The Journal of Korean
Institute of CALS/EC
Vol 7, No. 1, April 2002

ERP 기반의 비즈니스 프로세스 재설계 방법

정동길*

A Business Process Redesign Method within an ERP Framework

Dong-Gill Jung

Abstract

The behavioral and dynamic implications of an ERP implementation/installation are, to say the least, not well understood. Getting the switches set to enable the ERP software to go live is becoming straightforward. The really difficult part is understanding all of the dynamic interactions that accrue as a consequence. Dynamic causal and connectionist models are employed to facilitate an understanding of the dynamics and to enable control of the information-enhanced processes to take place. The connectionist model can be analyzing (behind the scenes) the information accesses and transfers and coming to some conclusions about strong linkages that are getting established and what the behavioral implications of those new linkages and information accesses are. Ultimately, the connectionist model will come to an understanding of the dynamic, behavioral implications of the larger ERP implementation/installation per se. The underlying connectionist model will determine information transfers and workflow. Once a map of these two infrastructures is determined by the model, it becomes a relatively easy job for an analyst to suggest improvements in both.

Connectionist models start with analog object structures and then use learning to produce mechanisms for managerial problem diagnoses. These mechanisms are neural models with multiple-layer structures that support continuous input/output. Based on earlier work performed and published by the author[10][11], a Connectionist ReasOning and LEarning System(CROLES) is developed that mimics the real-world reasoning infrastructure. Coupled with an explanation subsystem, this system can provide explanations as to why a particular reasoning structure behaved the way it did. Such a system operates in the background, observing what is happening as every information access, every information response coming from each and every intelligent node (whether natural or artificial) operating within the ERP infrastructure is recorded and encoded. The CROLES is also able to transfer all workflows and map these onto the decision-making nodes of the organization.

Key Word: ERP, BPR, Connectionist Approach

* 명지대학교 경영정보학부 교수

1. INTRODUCTION

Today business enterprises over the world interface more and more competitive environments to survive and succeed in more and more globalizing economy. In such environments the primary goals of the business enterprises are to adapt themselves continuously to changing environments and sustain competitive advantages by improving the oragnizational performance. Hammer advocates that the power of modern information and communication technology (ICT) should be used to radically redesign business processes to achieve dramatic improvements in their performance [7][8].

For the business enterprises to achive such goals, a new approach to co-ordinating processes across organisations is essential. The aim should be to maximise the performance of interrelated activities rather than individual business functions. A new class of packaged application software has emerged over the past decade. Usually called enterprise resource planning (ERP) systems, these comprehensive, packaged software solutions seek to integrate the complete range of an enterprise's business processes and functions in order to present a holistic view of the business from a single information and IT architecture [13].

ERP system automates core business activities such as manufacturing, human resource, finance and supply chain management. The benefit of such system is that it can speed decision making, reduce costs and give managers control over the whole business. Implementing a fully integrated ERP system requires the organisation to be process oriented and for all parts of an organisation to adhere to the same precise processes. This forces the organisation to undergo organization - wide business process reengineering (BPR), with ERP software merely being an enabler of the change[3][4].

However, using ERP system for organization-wide BPR usually have turned in very complex tasks and unsatisfactory results. The behavioral and dynamic implications of an ERP implementation/installation are, to say the least, not well understood. Getting the switches set to enable the ERP software to go live is becoming straightforward. The really difficult part is understanding all of the dynamic interactions that accrue as a consequence [1][4].

In this article, the purpose is to present an efficient and robust mechanism that can capture the patterns of information flow between business processes and to extract behavioral implications from them within an ERP framework. For the purpose of this research, the author synthesized existing research and practice results from ERP enterprise and business process modeling and the author's earlier research of CIROS (Connectionist Inexact

ReasOning System) for inexact reasoning [10][11].

This article consists of the following contents: in the second section following the introduction, the author review existing research results from enterprise and business process modeling for ERP and enterprise systems. In doing so, the author focuses on the problems and misuses and the possible causes for those problems and misuses in existing ERP systems. Also, the author review existing and industry-wide-adopted business process models.

In the third section, the author presents a simplified business process model for our research purpose by modifying existing ones. The result is a Simplified Business Process Model(SBPM). Although it has a simple network structure such that it consists of only two components, process nodes and links between nodes, the author believes it serves well for our research purpose and has enough extendability for practical applications.

In the fourth section, the author modifies the CIROS model to fit into our current research concern. Then the author synthesizes the SBPM and the CIROS, and the result is a new connectionist model, called CROLES (Connectionist ReasOning and Learning System). It has the capability of inferencing and learning for BPR within an ERP framework.

In the fifth section, the author suggests some further possible analyses that can be done based on the outputs of the CROLES. In the sixth section, the author summarizes the research concept and present some further research issues extending the research result here.

2. ENTERPRISE REENGINEERING AND BUSINESS PROCESS MODELING

2.1 Business Process Models for BPR within an ERP framework

The languages and methodologies for describing business process modeling for BPR are very diverse. The types of models required to design, analyze and operate business enterprises have been discussed by numerous researchers [9][12]. The general conclusion is simply that different models are required for different purposes. Thus a critical research and engineering issue is to identify which type of model should be used for what purpose.

As stated above, business processes can be described or modeled in various different ways for various purposes. However, our primary concern in this research is the patterns of information flow between business processes and to find out some behavioral implications from them within an ERP framework. For that purpose, the author is going to review a real business process model from an ERP software company and then, the author builds and presents a business process model for the research purpose by simplifying it, which is the task described in Section 3. In the following, the author briefly reviews and describes the business process model from SAP/R3 Business Blueprint [4].

2.2 SAP/R3 Business Process Model

A tool for business process engineering (and reengineering) in the SAP R/3 system is Business Blueprint. It includes various components for BPR. The language and methodology of Business Blueprint is based on a concept called EPC (Event-driven Process Chaining).

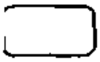
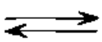
Designation	Icon	Definition	Example
Event		Events describe the occurrence of a status that in turn acts as a trigger	Order is received
Function		Functions describe transformations from an initial status to a final status	Verify order
Organization unit		Organization units describe the outline structure of an enterprise. The organization units in the R/3 system are system organization units.	Sales organization
Information, material or resource object		Information material, or resource objects portray objects in the real world (e.g., business objects, entities).	Sales order Inspection result
Process path		Process show the connection from or to processes (navigation aid).	Delivery processing
Logical operator		Logical operators describe the logical relationships between events and functions of processes	"XOR", "AND", "OR"
Control flow		Control flows describe the chronological and logical interdependencies of events and functions or processes.	
Information/ material flow		Information/material flows define whether a function is read, changed, or written.	
Resource/organization unit assignment		Resource/organization unit assignments describe which unit (employee) or resource processes a function or process.	

Figure 1. Process Description Language Elements of SAP R/3 EPC

EPC is based on four key elements: events (when should something be done?); tasks or functions (what should be done?); organization (who should do what?); and communication (what information is required to do the right task?). Events are the driving force behind a business process, prompting one or more activities to take place. The EPC provides the interconnections between tasks, data, and organizational units and the logical time sequence involved to define a business process. Figure 1 illustrates the process description language elements in EPC.

An event always triggers a task. It is important for each task to beinwith at least one event (the start event) and to end with at least one final event (the finish event). The organizational units (departments, people, etc.) responsible for doing the task are added to the business process to show a complete picture of how the business process is structured and performed.

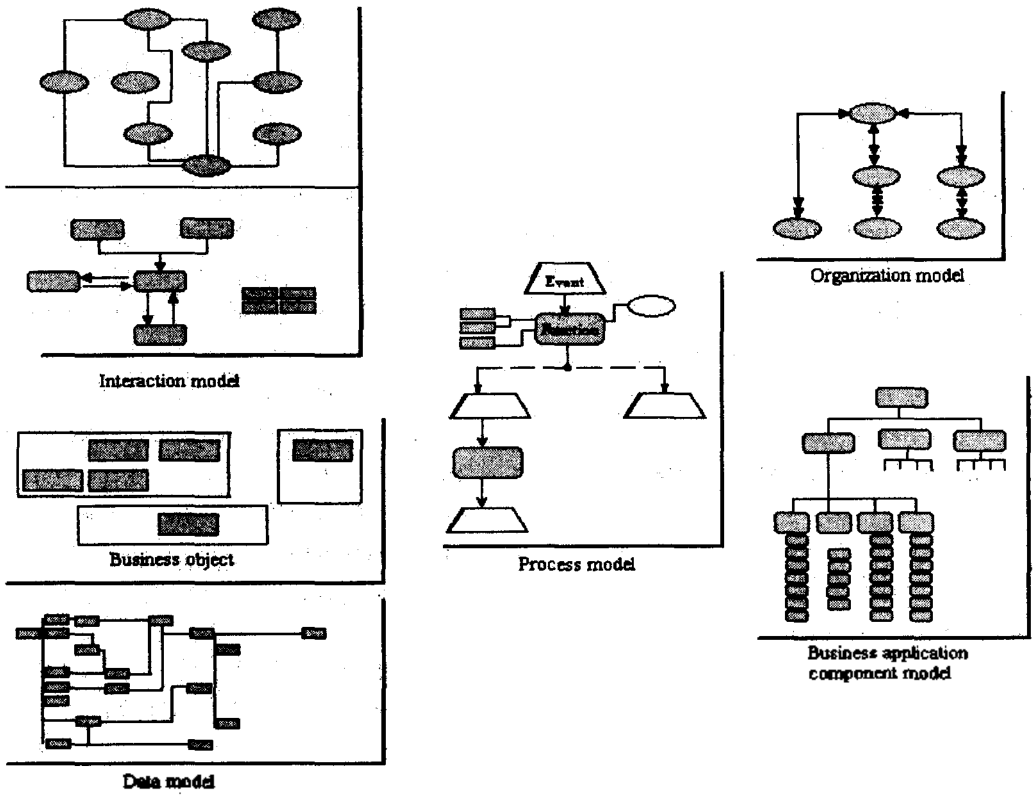


Figure 2. The Process Model

When companies are faced with the complexities of planning a business information concept, however, different issues require different analyses of aspects of the business. To fill this need, the Business Blueprint of SAP R/3 system provides the users with the capability of modeling from various points of view such as business functions and application components, business organization, data, business process, and interaction of the various model components. Each view answers a distinct question about the interaction of business processes.

Accordingly, the users of the Business Blueprint can produce different models from each view, i.e., application component model, organization model, data model, process model, and interaction model. Among these the process view is the central view, and hence the process model incorporates the other models to give the users in an integrated view of the whole business process under consideration, as illustrated in Figure 2.

2.3 Problems with existing ERP systems for BPR

ERP software projects often deliver disappointing organizational results, even if the information systems work well technologically. One explanation is that companies use new technologies to automate old, ineffective processes. Moreover, BPR has had a high failure rate with consultants estimating that as many as 70% of BPR projects fail [8].

ERP application systems must be developed for the organization as a whole, rather than for optimizing subordinate functions in isolation. If this principle is not followed, then a small part of an organization may not be operating for the benefit of the organization as a whole [18][19].

For example, the interaction model of SAP/R3 Business Blueprint allows companies to analyze how information flows between general applications. Interaction models reveal information flows from senders to receivers, and vice versa. However, the interaction viewpoint describes these interactions only at the partial areas of applications and functions without examining the partial processes from the viewpoint of their relationships to other processes and the whole business process.

In this regard, a new approach to coordinating processes across organizations is essential in order to achieve a sustainable competitive advantage. Existing ERP systems for BPR emphasized analytical approaches to business processes but typically started with existing business processes [20].

3. BUSINESS PROCESS MODEL FOR THE RESEARCH

In this section, the author defines a business process, to identify some important components of a business process, and then present the Simplified Business Process Model for our research purpose which is based on the definition and characteristics of the business process identified and the SAP's Business Process Reference Model.

3.1 Definition: Business Process

A business process is a specified activity in an enterprise that is executed repeatedly. According to J. Martin, business processes have the following characteristics [15].

- Business processes have definable beginning and end points. These points are actually events in Object-Oriented Modeling terms.
- Business processes have inputs consisting of information, material, energy, etc., which they transform into outputs that also consist of information, material, energy, etc.
- Business processes are created by higher level business processes that monitor and control their operation (i.e., Business processes are structured in hierarchical organizations).
- Business processes consume resources that are allocated to them by their higher level controlling business processes.
- Business processes report their status to their higher level controlling business processes.

Accordingly, a business process model should incorporate a hierarchical structure and include process components and the relationship between the processes as a minimum. In addition, events, resources such as information, actor or owner, and information sinks can be included as the modeler requires [5].

3.2 Process Dependency and Business Rules

A given business process depends on other business processes. The process dependency model shows how business processes relate to each other and how they are dependent on each other. A given process is dependent on another process in the sense that the process in question cannot take place until the other process has completed for whatever reason. Basically, process dependency is determined by the business rules of the enterprise. Because business rules are very important in building the business process model in this research, a thorough explanation will be given next. For the description of the process dependency, the author uses an example of business process diagram shown in Figure 3.

There can be several forms of process dependency:

- One-to-one: Exactly one process precedes another in sequence and shows straight-line connectivity. In Figure 3, Process 2 and Process 3 has this form.
- Many-into-one: Two or more processes precede the invocation of one process. In Figure 3, Process 2 and Process 4 (together) invocate Process 5.
- One-into-many: One process precedes the invocation of two or more processes. In Figure 3, Process 2 precedes to invocate Process 3 and Process 5.

The enterprise has a set of business rules representing the conditions, constraints, and policies that control its organization, direction, and operation. Business rules are, in a sense, a shorthand language for expressing the business knowledge. Carried to its logical extreme, the set of business rules of an enterprise act as the declarative script of the enterprise. No matter what happens, one or more business rules would control what happens after that [5].

The business rules

The enterprise is an example of a general system. Any general system can be analyzed in terms of three perspectives: a structural perspective, a functional perspective, and a behavioral (or dynamical) perspective. Business rules apply to three perspectives of the enterprise. Thus a given business rule may roughly be classified in one of the three perspectives.

Structural business rules are related to the structure of the enterprise. The structure of the enterprise is its set of entities and their relationships at a specific point of time. Structural business rules maintain the integrity of enterprise entities. Examples of structural business rules are as follows:

- The enterprise should have a marketing department, personnel department, finance department, accounting department, and customer service department. (Organization rule)
- Annual Total Profit = Annual Total Revenue – Annual Total Expense (Entity definition)

Functional business rules are the business rules that specify the goals and objectives of the enterprise. Basically they collectively define the “what should be done (by whom)”. Examples of these rules are:

- The enterprise should keep up to 35% of the domestic market of the product A.
- The management of human resources is the responsibility of the managers throughout the company (as

opposed to being established as separate organizational unit)

Behavioral business rules are used to control the preconditions and post conditions of the state changes of the enterprise. A typical behavioral business rule has the following form:

When certain events occur and certain conditions hold true, then the system states change

As can be seen in the form, behavioral business rules are closely related to the business process chains. In fact, the relationships between different business processes and chaining of the business processes are solely determined by the behavioral business rules of the enterprise in our Simplified Business Process Model (see the following subsection).

3.3 A Simplified Business Process Model (SBPM)

A business process model that is simplified for our purpose can be constructed as follows:

The whole business process under consideration consists of smaller (sub) process nodes and relationships between them. The relationship type between two process nodes that is captured in this model is the triggering control sequence. That is, there should be an event or a set of events between the two process nodes that satisfy some predefined (according to some business rules) pre- and post-conditions for the relationship between the two nodes.

Moreover, because the relationship type is a kind of triggering control sequence, it should have some temporal sequence (i.e., the process node before the occurrence of the event comes first and then the other process node after the occurrence comes) and therefore can be represented as a directed link. If we represent the process node as a circle and the relationship between two nodes as a directed link, a typical business process can be diagramed as in Figure 3.

Each process (node) must have as its components a task function which is basically a set of services that are expected to be done at this node, an actor who is in charge of or responsible for the task of this process node (an actor may be a person, an organizational unit, or an artificial (software) agent), and some kind of information storage(s) to read input information from or to store output information in. Here, the output information simply means the information resulting from the execution of the node's task (or function). The information storage(s) and the information stored in there may be owned (or used) by the process node only, but usually shared by other process nodes.

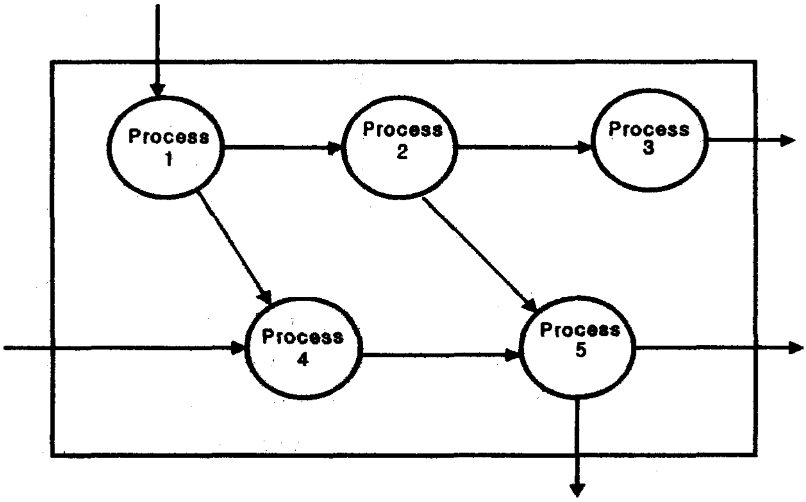


Figure 3. Simplified Business Process Diagram

The input information may be just simply an input from another node (or from outside of the whole business process in consideration) or read-in records or documents from the permanent storage(s) used by the node. The output information may be some temporary results to be delivered to the nodes that are linked to this node, or some permanent results that should be recorded and kept in a database as a record of a table or a document. Therefore, output information of a process node can be either temporary or permanent. The structure of our Simplified Business Process Model is diagramed in Figure 3.

For the research purpose, each business process (node) will have a very simplified architecture as shown in Figure 4.

Now we consider the network structure (topology) of the whole business process. Usually, this network of the process nodes will have a mesh topology and might contain loops or cycles [2]. The author uses the term "loop" to mean the link starts from a node and comes back directly to the same node. On the other hand, a cycle is a path (a sequence of links) on the network that starts from a node and ends with the same node but with at least one visit to another node.

As we have seen in the Business Process Model Section (BPM Section) of this article, the original business process diagram under consideration, if represented

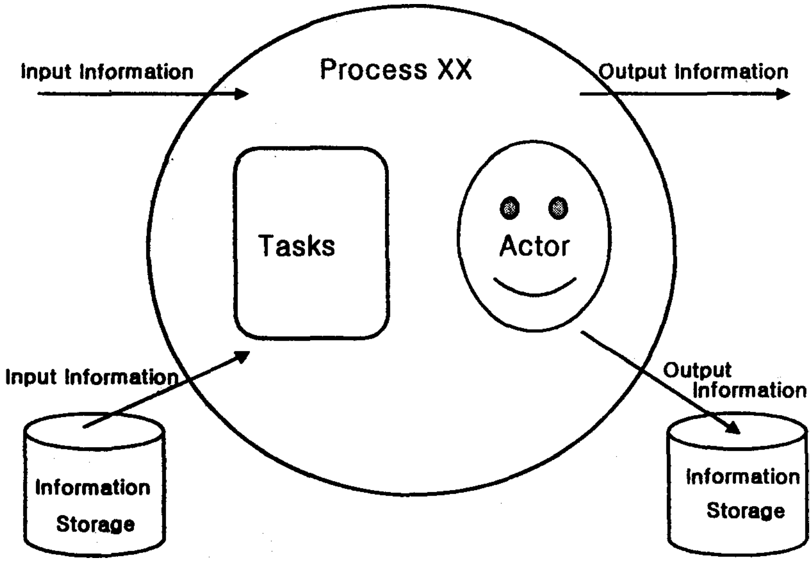


Figure 4. Process Node Architecture

in a network, might involve loops or cycles, and thus violate the assumptions made in CROLES. As we have seen in the BPM Section, however it seems that in real fields there are very rare cases where the business process diagrams include loops or cycles. So, it will be safely assumed that our business process model does not involve any loop or cycle

4. CONNECTIONIST APPROACH TO BUSINESS PROBLEM-SOLVING AND BPR

4.1 Design Principles for CROLES

As mentioned earlier in this article, design of CROLES is based on the author's earlier works on the design of inexact reasoning systems [10][11]. To fully understand CIROS and related works on connectionist's approaches to business problem solving, interested readers may reference those two papers.

The fundamental structure of CROLES will look like the Figure 5. It is basically a layered, acyclic network. At the bottom is the input layer that represents the collection of source or input process nodes. The top layer is the output layer that represents the collection of target or output process nodes. Between the two layers are one or more intermediate layers that represent various classes of intervening process nodes.

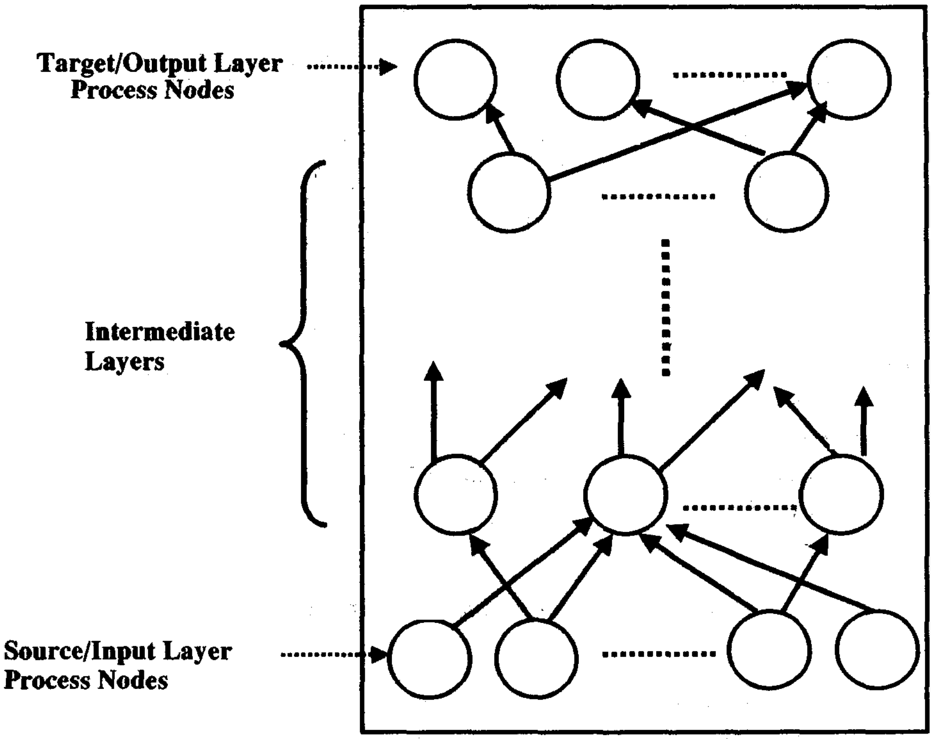


Figure 5. Fundamental Structure of CROLES

To capture and represent knowledge about the behavior of the ERP system, the following assumptions are made for CROLES:

1. The domain knowledge (K) consists of two components, the set (V) of business process nodes and the set (E) of relationships (links) among the business process nodes. Mathematically, K is a diad of V and E (K = <V, E>).

2. The whole node set consists of three mutually exclusive subsets: input (or source) process node set, output (or target) process node set, and intermediate process node set. A process node in CROLES should belong to exactly one of the subsets at a time (or in a session of running). However, the type of a process node may change at another time (or in another session of running).

An input (or source) process node is to represent a business process that captures or generates ultimate source information in the ERP system. Ultimate source information either is generated and captured inside of the whole business process under consideration or comes from outside. A former case is POS data and an example of the second case is EDI data from outside of the enterprise.

An output (or target) process node is to represent a business process which managers or analysts are being concerned about for whatever reasons there might exist. So, it can be a problematic business process at which managers or supervisors feel problems exist and therefore should be analyzed and fixed. Or, it can be just a business process in which users such as executives or managers, trainers or trainees of the ERP system, and analysts for BPR are concerned about for their own purposes.

An intermediate process node is to represent one of the middle steps to reach the target process node from the input nodes. Since there are usually many intermediate processes to reach the target, they collectively determine the behavior of the target.

3. The relationships taken into account in CROLES are just one of a kind such as a causal relationship. The causal relationship between two process nodes in CROLES represents a triggering (or 'invoking' in programming terminology) sequence, i.e. if the preceding process is finished and a certain condition holds true, then the (directly) succeeding process should follow.

As we have seen in the BPM Section of this article, the causal relationship between two process nodes are determined by the business rules of the enterprise. In fact, the whole business process dependency structure of an enterprise is determined by the whole set of business rules of the enterprise and represented as a network topology (See the item (1) in the following description).

Based on the fundamental assumptions about the knowledge of enterprise business processes, CROLES represents its knowledge as follows (design constructs of CROLES).

(1) Network representation of knowledge

The whole knowledge about the enterprise's business processes is represented in a network of process nodes and directed links (the term 'arc' or 'connection' or 'edge' are sometimes used instead of 'link,' but in this article, the term link is used consistently). The whole process nodes and their links constitute the network topology.

(2) Representing the relative importance of the sub-process over the whole business process

Associated with each node is its value from the continuous range of [0, +1], which represents the degree of the importance of the process (node) in terms of its relative criticalness over the whole business process (under consideration or of the whole enterprise), where the value +1 means that it is a process with the highest (utmost) degree of importance over the whole business process under consideration, and where the value 0 means that the process has no contribution (no value addition) to the whole business process under consideration. This number will be called "*PF*" *(Performance Factor)* of the process (node) from now on and will represent the degree of the contribution (value addition) to the whole business process under consideration.

(3) Representing the influence of a process node by weight values

Associated with each link is its value from the continuous range of [0, +1], called "weight." The weight of a link in CROLES measures and reflects the relative degree of influence by the preceding process node on the succeeding process node. The weight value +1 of the link from the preceding process node to the succeeding process means that the preceding process (node) has the maximum degree of influence on the performance (hence, the PF) of the succeeding process (node). Zero weight means that the preceding process (node) has no influence on the PF of the succeeding process (node).

(4) One-node-one-process representation (non-distributed representation)

Each node in CROLES represents a single process. It does not incorporate distributed representations. One possible implication of this principle is that, by adopting this principle, CROLES can be applied to the problem-solving tasks that involve highly abstract and qualitative concepts or objects. The reader who has an interest in the differences between distributed representations versus non-distributed representations may consult Gallant [6].

(5) Acyclic layered network

As we have seen in the BPM Section of this article, the original business process diagram under consideration, if represented in a network, might involve loops or cycles, and thus violates the assumptions made in CROLES. As we have seen in the BPM Section, however, it seems that in real fields there are very rare cases where the business process diagrams include loops or cycles. So it will be safely assumed that out business process model does not involve any loop or cycle as previously explained.

However, if a business process involves loops or cycles, it should be a matter of the abstraction level of the process nodes. For this case, there is suggested a method which can resolve the problem [2][10][11]. In any case, the knowledge network in CROLES, balanced in the abstraction level of the variables, will be safely assumed to be an acyclic network.

(6) Knowledge update by inferencing (update of fact-type knowledge)

CROLES changes (or updates) its knowledge basically in two ways. One way is by inferencing, and the other is by learning. Knowledge updates by inferencing in CROLES mean changing (or updating) the PF values of its nodes (variables). It is assumed that updating of node values occurs simultaneously within a layer, but it occurs serially across the different layers (top-to-bottom or bottom-to-top). In connectionist terms, CROLES is a type of feed-forward network (not a recurrent network).

(7) Knowledge update by learning (update of rule-type knowledge)

CROLES can modify its pattern of connectivity as a part of its learning process. Modifying the pattern of connectivity (thus learning) means changing (or updating) the whole or part of the weight values set where each weight value is associated with each link in CROLES. However, it is assumed that inference and learning are separate actions in CROLES.

4.2 Initialization of CROLES knowledge network

For CROLES to do its main job, inferencing, its knowledge network (knowledge base) should be initialized. The initialization of CROLES knowledge base is just simply the initialization of all its weight values. The weight values initialization in CROLES can be done basically in three ways.

One way is assigning the weight value for each link. The initial weight value of each link may be supplied by human experts who are in charge of the process. Another way is that we calculate the data traffic rate for each link between two process nodes and normalize it, and then assign the normalized value to the weight. The last method is using the CROLES learning mechanism. To use the CROLES learning system, we should have enough training examples. The specific learning method and actions of CROLES are explained in the section on Learning that follows.

4.3 Inference

One of the main purposes of CROLES is to find the relative degree of the performance of a business process and the relationships among different process nodes in terms of contribution or value addition. In other words, managers or analysts who are concerned about the overall performance of the whole business process (under consideration) may want more detailed information about the relative performance or contribution of each component process (more desirably in a quantitative term) and once they get this information, they can do whatever they think appropriate based on the information.

The inferencing mechanism of CROLES can provide that kind of information that managers or analysts need. There are basically four types of inferencing tasks that are possible in CROLES. These inferencing tasks in CROLES can be grouped into two groups, the first group involves forward-chaining computation of connectionist models and the second group involves backward-chaining computation. For fundamental computational mechanisms of connectionist models like forward- and backward chaining and for various learning models, readers may consult the book by Gallant [6]. For hyper fast inference engine architectures see Burns, et al. [2].

1. Forward-chaining computation:
   - Finding the effect of input/source process nodes;
   - Calculating the Performance Factor of the target process nodes;
2. Backward-chaining computation
   - Finding the causes or causal paths for a (or set of) target process node (probably which is a problematic one); and
   - Producing justifications/explanations for the conclusion.

Finding the effect of input/source process nodes means finding the causal paths from a given source process (or set of source nodes) to the target process nodes. This task is needed when managers or analysts want to figure out the exact (partial) process chain and the effect of change in some processes. Calculating the effect of input/source process node is needed when a manager or analyst wants to get information on the effect of the source process' PF (or the change of their values) on the target process.

Finding the causes or causal paths for a (or set of) target process node is similar to the task of finding the effect of input/source process nodes but, in this case, search direction is the opposite one (from target/top to source/bottom). This task is needed when manager or analysts want to figure out an answer to the question such that "Given a process with problem, what process chain do I need to address to resolve the problem?". Producing justification/explanation is similar to the task of finding the causes or causal paths but during the course, CROLES calculates all the PF values along the chain and shows them to the users.

For a more detailed explanation about each of the inferencing tasks presented in this article readers should consult the paper by the author [10][11].

4.4 Learning

Human experts learn their expertise from experience. Learning expertise means building and updating their knowledge structure. Learning in CROLES takes place in a similar manner. Because CROLES represents its knowledge structure as a network of process nodes and relationships between these nodes (K = <V, E>), its structural change assumes adjustment and update of relationships between process nodes. In CROLES, this is done by adjustment of weight values between nodes in different layers. Therefore, learning in CROLES means updates of weight values to enable CROLES to perform correct inference.

Learning in CROLES can take place in two ways. Once the structure of the knowledge network has been determined, a learning process is required to update the weight values because in CROLES an essential part of its knowledge is the weights and the initial weight values are by no means complete and correct. The other way is to execute the learning module of the CROLES in the middle of its operation. This would be required as CROLES builds up cumulative errors or there is a sudden change in the operating environment.

There are many theories, methods, and algorithms for learning in the connectionist paradigm. The learning method adopted in CROLES is the Generalized Delta Rule or back-propagation algorithm as suggested in [17]. Back-propagation is currently the most important and most widely used algorithm for connectionist learning. It is an algorithm for learning in feed-forward networks (like CROLES) using MSE (Mean Squared Error) and gradient descent.

For a detailed introduction to the back-propagation method, readers may consult chapter 11 and 12 of Gallant [6]. For a detailed explanation about the application of the back-propagation method to reasoning system designs, readers may be referred to the author's dissertation [10].

5. ANALYSIS OF THE DISCOVERED KNOWLEDGE NETWORK AND FURTHER RESEARCH

Three-pronged analyses of the knowledge network are possible. First, through the use of simulation and system dynamics, the performance and behavior of the associated process will be assessed. The assessment begins with a determination of the criteria by which performance is assessed (what is the bottom line)? Are there other networks/rules that might achieve better performance? What is the bottleneck or impediment that prevents performance from being better? Are there networks that might perform equally well at considerably less cost? This analysis type would address such questions.

A second possibility is to perform value analysis on the implied value chain embedded within the knowledge network. Such analysis might reveal that a particular node adds no customer-perceived value and thence can be removed. Another possibility might reveal that a simplified knowledge network will achieve the same result, reducing process cost. The problem of complexity in large ERP systems is a huge one and this analysis type alone can have a significant impact on the bottom line. It is complexity that adds so much cost and time to the testing and debugging effort.

Still, a third possibility is to perform carbon/silicon replacement analysis on the network. Such analyses might reveal where a carbon-based processor or node could be replaced with a silicon-based one. This might be possible when the business rules of a carbon-based processor are observed to be deterministic and programmable.

6. SUMMARY

The objective of this research is to automate the process of finding out what information/data professional people and managers are using and how they are reacting/responding to that information. The basic idea is to use a connectionist model developed by the author to get a picture of the structure of the interactions that exist within the process. With regard to process structure, the unknowns are the human information processors in terms of what information they take in as inputs and the information they put out as outputs. The rest is well known; specifically, the business rules are known. Once the process structure is known, there is opportunity to understand why the process behaves as it does and why performance is less than desirable.

References

1. Agerfalk, P.J. and Goeran Goldkuhl, "Business Action and Information Modeling – The Task of the New Millenium," in Information Modeling in the New Millenium, ed. By Matti Rossi and Keng Siau, Idea Group Publishing, Hershey, PA, 2001.
2. Burns, J.R., Wayland Winstead and Brian Hayworth. "Semantic Nets as Paradigms for Both Causal and Judgmental Knowledge Representation," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-19, No. 1, pp. 58-68, January 1989.
3. Cost, K., & K. Slaughter, Best Business: Spanning the Boundaries of Business and IT, Baan Institute Press, The Vanenburg, Putten, The Netherlands, 1997.

4. Curran, Thomas and Gerhard Keller, SAP R/3 Business Blueprint, Prentice Hall, Upper Saddle Hill, NJ, 1998.
5. Gale, Thornton and James Eldred, Getting Results with the Onject-Oriented Model, SIG Books, New York, NY, 1996.
6. Gallant, S.I., Neural Network Learning and Expert Systems, MIT Press, Cambridge, MA., 1993.
7. Hammer, M. "Reengineering Work: Don't Automate, Obliterate", pp. 104-112, Harvard Business Review, July-August, 1990.
8. Hammer, Michael and James Champy, Reengineering the Corporation: A Manifesto for Business Revolution, New York: Harper, 1993.
9. Jonkers, H. and H.M. Franken, "Quantitative Modeling and Analysis of Business Processes," Proceedings of 8th European Simulation Systems: Simulation in Industry, pp. 175-179, Vol. 1, 1996.
10. Jung, Dongill, Design of Inexact Reasoning Systems for Managerial Problem Diagnosis, Ph. D. Dissertation, Texas Tech University, Lubbock, TX, 1990.
11. Jung, Donggill, and James R. Burns, "Connectionist approaches to Inexact Reasoning and Learning Systems for Executive and Decision Support," Decision Support Systems, Vol. 10, pp. 37-66, 1993.
12. Kettinger, W. J. and V. Grover, "Toward a theory of business process change man-agement," Journal of Management Information Systems, Summer, 12, 18, 1995.
13. Klaus, H., Michael Roseman, and Guy G. Gable, "What is ERP, " Information Systems Frontiers, 2:2, pp. 141-162, 2000.
14. Koulopoulos, Thomas M., The Workflow Imperative, New York: Van Nostrand Reinhold, 1995.
15. Martin, J., Principles of Object-Oriented Analysis and Design, Prentice-Hall, Englewood Cliffs, NJ, 1993.
16. Rohloff, Michael, "Reference Model and Object Oriented Approach for Business Process Design and Workflow Management," Proceedings of IEEE, 1996.
17. Rumelhart, David E., Bernhard Widrow, Michael A. Lehr, "The Basic Ideas in Neural Networks," CACM 37(3): pp. 87-92, 1994.
18. Senge, Peter, The Fifth Discipline: The Art & Practice of the Learning Organization, New York, NY, Doubleday Currency, 1990.
19. Scheer, A.-W., Enterprise-Wide Data Modelling, Spring-Verlag, Berlin, Germany, 1989.
20. van Es, Rodim M., and Henk A. Post , Dynamic Enterprise Modeling: A Paradigm Shift in Software Implementation, Baan Business Innovation, 1996.

저자소개

저자 정동길은 현재 명지대학교 경영정보학과의 교수로 재직중이다. 1977 년 서울대 경제학과, 1979 년 서울대 대학원 경영학과, 1981 년 한국과학기술원 산업공학과를 거쳐 1981 년부터 1985 년까지 ㈜대우조선, 데이콤, 성균관대 산업공학과에 재직하였으며, 1990 년 미국 텍사스 텍 대학교에서 경영정보학으로 박사학위 취득하여 귀국한 후 한국전산원에서 컴퓨터통신망과 E1996.01~96.08 Visiting scholar, 미국 AMD(Advanced Micro Device) DI 전자문서 표준화분야에서 일한 후 1993 년 명지대 경영정보학과로 옮겼다.

현재 관심분야는 전자상거래와 가상기업, 그리고 그 관련 분야이다. 특히 전자상거래과정에서 지능적인 에이전트간의 상거래 협상과 가상기업의 형성과 운영에 필요한 여러가지 프로토콜과 그 구조의 연구에 촛점을 두고 있다.