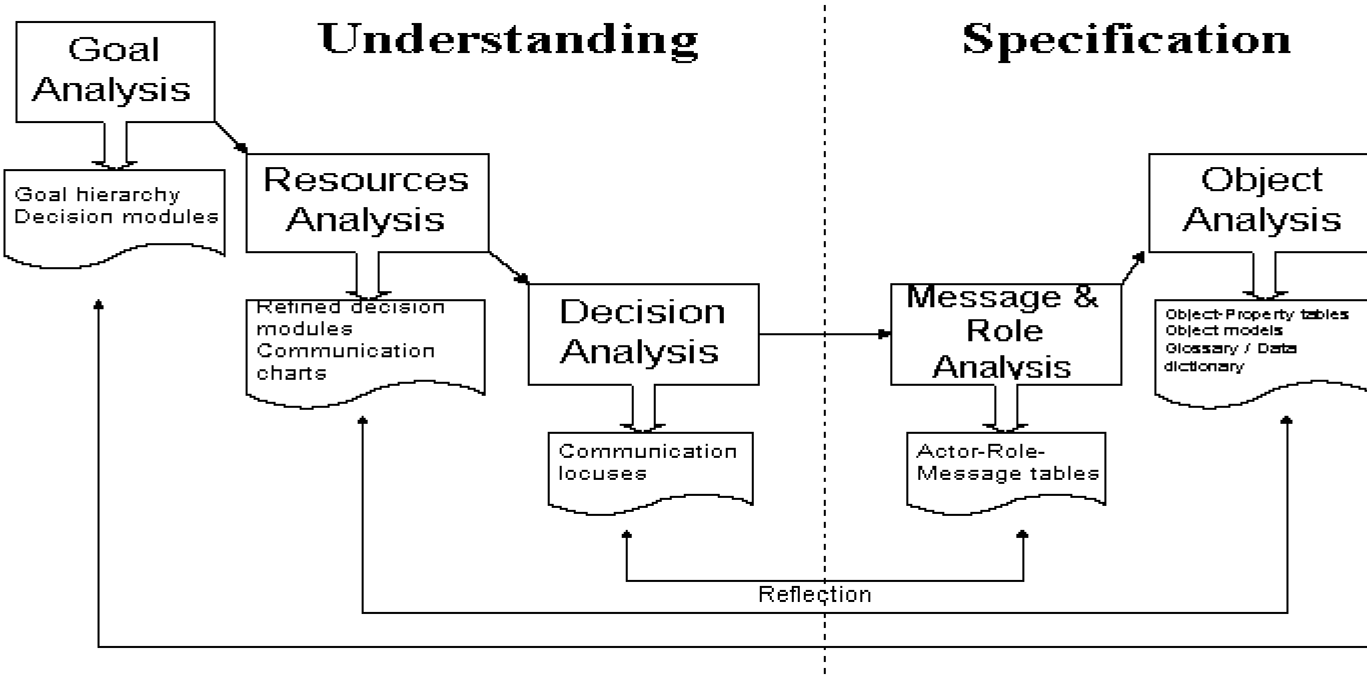


Figure 1 The Decision-Driven Object-Oriented Analysis

IV. An Illustrative Example of DDOOA

Here we discuss an abbreviated case example to illustrate DDOOA. ABC Taxicab, Co. is licensed to operate in the city of Lubbock. It has three departments: Dispatching, Maintenance, and Management Control. It divides the region into eight areas and provides service to calling customers. Cab drivers fill out a "shift work sheet form" for each work-shift. The form has two sections: Pickup and Dropback information.

1. **Goal analysis**
After discussing with the management, the information system development team (ISDT) identified goal hierarchy as shown in Figure 2.

2. **Resource analysis**
Taking "Reduce customer's waiting time" as the goal, ISDT derived a decision module as shown in Figure 3. It shows the decision variables and decision net for carrying out the

means in this module.

3. **Decision analysis**
From the decision net, ISDT proceeded to record the communication locus for this module, which is shown in Figure 4. Information-exchange in the decision net is the focus of the communication locus. In this step, ISDT identified an information-carrying medium, a work sheet form, as an actor.

4. **Message & role analysis**
Communication locuses may be aggregated together based on several principles. In this example, ISDT extracted messages and roles by analyzing the information flows. The messages and roles were recorded in Actor-Role-Message (ARM) Table as shown in Figure 5.

5. **Object analysis**
From ARM Tables, ISDT resolved name and structure (object/attribute) conflicts, normalized objects, and constructed a preliminary object model for each group of decision modules. An example of an object is shown in Figure 6.

6. **Class analysis**
Classes are constructed based on the group of decision modules (external view) and the structures (internal structures). In this example, ISDT constructed an internal structure for the decision module 'Reduce customer's waiting time," as shown in Figure 7.

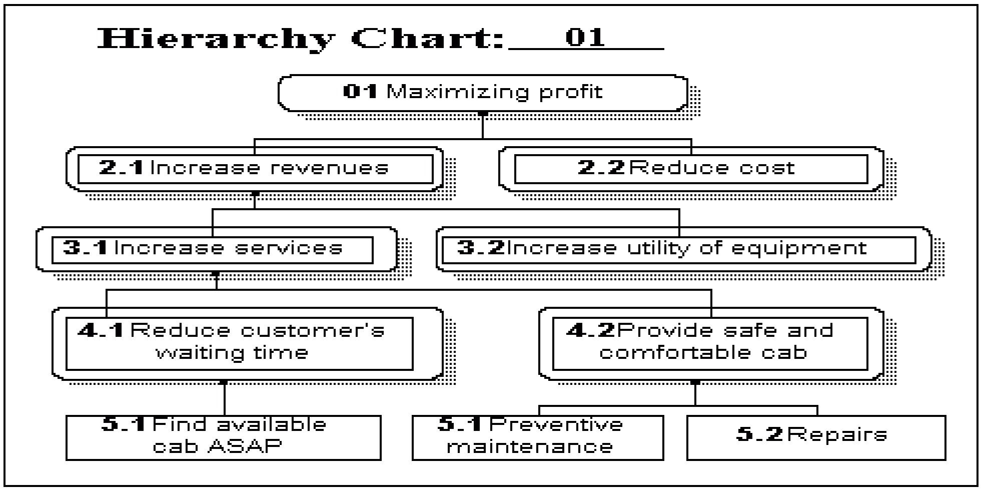
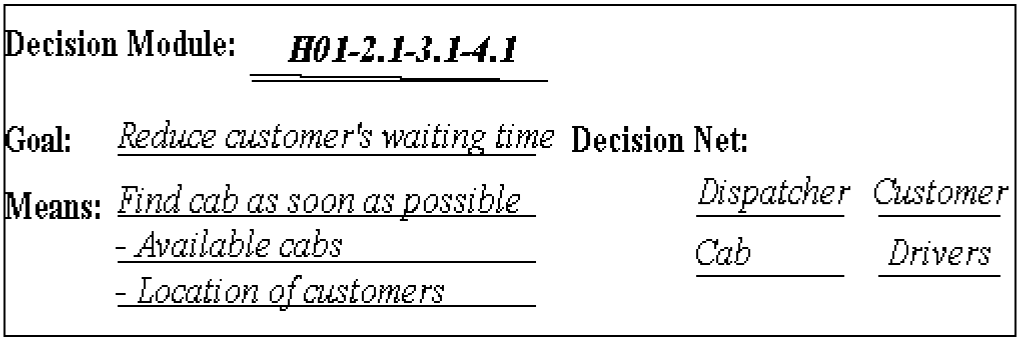

Hierarchy Chart: 01
01 Maximizing profit
2.1 Increase revenues
2.2 Reduce cost
3.1 Increase services
3.2 Increase utility of equipment
4.1 Reduce customer's waiting time
4.2 Provide safe and comfortable cab
5.1 Find available cab ASAP
5.1 Preventive maintenance
5.2 Repairs

Figure 2 Goal Hierarchy



Decision Module: *H01-2.1-3.1-4.1*

Goal: *Reduce customer's waiting time* Decision Net:

Means: *Find cab as soon as possible*

- Available cabs

- Location of customers

Dispatcher *Customer*

Cab *Drivers*

Figure 3 A Decision Module

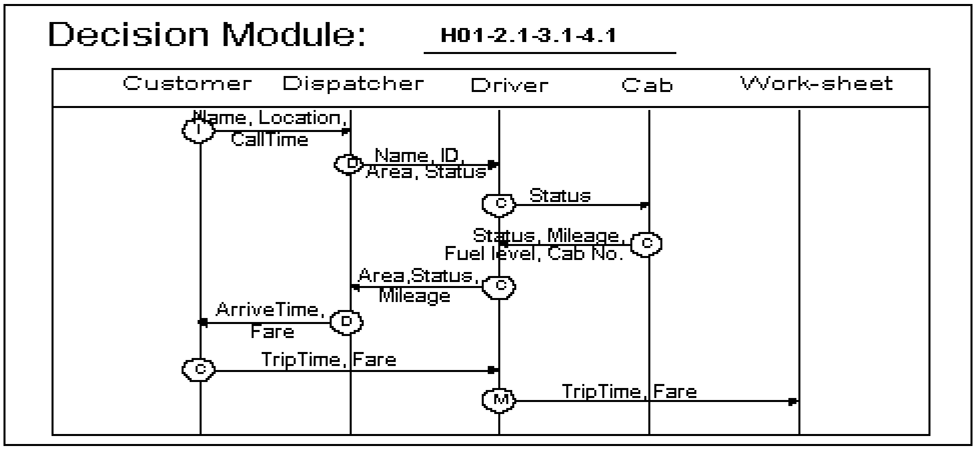
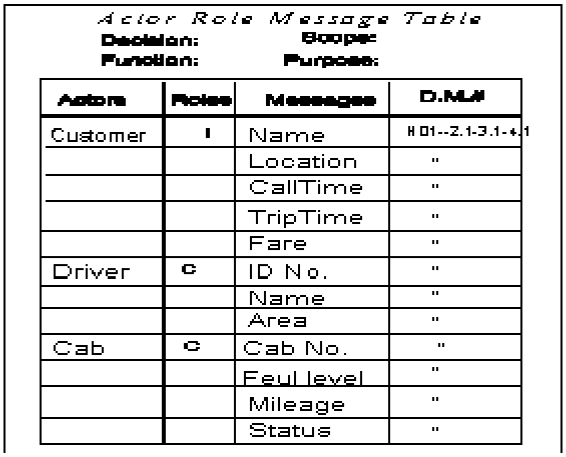


Figure 4 A Communication Locus



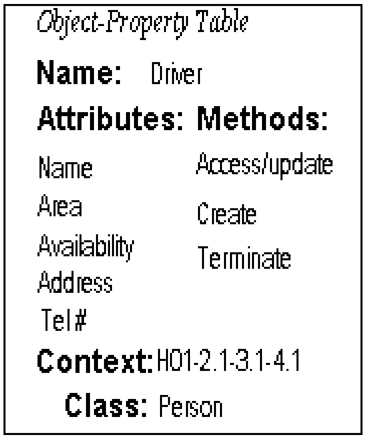
Actor Role Message Table

Decision: Scope:

Function: Purpose:

Actors	Roles	Messages	D.M.#
Customer	I	Name	H01--2.1-3.1-4.1
		Location	"
		CallTime	"
		TripTime	"
		Fare	"
Driver	C	ID No.	"
		Name	"
		Area	"
Cab	C	Cab No.	"
		Feul level	"
		Mileage	"
		Status	"

Figure 5 An ARM Table



Object-Property Table

Name: Driver

Attributes:	Methods:
Name	Access/update
Area	Create
Availability	Terminate
Address	
Tel#	

Context: H01-2.1-3.1-4.1

Class: Person

Figure 6 An Object

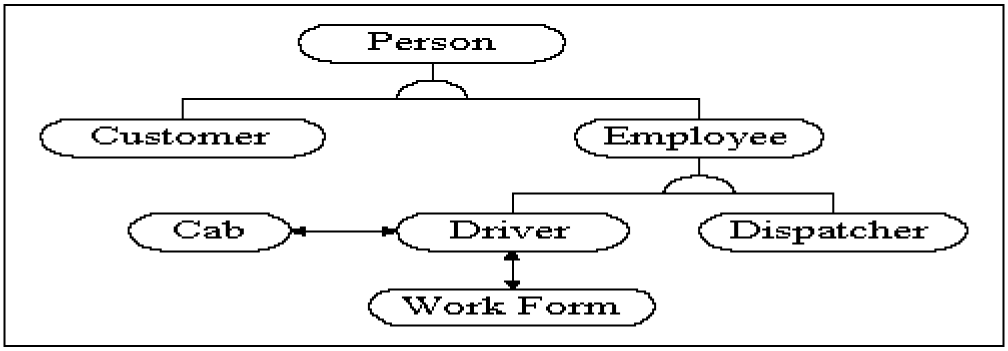


Figure 7 An Object&Class Model

Selected References

[1] Davis, Alan M., Software Requirement - Analysis & Specification. Prentice Hall, 1990.

[2] Eilon, Samuel. Taxonomy of Communication. Admin. Sciences Quarterly. 13(2), 1968. pp. 266-288.

[3] Hoydalsvik, Geir Magne *et al*. On the Purpose of Object-Oriented Analysis. OOPSLA '93, pp. 240-255.

[4] Monarchi, David E. *et al*. A Research Typology for Object-Oriented Analysis and Design. CACM, 35:9, Sep., 1992.

[5] Rubin, Kennehth S. *et al*. Object Behavior Analysis. CACM, 35:9, Sep., 1992.

[6] Simon, Herbert A. Administrative Behavior, 3rd Edition. New York: The Free Press, 1976.

(Note: A longer version of the paper is available)