An Information Structure Graph : A Structural Formalization of Information Semantics

Choon-Yeul Lee*

Abstract

Information semantics is a well-known issue in areas of information systems researches. It describes what data mean, how they are created, where they can be applied to ; thus, it provides indispensable information for management of data.

This article proposes to formalize information semantics by the processes that data are created or transformed. A scheme is proposed to describe an information production structure, which is called an information structure graph. An information structure graph is a directed graph, whose leaves are primary input data objects and whose root and internal nodes are output data objects.

Information semantics is derived from an information structure graph that has the data as its root. For this, rules are proposed to manipulate and compare graphs. The structural relationships among information structure graphs are mapped into semantic relationships among data.

* Management Information Systems Division, Kookmin University

1. Introduction

Information semantics has been recognized as a major issue in information management. Especially, with regard to information use, the primary concern has been misuse of data, which results from users ignorance of semantics [Brackett 1996] [Redman 1995][Liepens and Uppuluri 1990]. In contrast to its necessity, only a few methods have been proposed to describe information semantics, which are mostly descriptive [Hammer and McLeod 1977].

One way to describe information semantics is to formalize it from the perspective of information production. For example, if a quantity on hand is measured by an observation, we might say that it represents the quantity that is stored in a warehouse. However, if it is calculated from the previous month's quantity by deducting an outbound quantity and adding an inbound quantity, it represents the number that appears on an inventory book. Based on this idea, a scheme is proposed to materialize information semantics by information production structures.

Formalization of information semantics includes several tasks. The first is to define scopes of semantics. Information semantics is not a well-defined terminology. It might describe what data mean, how they are created, where they can be applied to, to name a few. The second is to develop a formal structure to describe semantics. Information semantics has been modeled as a part of data modeling. SDM ([Hammer and McLeod 1977]), the functional data model ([Shipman 1981]), and extended Entity Relationship models ([Teory et. al. 1986]) are typical ones. The third is to propose algorithms to compare information semantics. They help us to manipulate information semantics as an objective thing, not a subjective judgement.

In the next section, scopes of information semantics are described. In section 3, a scheme is proposed to formalize information semantics from the perspective of information production. It is formalized as a directed graph, which shall be called an information structure graph. In chapter 4, ideas are proposed to compare information semantics. At last, in chapter 5, implications of the research are summarized.

2. A Description of Information Semantics

2.1 Scopes

Information semantics includes almost everything we may think about data. For example, we may need to know encyclopedic definitions of data names, the situations that data are captured or calculated, persons who create data, to name a few. This characteristic of semantics has caused the ambiguity about its definition.

To formalize information semantics, we focus on processes that information is produced. That is, information semantics are indirectly formalized by the structure of information production processes.

2.2 A Conceptualization of Information Semantics : Information Structure Graph

A conceptualization of information production processes includes a conceptualization of proc-

esses as well as data. In this regard, it differs from both data models and process models. Data models such as an extended Entity-Relationship model ([Teory et. al. 1986]) and SDM ([Hammer and McLeod 1977]) focus on data. Process models are for system implementation and, thus, include all specifications for implementation ([Gane and Sarson 1979] [Warnier 1981] [Yourdon 1990], to name a few); however, a conceptualization of information production is for understanding information semantics from the perspective of information production. Thus, it includes structures of processes as well as data.

An attempt to model information production processes is an information manufacturing system [Wang 1998]. It decomposes information production into data units, vendors, data quality blocks, processing blocks and consumers. A data unit supplied by a vendor passes through data quality blocks and processing blocks, and delivered to consumers.

Another one is FIP (Functions of Information Processing) model ([Redman 1996]). It models data processing as processes of producing OIP (output information product) from IIP (input information product) by applying FIP. That is, it models the process of creating data_set_C from data_set_A and data_set_B as follows ;

Data_set_A FIP Data_set_B = Data_set_C

Here, data_set_A and data_set_B are IIPs, and data_set_C is an OIP. FIP is an classification of information production processes into associate, filter, prompt, queue, regulate and transmit.

For conceptualization of information production processes, this research proposes an information structure graph, which is based on a conceptual graph [Sowa 1984]. A conceptual graph has been applied to systems requirements formalization, especially for hardware systems [Cyre 1997]. An information structure graph applies it to conceptualization of information production processes.

An information structure graph is a directed graph that connects data objects. A parent node is an output data object (output information products) and children nodes are input data objects (input information products).[1)]

In general, information production means two different things. One is to create new data objects ; the other is to update values of data objects. Based on this distinction between data objects and values, we categorize information production processes into three classes as shown in <Table 1>.[2)]

(1) The first one is to produce a new data object, which shall be called an object creation. A typical example might be to decide an order quantity based on a quantity on hand and a safety stock. However, some data objects are captured without any reference to other data objects. They

1) In this study, data objects or information products are used interchangeably. They denote data items, fields, records, reports or documents. We may call data objects or information products as data or information, for the sake of simplicity.

2) In information production, data values are created or new data objects are named. If neither occurs, it means that nothing has been created. In this sense, the fourth one is not included in the categories of information production.

are called primitives or primitive data objects.

(2) The second one is to update a current value of a data object. It shall be called a value update. A typical example is an update of a quantity on hand when goods are deposited or shipped.

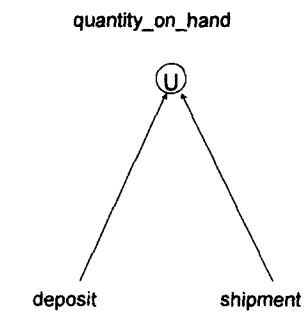
(3) The third one is to produce a new data object without creating new data values. It shall be called a structure transformation. Typical examples are composing documents from data items and classifying data objects into specialized or generalized ones.

〈Table1〉 Categories of Information Production

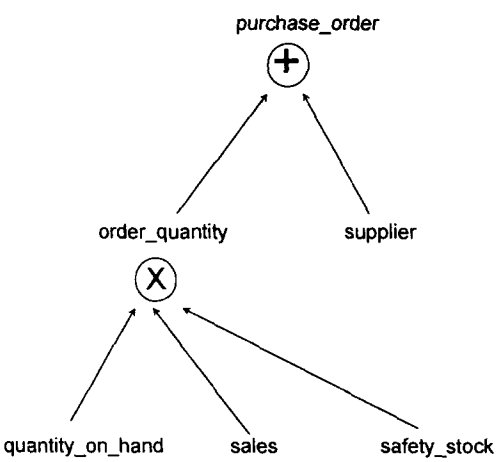
		Data Values	
		created	not created
Data Objects	Created	Object creation	Structure transformation
	not created	Value update	-

For example, let us assume that quantity on hand is updated from goods deposit and shipment. Then it is conceptualized as shown in (Figure 1). In the figure, deposit and shipment are input data objects. And, quantity on hand is an output data object. A symbol assigned to a quantity on hand, ⓤ, represents that it is a value update.

Further, let us assume that an order quantity is calculated by comparing quantity on hands and sales against a safety stock. And, a purchase order is issued to a supplier. Then the processes are conceptualized as shown in (Figure 2). It depicts that an order quantity is produced from quantity on hand, sales and a safety stock. A symbol assigned to order quantity, ⓧ, represents that it is an object creation. The order quantity is combined with supplier to form a purchase order. A symbol assigned to a purchase order, ⊕, represents that it is a structure transformation.



(Figure 1) An Information Structure Graph for Quantity On Hand



(Figure 2) An Information Structure Graph for Purchase Order

As shown in the figures, an information structure graph shows input data objects used to produce an output data object and categories

of information production. Detail processes or functions are not included in the graph.

3. A Formal Description of An Information Structure Graph

An information structure graph is a directed graph composed of nodes and arches.

Definition 1 *: An information structure graph is a graph* $G = \langle V, E, \mu \rangle$ *such that,*

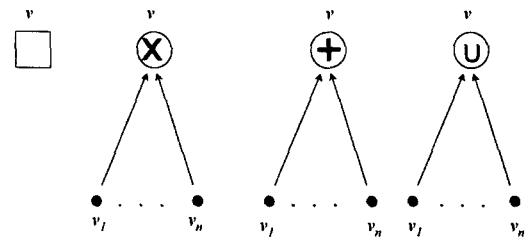
(1) V *are nodes that represent data entities, and* E *are arches that connect nodes. I.e.,* $\langle V, E \rangle$ *is a directed graph.*

(2) μ *is a function from* V *to the set of information production types {□, ⊗, ⊕, ⓤ} that satisfy the following condition.*

(a) $\mu(\nu) = \square$, *iff* ν *is a leaf.*

(b) If $\mu(\nu) = \otimes, \oplus, ⓤ$, *then the children of are distinct nodes.*

$\mu(\nu)$ is called an information production type of a data object ν. As shown in categories of information production, information production is classified into an object creation, a structure transformation, and a value update, which are represented by ⊗, ⊕, ⓤ, respectively. That is, $\mu(\nu) = \otimes$ means that an information production type of a data object ν is an object creation ; $\mu(\nu) = \oplus$ means that an information production type of a data object ν is a structure transformation ; and, $\mu(\nu) = ⓤ$ means that an information production type of a data object ν is a value update ; in addition, $\mu(\nu) = \square$ means that a data object ν is a primitive one. Thus, for the sake of simplicity, we include input data objects into information production types as shown in the followings (see (Figure 3)) :



(Figure 3) Nodes in an Information Structure Graph

(1) $\mu(\nu) = \square$ denotes that "ν is a primitive data objects and there exists no edge with head ν."

(2) $\mu(\nu) = (\otimes, \nu_1, \cdots, \nu_n)$ denotes that "$\mu(\nu) = \otimes$ and ν has n children $\nu_1, \cdots, \nu_n$. And, there exist exactly n edges with head ν and their tails are $\nu_1, \cdots, \nu_n$."

(3) $\mu(\nu) = (\oplus, \nu_1, \cdots, \nu_n)$ denotes that "$\mu(\nu) = \oplus$ and ν has n children $\nu_1, \cdots, \nu_n$. And, there exist exactly n edges with head ν and their tails are $\nu_1, \cdots, \nu_n$."

(4) $\mu(\nu) = (ⓤ, \nu_1, \cdots, \nu_n)$ denotes that "$\mu(\nu) = ⓤ$ and ν has n children $\nu_1, \cdots, \nu_n$. And, there exist exactly n edges with head ν and their tails are $\nu_1, \cdots, \nu_n$."

Root(G) is the root of an information structure graph G ; and, leaf(G) is a set of leaves of the graph G. For an data object ν, G() is its information structure graph. Thus, the root of an information structure graph $G(\nu)$ is the data object ν. I.e.,

$$\nu = \text{root}(G(\nu))$$

In addition, other managerial information might be added to information structure graphs. They might be times (denoted as $\tau(\nu)$) that information production processes are executed, implementation types of processes (denoted as $\rho(\nu)$), to name a few.

$$\tau(\nu) = (t = t0) \mid (t = t + t) \mid cond$$

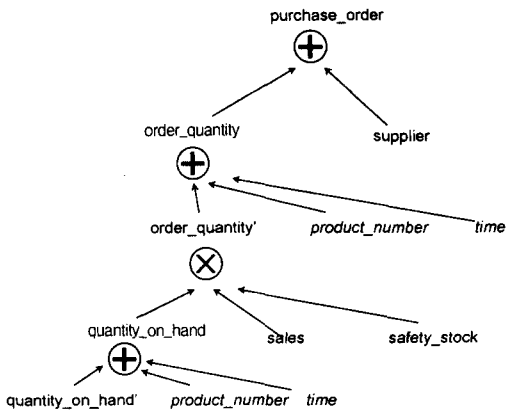
cond = $\nu\ \theta$ a | $\nu\ \theta\ \nu'$ | cond AND cond | cond OR cond | NOT cond

where, θ is a relational operator (=, =/=, >, >=, <, <=),
a is a constant, and
ν and ν' is a data object,

$$\rho(\nu) = \text{manual} \mid \text{program} \mid \text{system}$$

In information structure graphs, data objects are relations that are not necessarily first normal forms. For example, in (Figure 2), purchase_order is represented as a nested relation that contains order_quantity, supplier, where order_quantity might be composed of product_identifier, quantity_on_hand, and time. Supplier is also composed of primitive data objects of supplier_identifier, name, address, and phones.

In information structure graphs, a structure transformation depicts a composition of an output data object. Purchase_order is composed of order_quantity and supplier. Though it is not shown in (Figure 2), order_quantity might be composed of product_identifier, time, and order_quantity. Quantity_on_hand might be composed of product_identifier, time, and quantity_on_hand. In other words, every data object needs to include an identifier in addition to values. In this sense, (Figure 2) needs to be expanded to (Figure 4) to include this kind of structure transformations ; however, structure transformations for quantity_on_hand and order_quantity are obvious even though they are not decomposed. Thus, for the sake of simplicity, these structure transformations are not included in information structure graphs as shown in (Figure 2). In sum, product_identifier, time and quantity_on_hand are treated as a single data object, which is quantity_on_hand.



(Figure 4) An Information Structure Graph for Purchase Order (revised)

Information structure graphs may be formulated into different ways even though they represent equivalent information production processes. Here, we introduce rules to test an equivalence of information structure graphs.

Rules : *An information structure graph* $G = \langle V, E, \mu \rangle$ *is reducible with an order of* $\langle \textcircled{u}, \otimes \rangle \oplus$. *That is, the following rules are satisfied ;*

(1) If $\mu(\nu) = (\oplus, \nu_1, \nu_2)$ *and* $\mu(\nu_1) = (\oplus, \nu_3, \nu_4)$, *then* $\mu(\nu) = (\oplus, \nu_3, \nu_4, \nu_2)$.

(2) *If* $\mu(v) = (\otimes, v_1, v_2)$ *and* $\mu(v_1) = (\oplus, v_3, v_4)$, *then* $\mu(v) = (\otimes, v_3, v_4, v_2)$.

(3) *If* $\mu(v) = (\oplus, v_1, v_2)$ *and* $\mu(v_1) = (\otimes, v_3, v_4)$, *then* $\mu(v) = (\otimes, v_3, v_4, v_2)$.

(4) *If* $\mu(v) = (\cup, v_1, v_2)$ *and* $\mu(v_1) = (\oplus, v_3, v_4)$, *then* $\mu(v) = (\cup, v_3, v_4, v_2)$.

(5) *If* $\mu(v) = (\oplus, v_1, v_2)$ *and* $\mu(v_1) = (\cup, v_3, v_4)$, *then* $\mu(v) = (\cup, v_3, v_4, v_2)$.

(6) *If* $\mu(v) = (\otimes, v_1, v_2)$ *and* $\mu(v_1) = (\otimes, v_3, v_4)$, *then* $\mu(v) = (\otimes, v_3, v_4, v_2)$.

(7) *If* $\mu(v) = (\cup, v_1, v_2)$ *and* $\mu(v_1) = (\cup, v_3, v_4)$, *then* $\mu(v) = (\cup, v_3, v_4, v_2)$.

(8) *If* $\mu(v) = (\cup, v_1, v_2)$ *and* $\mu(v_1) = (\otimes, v_3, v_4)$, *then* $\mu(v) = (\cup, v_3, v_4, v_2)$.

(9) *If* $\mu(v) = (\otimes, v_1, v_2)$ *and* $\mu(v_1) = (\cup, v_3, v_4)$, *then* $\mu(v) = (\otimes, v_3, v_4, v_2)$.

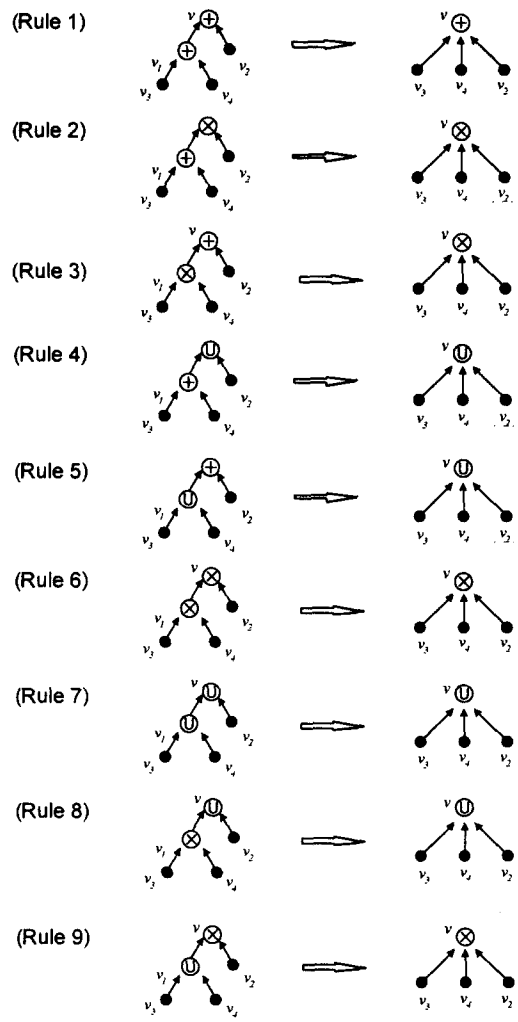
These rules are shown in (Figure 5). In the figure, information structure graphs in the left side are said to be reducible to the ones in the right side. And, graphs in the right side are said to be expandable to the ones in the left side.

For example, the information structure graph in (Figure 2) can be reduced to the following, by the Rule 3. The reduced graph shows that a purchase order is created from quantity_on_hand, sales, safety_stock and supplier.

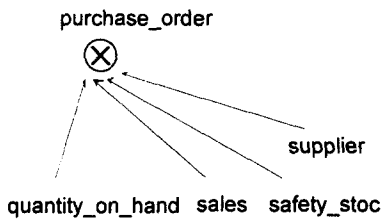
An information structure graph is said augmented if nodes and/or edges are added without altering its structure.

Definition 2. *An information structure graph* $G' = \langle V', E', \mu \rangle$ *is an augmentation of a graph* $G = \langle V, E, \mu \rangle$ *if the following conditions are satisfied ;*

(1) $V \subseteq V'$



(Figure 5) Information Structure Graph Rules



(Figure 6) An Information Structure Graph for Purchase Order (reduced)

(2) $E \subseteq E'$

(3) If $(v_1, v_2) \in E' - E$, *then* $v_2 \in V'$.

I.e., all new edges are between new nodes, or from nodes in V to a new node.

An augmentation adds new nodes to a graph without altering its structure. For example, in (Figure 7), $graph_2$ and $graph_3$ are augmentations of $graph_1$. $Graph_2$ adds a root (or an intermediate node) to $graph_1$. $Graph_3$ adds a leaf as well as a root (or an intermediate node) to $graph_1$. However, $graph_4$ is not an augmentation of $graph_1$ because the structure of $graph_1$ has been changed in $graph_4$.

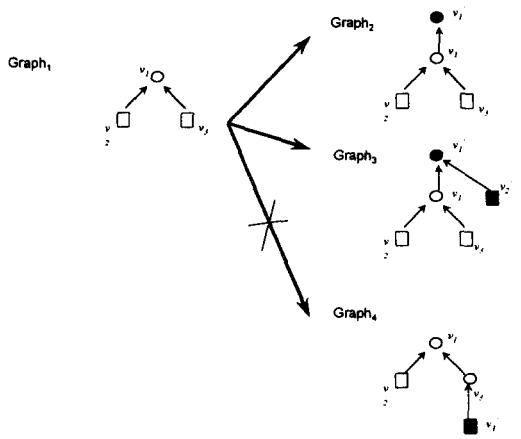
Information structure graphs are augmented repeatedly. I.e., if G_2 is an augmentation of G_1 and G_3 is an augmentation of G_2, then G_3 is an augmentation of G_1. Among augmented graphs, some have identical leaves. They are called homogeneously augmented graphs.

Definition 3. *Information structure graphs* G_1 *is called a homogeneous augmentation of* G_2 *if* G_1 *is an augmentation of* G_2 *(or vice versa), and* $leaf(G_1) = leaf(G_2)$. *A homogeneous augmentation set is a set of information structure graphs that are homogeneous to each other.*

Homogeneously augmented graphs do not add leaves to a graph. They adds intermediate nodes or a root to a graph, which means that additional objects are produced from the same primitive data objects.

4. An Analysis of Information Semantics

An information structure graph represents the structure of information production processes. Based on this characteristic of an information structure graph, we define semantic relationships among information products. If an information structure graph is a homogeneous augmentation of another one, it means that both information products are derived from the same primitive data objects.



(Figure 7) Augmentations of Information Structure Graphs

Definition 4. *Semantic derivation : Let* $G = \langle V, E, \mu \rangle$, $G' = \langle V', E', \mu' \rangle$ *be information structure graphs of data object d, d', respectively. Then data object d' is semantically derived from data object d if G' can be homogeneously augmented from G. That is,*

(1) there exist an information structure graph G'' such that G' is reducible to G'' (or G'' is expandable to G')

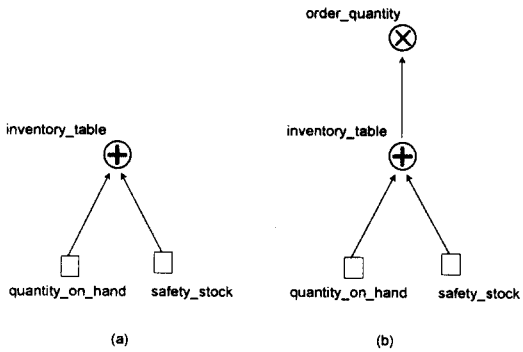
(2) $G'', G \in \underline{G}$

where, $\underline{G}$ *is a homogeneous augmentation set.*

A semantically derived data object *d'* has the

same primitive data objects as a data object *d* and includes additional data objects which are produced from those input data objects. It means that a semantically derived data object is produced from *d* without using any additional input data.

For example, let us assume that order quantity is determined based on inventory table. Then, as shown in (Figure 8), an information structure graph of order quantity is a homogeneous augmentation of inventory table's. And, an order quantity is said to be semantically derived from an inventory table.



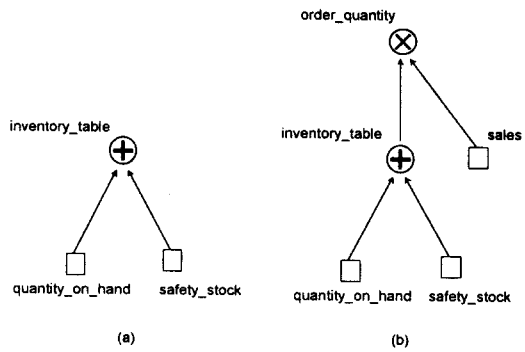
(Figure 8) Semantically Derived Data Objects

If an information structure graph of order quantity is depicted as shown in (Figure 9), order quantity is not semantically derived from inventory table. It is derived from sales as well as inventory table. We call it a semantically augmented data object.

Definition 5. *Semantic augmentation : Let G = <V, E, μ>, G' = <V', E', μ'> be information structure graphs of data object d, d', respectively. Then data object d' is a semantic augmentation of a data object d if G' can be augmented from G. That is,*

(1) there exist an information structure graph G'' such that G' is reducible to G'' (or G'' is expandable to G')
(2) G'' is an augmentation of G

A semantically augmented data object utilizes extra information (i. e., primitive data objects) in addition to the original one. As shown in (Figure 9), an order quantity is created using additional information of sales as well as an inventory table. From definitions, semantic derivation is a special case of semantic augmentation. That is, semantically derived data objects are semantically augmented ones without additional primitives.



(Figure 9) Semantically Augmented Data Objects

In sum, information structure graphs help to test semantic relationships among data objects. If an information structure graph is an augmentation of another, one is semantically related to the other. If not, one is not semantically related to the other. If an augmentation is homogeneous (i. e., an augmentation does not includes additional primitives), one is derived from the other.

If not, one utilizes additional input data and, thus cannot be derived from the other.

5. Conclusion

This study has proposed a scheme to formalize information semantics. For this, information production processes are conceptualized into information structure graphs. An information structure graph helps

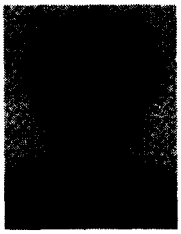
(1) to represent information production processes. This kind of information can be utilized for management of data.
(2) to define and manipulate information semantics. By comparing structures of information structure graphs, information semantics are compared as objective things, not as subjective judgements.

This paper is an initial attempt to draw structures about information semantics. Its utilization has to be tested by further researches.

References

[1] Brackett, M.H., *The Data Warehouse Challenge : Taming Data Chaos*, John Wiley & Sons, Inc., 1996.
[2] Cyre, W.R., "Capture, Integration, and Analysis of Digital System Requirements with Conceptual Graphs," *IEEE Transactions on Knowledge and Data Engineering*, Vol.9, No.1, 1997, pp.8-23.
[3] Gane, C., and Sarson, T., Structured Systems Analysis : Tools and Techniques, Prentice-Hall, 1979.
[4] Hammer, M. and McLeod, D., "Database Description with SDM : A Semantic Database Model," *ACM Transactions on Database Systems*, Vol.6, No.3, 1981, pp.351-386
[5] Lee, C. "A Data Manufacturing Model," *'96 Database Symposium*, November 1996, pp. 311-326. (in Korean)
[6] Liepens, G.E. and Uppuluri, V.R.R., *Data Quality Control : Theory and Pragmatics*, Marcel Dekker, 1990.
[7] Redman, T.C., "Improve Data Quality for Competitive Advantage," *Sloan Management Review*, 1995 winter, pp.99-107.
[8] Redman, T.C., Data Quality for the Information Age, Artech House, 1996.
[9] Shipman D.W., "The Functional Data Model and the Data Language DAPLEX," *ACM Transactions on Database Systems*, Vol.6, No.1, 1981, pp.141-173
[10] Sowa, J. F., *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, 1984.
[11] Teory, T.J., Wang, D., & Fry, J.P., "A Logical Design Methodology for Relational Databases Using the Augmented Entity-Relationship Model." *Computing Survey*, Vol.18, No.2, 1986, pp.197-221.
[12] Wang, R.Y., Storey, V.C. and Firth, C.P., "A Framework for Analysis of Data Quality Research," *IEEE Transactions on Knowledge and Data Engineering*, Vol.7, No.4, 1995, pp.623-639.
[13] Wang, R.Y., "A Product Perspective on Total Data Quality Management," *Communications of ACM*, Vol.41, No.2, 1998, pp.58-65.
[14] Warnier, J.D., *Logical Construction of Systems*, Van Nostrand Reinhold, 1981.
[15] Yourdon, E.N., *Modern Structured Analysis*, Prentice-Hall, 1990.

■ 저자소개



이 춘 열

저자는 서울대학교에서 수학하였으며, 미국 미시간 대학교에서 경영정보학(Computer and Information Systems) 박사학위를 취득하였다. 이후 한국통신 소프트웨어 연구소에 근무하였으며, 현재 국민대학교 정보관리학부에 부교수로 재직하고 있다. 주요 관심분야는 데이터베이스 및 이의 응용, 자료 검색, 정보 공학 등이며, 현재 데이터웨어하우징 도구 및 데이터 관리 기법 등에 대한 연구를 수행중이다.