ELSEVIER

Information & Management 32 (1997) 303–315

INFORMATION & MANAGEMENT

Applications

Modeling information architecture for the organization

Shouhong Wang*

Faculty of Business, University of New Brunswick, Saint John, NB E2L 4L5, Canada

Accepted 20 May 1997

Abstract

The issue of information architecture (IA) for organizations has recently received considerable attention in IS development. However, as yet little research has been reported on modeling IA using a systematic approach. This paper describes an object-oriented method for modeling it. The proposed method extends the traditional concept of IS analysis into the context of contemporary information technology (IT), and is useful for planning IT-enabled business process reengineering for the organization. © 1997 Elsevier Science B.V.

Keywords: Information architecture; Information systems analysis and design; Object-oriented approach

1. Introduction

An information architecture (IA) is a high-level map of the information requirements of an organization [3]. IA (or information systems (IS) architecture) is becoming one of the most important issues in IS development [1]. This is due to the extraordinary rate of growth of information technology (IT) and the wave of business process reengineering. In the information era, traditional monolithic computer systems no longer support cross-functional business processes. Organizations are seeking opportunities to exploit new IT, such as office automation, imaging documentation, networks, and client–server technology to redesign business processes. The analysis and design of high-level architecture for the enterprises is important in assessing the business, technology, and information needs of the organization.

The analysis and design of an IA are the tasks of information specialists in the organization [30]. From the point of view of modeling IA, a natural language is universal, in the sense that it can describe anything important. Yet natural languages have potential ambiguities. Especially, in the system implementation stages, natural language descriptions do not provide measures to map the real world to the computer world. Conventional systems analysis methods, such as the data flow diagram method [8, 14], entity-relationship method [6] and the combination of these methods [25], emphasize descriptions of functional and data requirements within the context of monolithic computing systems. On the other hand, contemporary IS analysis must present many global system aspects including special techniques (e.g. imaging and EDI), locations of data and applications (company-wide and locally managed), and adopted system standards and rules. Although there have been many discussions about IA, there is no theory of its modeling. One of the problems of research in IS development is the failure to integrate

*Corresponding author. Tel.: 001 506 6485731; fax: 001 506 6485587; e-mail: SWANG@UNB.CA.

PII S-0378-7206(97)00025-6

the modeling of IA with systems analysis. This lack of integration may cause difficulties in applying a uniform technique to analyzing the global IA and the specific applications for a reengineered business process.

In recent years, the object-oriented approach has received attention from the IS communities [2, 16, 32, 36]. A recent information industry survey indicated that the adoption rate of object-oriented methods is increasing dramatically. Organizations that embraced the approach have experienced significant cost savings in the systems development area [27]. Nevertheless, little research into the integration of modeling IA with the object-oriented method can be found. This paper presents a framework for modeling IA for an organization.

Although IS architecture is related to information and business strategy, our research was limited to modeling IS architecture and should not be construed as presenting a strategic planning methodology. While the development of a strategy for organizations which could manifest itself in architectural expression is an important subject, it is outside this study.

2. Object-oriented approach in modeling IS

The object-oriented approach [23, 28, 37] has become popular in recent years; however, it is still in a period of development. Using the object-oriented approach, analysts model the system being investigated by identifying a set of objects in conjunction with their attributes (i.e. data) and methods (i.e. internal operations and messages) that manipulate the object data or request services from other objects. The encapsulation of the attributes and exclusive methods form the basis for treating the attributes and methods as a whole. Objects are grouped into classes which have common properties. Classes are organized into hierarchies in which the subclasses inherit properties, including data definitions and methods. Interactions between objects are handled by means of message sending. The dynamic relationships between objects are built into the descriptions of the classes through the definition of message sending. All of these characteristics make the object-oriented approach more effective than the traditional data flow diagram method in IS development (see [19, 26] for a more detailed discussion). More importantly, the model represented in the paradigm can be implemented by a computer-based IS using object-oriented programming without the requirement of a creative system logical design phase [13].

The methodology of object-oriented analysis is far from mature. A serious criticism of current work is the piecemeal fashion of object-oriented systems development. From the point of view of IA, research should investigate an extension of the object-oriented approach into modeling IA at a macro level.

There are a variety of tools and techniques for object-oriented analysis (see a survey in [10]). Coad–Yourdon's method [7] was selected as a base for this study because of its simplicity. Nevertheless, two significant modifications were made to it. First, cardinalities (0, 1, *N*) between classes were omitted because they do not play an important role in modeling macro IA. The second modification was that data transmitted by messages between object classes were explicitly annotated. The concept of data flows passing in the object-oriented paradigm is quite different from data flows in the structured data flow diagram in that data flows must be associated with the messages between objects. The elements in object-oriented modeling are:

1. *Attributes*: Encapsulated data descriptions of the object class.
2. *Operations*: Processes that apply to the object class. There are two types of operations:
   - 2.1. *Method*: An operation which manipulates the encapsulated data.
   - 2.2. *Message*: An operation procedure which requests service from other object(s). In the object-oriented paradigm, message sending from one object class to another makes dynamic connections between the object classes.
3. *Data flow*: A group of data elements that are associated with a message and specify the communication between the objects.
4. *Inheritance*: In a hierarchical relationship between object classes, subclasses inherit properties, including data definitions and operations, from their superclasses. Inheritance results in static connections between object classes.