# METAGRAPHS AND THEIR APPLICATIONS

# INTEGRATED SERIES IN INFORMATION SYSTEMS

*Series Editors*

Professor Ramesh Sharda
*Oklahoma State University*

Prof. Dr. Stefan Voß
*Universität Hamburg*

**Other published titles in the series:**

E-BUSINESS MANAGEMENT: *Integration of Web Technologies with Business Models*/ edited by Michael J. Shaw
VIRTUAL CORPORATE UNIVERSITIES: *A Matrix of Knowledge and Learning for the New Digital Dawn/* Walter R.J. Baets and Gert Van der Linden
SCALABLE ENTERPRISE SYSTEMS: *An Introduction to Recent Advances*/ edited by Vittal Prabhu, Soundar Kumara, Manjunath Kamath
LEGAL PROGRAMMING*: Legal Compliance for RFID and Software Agent Ecosystems in Retail Processes and Beyond*/ Brian Subirana and Malcolm Bain
LOGICAL DATA MODELING: *What It Is and How To Do It*/ Alan Chmura and J. Mark Heumann
DESIGNING AND EVALUATING E-MANAGEMENT DECISION TOOLS: *The Integration of Decision and Negotiation Models into Internet-Multimedia Technologies*/ Giampiero E.G. Beroggi
INFORMATION AND MANAGEMENT SYSTEMS FOR PRODUCT CUSTOMIZATION/ Thorsten Blecker et al.
MEDICAL INFORMATICS: *Knowledge Management and Data Mining in Biomedicine/* edited by Hsinchun Chen et al.
KNOWLEDGE MANAGEMENT AND MANAGEMENT LEARNING: *Extending the Horizons of Knowledge-Based Management*/ edited by Walter Baets
INTELLIGENCE AND SECURITY INFORMATICS FOR INTERNATIONAL SECURITY: *Information Sharing and Data Mining*/ Hsinchun Chen
ENTERPRISE COLLABORATION: *On-Demand Information Exchange for Extended Enterprises*/ David Levermore and Cheng Hsu
SEMANTIC WEB AND EDUCATION/ Vladan Devedžić
INFORMATION SYSTEMS ACTION RESEARCH: *An Applied View of Emerging Concepts and Methods*/ Ned Kock
ONTOLOGIES: *A Handbook of Principles, Concepts and Applications*, edited by Raj Sharman, Rajiv Kishore and Ram Ramesh

# METAGRAPHS AND THEIR APPLICATIONS

**Amit Basu**

Charles Wyly Professor of Information Systems
ITOM Department
Edwin L. Cox School of Business
Southern Methodist University
Dallas, TX 75275, USA

**Robert W. Blanning**

Professor of Management
Owen Graduate School of Management
Vanderbilt University
Nashville, TN 37203, USA

Springer

Amit Basu
Southern Methodist University
Dallas, TX, USA

Robert W. Blanning
Vanderbilt University
Nashville, TN, USA

Library of Congress Control Number: 2006930395

ISBN-10: 0-387-37233-4
ISBN-13: 978-0387-37233-4

e-ISBN-10: 0-387-37234-2
e-ISBN-13: 978-0387-37234-1

Printed on acid-free paper.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

# CONTENTS

Preface vii

Chapter 1: Graphs, Hypergraphs, and Metagraphs 1
1. Graphs and Data Visualization 1
2. Graph Structures 4
3. Metagraph Theory (Part I) 9
4. Applications of Metagraphs (Part II) 11

**Part I. Metagraph Theory 13**

Chapter 2: The Algebraic Structure of Metagraphs 15
1. Formal Representation of a Metagraph 15
2. The Incidence and Adjacency Matrices 17
3. Identifying Metapaths 23

Chapter 3: Connectivity Properties of Metagraphs 27
1. Dominant Metapaths 27
2. Cutsets and Bridges 29

Chapter 4: Metagraph Transformations 33
1. Hierarchical Abstraction Using Projection 33
2. The Inverse Metagraph 46
3. The Element Flow Metagraph 48

Chapter 5: Attributed Metagraphs 53
1. Qualitative Attributes 53
2. Quantitative Attributes 55
3. Conditional Metagraphs 55
    3.1. Projections in Conditional Metagraphs 58
    3.2. Connectivity and Redundancy 61

Chapter 6: Independent Sub-Metagraphs 65

**Part II. Applications of Metagraphs 69**

Chapter 7: Metagraphs in Model Management 71
1. Models as Metagraphs 72
2. Model Selection and Integration 74

3. Hierarchical Modeling 76
4. Assumptions in Model Bases 86

Chapter 8: Metagraphs in Data and Rule Management 97
1. Representing Rule Bases as Metagraphs 99
2. Integrating Rules, Models and Data 105
3. Discovering Implicit Integrity Constraints 111
4. Metagraph Models of Decision Support Systems 115

Chapter 9: Metagraphs in Workflow and Process Analysis 117
1. Representing Workflows and Processes with Metagraphs 118
2. Views of Workflows 123
3. Analysis of Information Interactions 128
4. Analysis of Task Interactions 131
5. Analysis of Resource Interactions 133
6. Interactions among Different Types of Components 136
7. Synthesis of Processes 137
8. Decomposition of Processes and Implications for Organizational Design 143
9. Representing Time-Critical Workflows with Attributed Metagraphs 146

Chapter 10: Conclusion 153
1. The Metagraph Modeling Process 153
2. Towards a Metagraph Workbench 156
3. Metagraphs and Social Networks 158
4. And Finally 160

References 161

Index of Definitions 165

Index 167

# PREFACE

An important concept in the design of many information processing systems – such as transaction processing systems, decision support systems, and workflow systems – is that of a graph. In its simplest form a graph consists of a set of points (or nodes) and a set of ordered or unordered pairs of nodes (or edges). If the pairs of nodes are unordered, the graph is called a simple graph, and if they are ordered, the graph is called a directed graph, or digraph. In both cases, the graph represents a network through which materials, people, information, etc. can flow. The difference is whether the flow is restricted to one direction or whether there is no such restriction.

Simple graphs and digraphs allow for the construction of a variety of diagrammatic system design tools – such as entity-relationship diagrams, functional dependency diagrams, data flow diagrams, Petri nets, semantic nets, and the like. We note that most of these tools are representational, not analytical. That is, they provide a convenient and visually appealing format for illustrating information infrastructures, while allowing any subsequent analyses to be performed by the user.

Another problem with such graphical structures is that they usually associate individual information elements and not sets of elements. Yet in many cases it is necessary to associate sets of elements – such as multiple attributes in data relations, multiple variables in decision models, multiple logical variables in decision rules, and multiple documents in workflow systems. Furthermore, it may be necessary to integrate data relations, decision models, decision rules, and workflows into an integrated information processing system. Two multiple-element structures, hypergraphs and higraphs, allow a few such representations, but they have their limitations.

A recently developed graphical structure that overcomes the limitations and shows great promise in modeling information processing systems is a *metagraph*. Metagraphs are more complex than the graph structures described above, but they allow representation and analysis of more complex systems. Although there is a substantial literature on metagraphs, this is all in the form of journal articles and papers in conference proceedings. There have been no books presenting a comprehensive picture of the foundations of metagraphs and the applicability of these foundations to the design of information processing systems. This book attempts to fill that gap by providing a single and comprehensive treatment of metagraphs.

We begin with a brief introduction to metagraphs. A metagraph is a collection of directed set-to-set mappings. Although this is a simple definition, it leads to several powerful theoretical results and several interesting applications. We then present the material in this book in two parts. The first develops the theoretical results. Although we will include diagrams for purposes of exposition, the emphasis will be on the development of a metagraph algebra. This is a matrix algebra defined over the elements and edges of a metagraph, resulting in incidence and adjacency matrices. This in turn will lead to a more sophisticated view of paths in a metagraph, resulting in the concept of a metapath. We will also be concerned with (1) certain transformations of metagraphs, especially the projection of a metagraph to produce a simpler metagraph, (2) conditional metagraphs, in which the calculations performed early in a metagraph process determine the structure of the later part of the metagraph, and (3) submetagraphs that are largely independent of their containing metagraphs.

In the second part of the book we will examine four promising applications of metagraphs. The first is the modeling of data relations, each of which is viewed as a mapping from a set of key elements to a set of content elements. The second is the modeling of decision models, each of which is viewed as a mapping from a set of input variables to a set of output variables. The third is the modeling of decision rules, each of which is viewed as a mapping from a set of logical antecedent variables to a set of logical consequent variables. The fourth is the modeling of workflow tasks, each of which is viewed as a mapping from a set of input documents to a set of output documents. We will apply the theoretical results of the first part of the book to the application areas of the second part.

We conclude this book by briefly examining several possible extensions of this work. Of special interest is the structuring of the metagraph modeling process, which may enhance the body of work on systems analysis and design (and also software engineering), the development of a metagraph workbench to support such a process, and the possible application of our results, suitably enhanced, to social networks.

## Chapter 1

# GRAPHS, HYPERGRAPHS, AND METAGRAPHS

An important concept in the design of many information processing systems – such as transaction processing systems, decision support systems, project management systems, and workflow systems – is that of a graph. In its simplest form, a graph consists of a set of elements (or nodes) and a set of ordered or unordered pairs of nodes (or edges). A substantial body of theoretical and applied research on various types of graphs has made it possible to develop powerful analytical tools for systems design. The purpose of this chapter is to summarize some of the existing graph-based tools used in this area, and the purpose of this book is to present a new graphical structure, called metagraphs, that enhances existing structures and overcomes some of their disadvantages.

We begin in Section 1 by describing some of the traditional uses of graphs – tools for visualizing relationships between data elements, data aggregates, data structures, files, documents, and the like. Specifically, we examine entity-relationship diagrams, functional dependency diagrams, data flow diagrams, and semantic nets. In each of these cases the purpose of the graph is to display the structure of data so that a user can infer possible relationships of interest. Although it may be possible to use these structures as the basis of an analytical model, the purpose of the diagram/network is to assist the user's intuition in understanding important relationships among data elements, aggregates, etc.

The three remaining sections of this chapter summarize the remainder of the book. First, in Section 2 we review graph structures related to metagraphs – especially, simple graphs, directed graphs, hypergraphs, higraphs, and Petri nets. Then in Section 3 we provide a brief overview of metagraph theory, which we will examine in more detail in Chapters 2–6. Finally, in Section 4 we provide a brief overview of metagraph applications, which we will examine in more detail in Chapters 7–10. The ideas in this book are based on a set of papers published by the authors in a variety of journals and conference proceedings. These papers are included in the references at the end of the book.

## 1. GRAPHS AND DATA VISUALIZATION

We begin by describing three types of graphical structures, used in three types of diagramming conventions. The first type of diagramming convention concerns the static nature of stored data – that is, the structure of databases. We will examine two approaches to diagramming databases. The first of these is based on the assumption that the data base is in relational form, so that the files (tables, relations) describe both the entities about which data is recorded, such as suppliers and the parts they supply, and the relationships among the entities. This results in the entity-relationship approach to data, illustrated in Figure 1.1.

In Figure 1.1 the supplier relation consists of two data attributes: the ID of the supplier, which is the key attribute, and the location of the supplier, which is the content attribute. Similarly, the part relation consists of two data attributes, the part ID, which is the key, and the weight of the part, which is the content. Finally, there is a many-to-many relationship between suppliers and parts, resulting in an intersection relation with a compound key (i.e., the two IDs), along with a content element (the price that the particular supplier charges for the particular part). Of course, if all suppliers charged the same price for any particular part, then the price would be a content attribute in the part relation. Thus, the structure of the data base depends on the structure of the real world about which data is being stored and/or the business rules of the organization.

Yet another approach to diagramming data bases is to focus on the functional dependencies among the data attributes. This is illustrated by the functional dependency diagram illustrated in Figure 1.2. We can see that the sup-

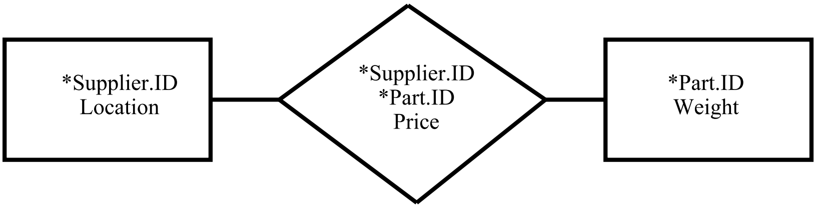

Figure 1.1. An entity-relationship diagram.

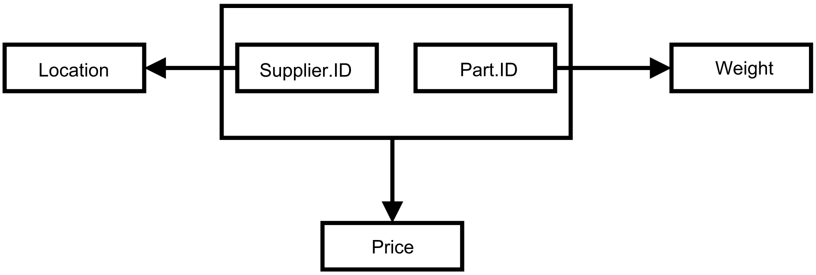

Figure 1.2. A functional dependency diagram.

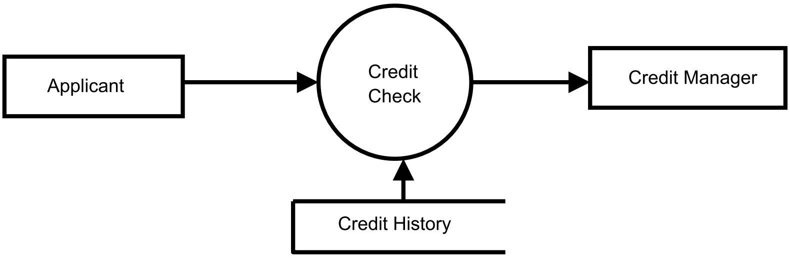

Figure 1.3. A data flow diagram.

plier ID uniquely determines location, the part ID determines weight, and the two together determine price. Both diagrams denote the same information, and both can be augmented with additional semantic information, such as the supply relationship in this case (i.e., the fact that suppliers supply parts).

The two structures outlined above describe the static structure of data in an organization, but they do not describe the dynamic nature of data as information flows throughout an organization. A common way of doing this is with a data flow diagram, illustrated in Figure 1.3. We assume that an applicant for credit submits an application, a credit check is performed, using a credit history file, and a report is sent to a credit manager. The diagram illustrates the relationships among the sources (applicant) and destinations (credit manager) of data, along with credit check process and the credit history file. This is a top-level (or Level 0) diagram, which might then be decomposed into lower-level (Levels 1, 2, etc.) diagrams, and the processes are usually numbered to make it apparent how the more detailed processes relate to each other.

Finally, we look at another type of data structure, one that describes relationships among concepts. This is captured by a semantic net, illustrated in Figure 1.4. The semantic net captures relationships among the concepts, such as instance, subclass, and others (e.g., a mouse eats cheese) and allows concepts to inherit properties from other concepts. For example, since a mouse is a mammal and Mickey Mouse is a mouse, then Mickey Mouse is a mammal. In addition, Mickey Mouse eats cheese and is an animal.

In summary, simple diagrammatic frameworks, based on graphical structures, can be used to illustrate relationships among items of interest by means of simple visualization. This allows analysts to structure the systems they must deal with and draw inferences about the behavior of these systems. But graphs can serve not only as a foundation for visualization-based inference, but they can also serve as a foundation for algebraic operations that allow for more rigorous calculation of properties.

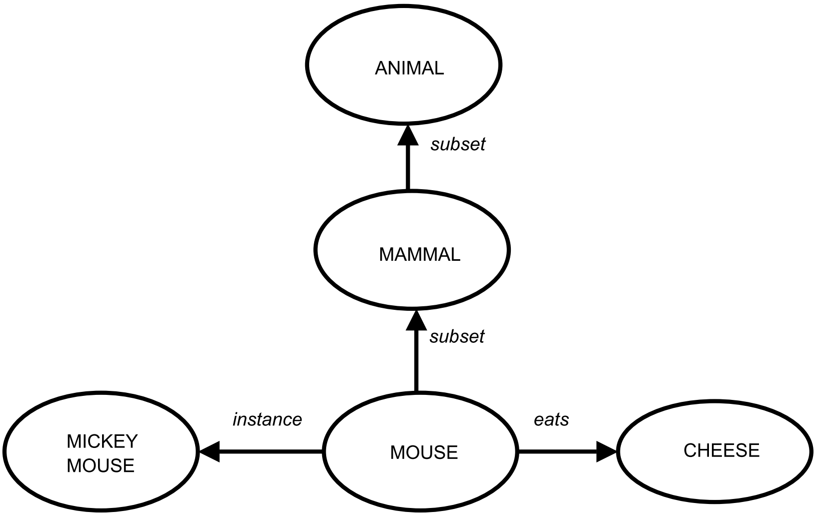

Figure 1.4. A semantic net.

## 2. GRAPH STRUCTURES

We next review two traditional graph structures (ssimple graph and directed graphs), three more recent structures (hypergraphs, higraphs, and Petri nets), and finally metagraphs. To illustrate these structures, consider a system in which there are three input variables:

Pri = the sale price of a product,
Vol = the sales volume,
Wage = the prevailing wage rate.

There are also two intermediate variables:

Rev = the revenue realized, which depends on the price and the volume,
Exp = the expense incurred, which depends on the volume sold and the wage rate.

Finally there are two output variables:

Prof = the realized profit,
Notes = notes payable as a result of borrowings to cover expenses.

We assume that Pri and Vol determine Rev, Vol and Wage determine Exp, Rev and Exp determine Prof and Notes, and Exp determines Notes. We note that Notes can be determined either from Rev and Exp (along with Prof) or directly from Exp. Thus, there is a limited amount of redundancy in this set of calculation procedures, which may give the user a limited amount of discre-

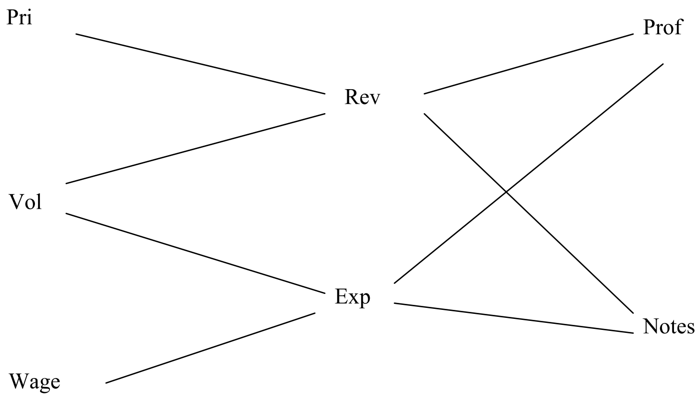

Figure 1.5. A simple graph.

tion in implementing them; however, this could lead to inconsistencies in the results.

The traditional graph structures for describing these variables and the relationships between them are simple graphs and directed graphs (Berge, 1985). A simple graph is illustrated in Figure 1.5. It consists of seven nodes, one corresponding to each of the seven variables defined above, along with seven (unordered) pairs of nodes, one for each of the edges (line segments) in the figure. Thus, we can see that there is a direct relationship between, for example, Price and Revenue, although the direction of the relationship is not clear. There is also an indirect relationship between Price and Profit; Price does not directly determine Profit, but Price does determine Profit through Revenue. The sequence of edges connecting Price to Profit is called a path. The problem is that there is also a path connecting Price to Volume, with Revenue as an intermediate node. Since we do not know the directions of the relationships, we might also conclude that Price determines Volume through Revenue, which is not the case.

A more revealing graph is a directed graph, or digraph, in which the edges are ordered pairs of nodes, represented visually by arrows. The edges of a directed graph describe the directions of the relationships among variables (nodes). This is illustrated in Figure 1.6. We can see that Price is necessary to determine Revenue, and not vice versa, and there is a path Price to Profit through Revenue. But now there is another problem. The directed graph reveals that Price and Volume determine Revenue, but it is not clear whether either Price and Volume alone are sufficient to determine Revenue, or whether both are needed. This can be overcome with AND/OR graphs, in which arcs spanning the directed edges specify whether the relationships are conjunctive or disjunctive. However, AND/OR graphs are clumsy for large numbers of

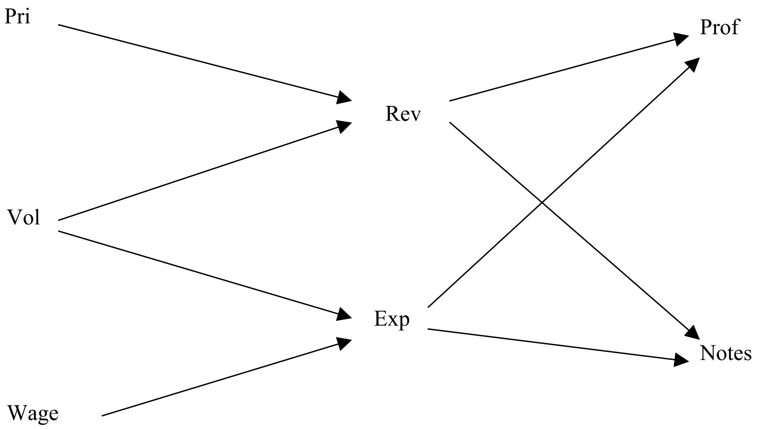

Figure 1.6. A directed graph.

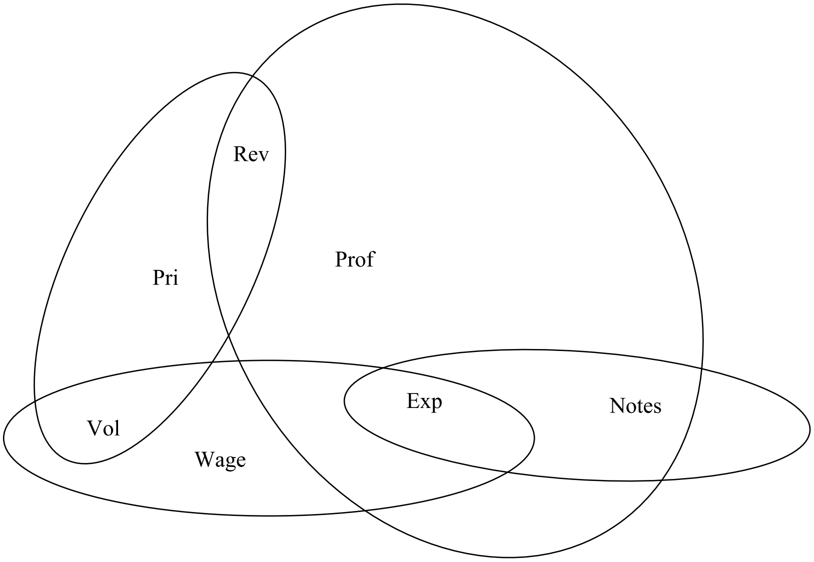

Figure 1.7. A hypergraph.

nodes and edges (i.e., variables and relationships), and a less complicated approach is needed.

A partial solution is offered by hypergraphs (Berge, 1989). In a hypergraph each edge is a set of one or more elements, which allows us to represent relationships among multiple elements. This is illustrated in Figure 1.7. We can see, for example, that Price, Volume, and Revenue are all part of a single relationship. As before, we can identify paths consisting of sequences of hypergraph edges connecting variables such as Price and Profit. The problem, as

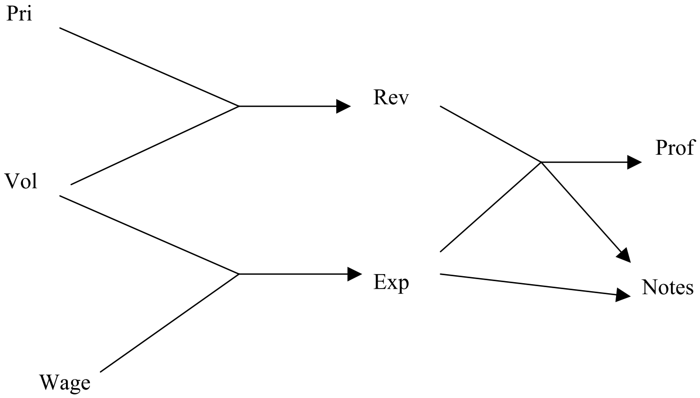

Figure 1.8. A directed hypergraph.

with simple graphs, is that the edges do not capture any sense of direction. For example, the hypergraph does not tell us whether Price and Volume are used to determine Revenue or whether some other relationship is intended – for example, that Price and Revenue are used to determine volume.

A solution is to combine the directed character of a digraph with the multivariate character of a hypergraph, resulting in a directed hypergraph, as illustrated in Figure 1.8 (Ramaswami, Sarkar and Chen, 1997). In Figure 1.8, the set {Rev, Exp} is called the tail of the edge and the set {Prof, Notes} is called the head of the edge between them. Using directed hypergraphs, we can define relationships between sets of variables, such as {Pri, Vol, Wage} and {Prof}.

Another structure is higraphs – or hierarchical graphs (Harel, 1988). A higraph is a collection of "blobs", each of which may contain elements and sub-blobs, which may in turn contain certain elements and other sub-blobs, etc. (Figure 1.9). Higraphs have the advantage of flexibility – for example, edges can originate and terminate within blobs. But this comes at the expense of analytical complexity. A related structure is statecharts (Harel, 1987), which can be used to represent sequences of calculations.

Another dynamic structure is Petri nets (Peterson, 1981). Petri nets are directed graphs containing of two types of nodes – places and transitions (Figure 1.10). Places may contain tokens, and when all of the places leading into a transition are enabled (i.e., contain at least one token), the transition may fire, removing a token from each of the places leading into it and placing a token in each place leading out of it. The process in Figure 1.10 begins with the transitions on the left side of the net firing in either order, removing the tokens from the Pri, Vol, and Wage places. A token would now appear in the Rev place and two tokens would appear in the Exp place. Now the two transitions on the right side of the net can fire, again in either order, placing tokens in Prof and Notes places. At this point no further transitions can fire and the process terminates.

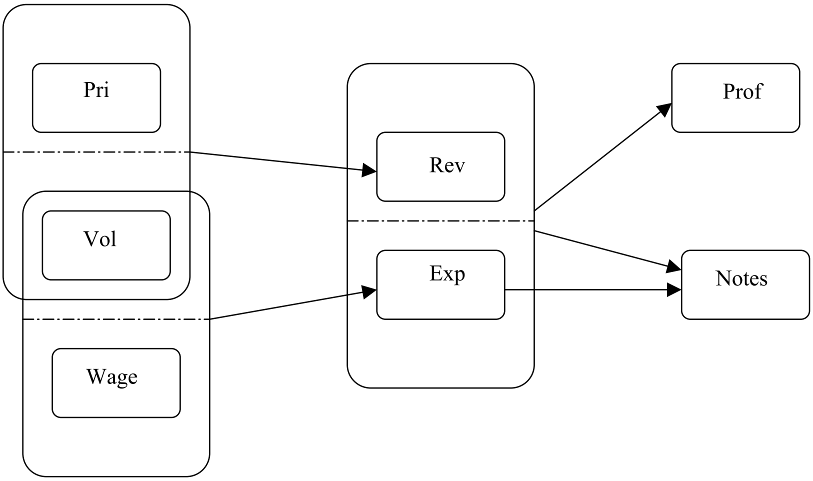

Figure 1.9. A higraph.

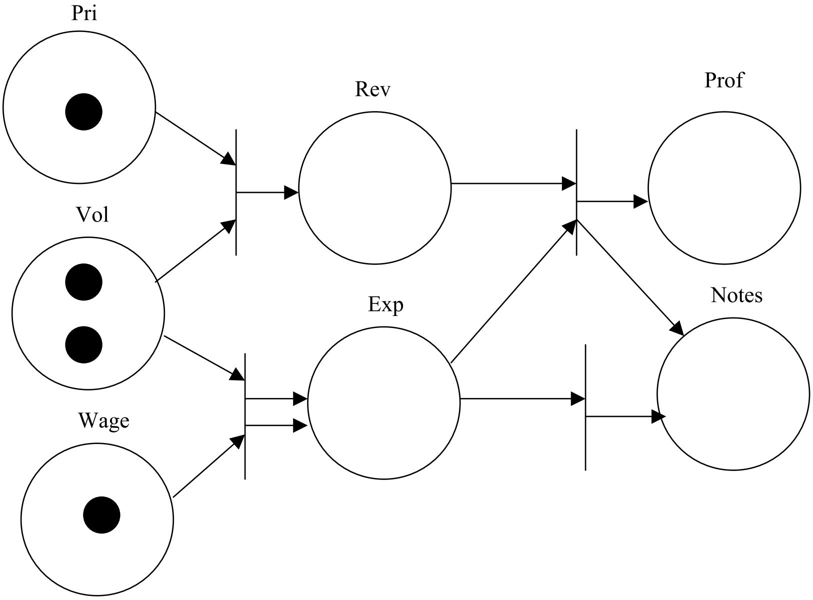

Figure 1.10. A Petri net.

Finally, we introduce the structure on which this book will focus – that of metagraphs, illustrated in Figure 1.11. A metagraph is a set of elements, which are assumed to be atomic, along with a set of edges. Each edge is an ordered pair of sets of elements, the first of which is called the invertex and the second of which is called the outvertex. Thus, metagraphs can be used to model:

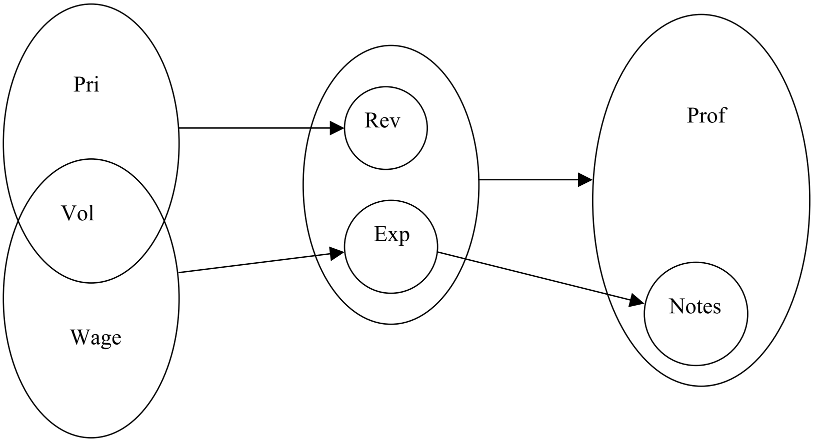

Figure 1.11. A metagraph.

- Data bases, in which invertices represent key attributes and outvertices represent content elements;
- Model bases, in which invertices represent model inputs and outvertices represent model outputs;
- Rule bases, in which invertices represent antecedent variables and outvertices represent consequent variables, and
- Workflow systems, in which invertices represent information flows entering a workstation and outvertices represent information flows emanating from a workstation.

Of these structures, the one closest to metagraphs is directed hypergraphs, in which the edges are also ordered pairs of sets of elements. The principal difference between metagraphs and directed hypergraphs is in the type of research done in these areas. Much of the work done on metagraphs is in decision support systems (DSS), and especially model-based DSS and in workflow management systems, although, as we will see, metagraphs are also relevant to other information structures, such as data management systems and rule-based systems.

We will examine two aspects of metagraphs, corresponding to the two parts of the book. The first is *Part I: Metagraph Theory*, which consists of five chapters, beginning with Chapter 2. The second is *Part II: Applications of Metagraphs*, which consists of four chapters. These are described below.

## 3. METAGRAPH THEORY (PART I)

Our purpose in Part I of this book is to present fundamental constructs (definitions, theorems, and interpretations of their significance) in a way that is independent of any problem context. This will provide a background for Part II in that it will provide the mathematical underpinnings essential to understanding the role that metagraphs play in analyzing the applications. But it will also provide an understanding of metagraphs that may be of assistance to anyone considering other application areas for which metagraphs may be useful.

- Chapter 2, **The Algebraic Structure of Metagraphs**, presents the use of matrices to describe metagraphs. An example is the adjacency matrix, a square matrix with one row and one column for each element in the generating set. Each member of the matrix is a set of triples, one for each edge connecting the row element to the column element. The triples define the invertex, outvertex, and the edge. We define addition and multiplication operators for the adjacency matrix, which allows us to define a transitive closure (i.e., a sum of powers) of the matrix. This will form the basis for a specification of the connectivity properties of metagraphs to be discussed in the next chapter.
- Chapter 3, **Connectivity Properties of Metagraphs**, examines the principal use of metagraphs discussed in this book. This is to determine whether there is a path connecting one set of elements to another set of elements. The definition of paths used in simple graphs and directed graphs, in which a path is a sequence of edges connecting a source element to a target element, does not apply here. Rather we define a metapath, which is a set (rather than a sequence) of edges connecting a set of source elements to a set of target elements, and this allows us to represent the parallelism found in more complex systems. In addition, we define metapath dominance, in which superfluous input elements and superfluous edges do not appear. We also investigate cycles, which are metapaths from a set of elements to itself.
- Chapter 4, **Metagraph Transformations**, examines the projection of a metagraph along a subset of its generating set. The elements in the projection consist only of those in the subset, and the edges in the projection correspond to metapaths in the original metagraph. Thus, a projection captures the connectivity relationships in a subset of a metagraph and thus represents a view of the metagraph taken by a person who is interested only in the elements contained in the projection set and the relationships between them. We also examine two related constructs, the inverse of a metagraph, in which edges become elements and elements become edges, and two related constructs – the pseudo-dual metagraph and the element-flow metagraph.
- Chapter 5, **Attributed Metagraphs**, presents an enhanced view of metagraphs in which additional variables, called attributes, are associated with the edges. One type of attribute is a resource, a qualitative or quantitative