

A Comprehensive Modelling Methodology for the Development of Manufacturing Enterprise Systems

Ang C. L.

Gintic Institute of Manufacturing Technology

Nanyang Technological University

71 Nanyang Drive

Singapore 638075

Tel: 65-799-4921

Fax: 65-791-6377

E-mail: clang@gintic.gov.sg

ABSTRACT

To completely describe a manufacturing system, several models are usually created, each from a different viewpoint. Traditionally, these models are created independently using different methodologies and in different environments. This approach poses many problems, for example, time-consuming, inconsistency between models, difficulties in model maintenance, difficulties in ensuring a seamless transition in the system development life cycle, etc.. To overcome these problems, a comprehensive methodology termed IDEF* has been developed for the modelling of enterprise systems. The purpose of this paper is to introduce to the readers the features and concepts of the methodology.

KEYWORDS

Modeling methodology, manufacturing enterprise systems, CIMOSA, IDEF, IDEF*, IDEF0, IDEF1x, IDEF3, IDEF0-based IDEF2.

1. INTRODUCTION

Models are descriptions of systems. To completely describe a manufacturing enterprise, several models are usually created, each from a different viewpoint. According to CIMOSA reference architecture [AMICE 93], models from four complementary viewpoints are required: function, information, resource, and organisation. Traditionally, these models are built independently using different methodologies and in different environments. However, a methodology tailored to a particular viewpoint is incompatible with another. This traditional approach thus poses the following problems:

- The whole modelling process involves repeated capturing of the same information and is time-consuming.
- It is very difficult to identify the effect of any changes of one model on the others.
- Inconsistency between the different but interrelated models.
- Model maintenance is difficult.
- Seamless transition in the systems development life cycle is difficult.
- It is difficult for system users and system developers to communicate and work together.
- Similarly, it is difficult for several system developers who have different purposes and backgrounds, but who are working on the same system to communicate and work together.

To overcome these problems, a comprehensive modelling methodology termed IDEF* and an associated software tool are being developed in Gintic. The purpose of this paper is to explain to the readers the features and concepts of this methodology.

The methodology is termed IDEF* because it is basically an enhancement of the IDEF (ICAM DEFINITION language) methodology, which was developed by the USAF ICAM programme in the 80's for CIM design and implementation. The IDEF methodology comprises IDEF0, IDEF1, IDEF2, and IDEF3 which are methodologies for functional, information, dynamic, and process modelling, respectively.

2. THE IDEF* METHODOLOGY

The main features of the IDEF* methodology are:

- 1) CIMOSA-compliant;
- 2) IDEF0-based; and
- 3) Integrated.

2.1 CIMOSA COMPLIANCE

The IDEF* methodology conforms to the CIMOSA modelling framework [AMICE 1993], which is a diagrammatic representation of three modelling views (Figure 1).

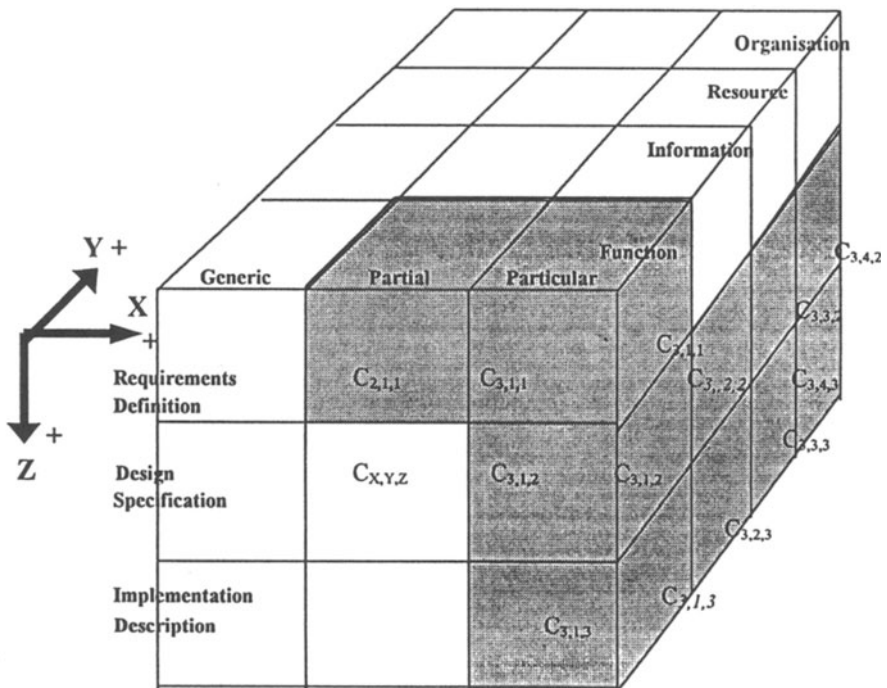


Figure 1: CIMOSA Modelling Framework

In one view, modelling is described as consisting of generic, partial, and particular requirements definition phases. In another view, modelling has function, information, resource, and organisation levels of analysis. Finally, in the third view, modelling is described as having requirements definition, design specification, and implementation description stages of development. The combination of each of these three views results in 36 different modelling domains or cubes $C_{x,y,z}$ s, giving rise to many possible "routes" for enterprise system development. However, not all modelling domains are necessary. Some cubes are

actually redundant. The IDEF* methodology supports the following route, which consists of four sections and is thought to be one of the “shortest” in terms of development effort.

- i.* $C_{2,1,1} \rightarrow C_{3,1,1} \rightarrow C_{3,1,2} \rightarrow C_{3,1,3}$
- ii.* $C_{3,1,2} \rightarrow C_{3,2,2} \rightarrow C_{3,2,3}$
- iii.* $C_{3,1,2} \rightarrow C_{3,3,2} \rightarrow C_{3,3,3}$
- iv.* $C_{3,1,2} \rightarrow C_{3,4,2} \rightarrow C_{3,4,3}$

In IDEF* methodology, section *I* of the development route involves: a) customising a partial function model into a particular function model at the requirement definition stage; b) extending the particular function model at the design specification stage to enable visual and computer workflow analysis; and c) further extending the particular function model at the implementation description stage to enable computer workflow execution. Section *II* involves the derivation of a particular information model from the particular function model at the design specification stage, and the development of relational databases at the implementation description stage. Sections *III* & *IV* involve extending the particular function model to include resource view and organisation view, respectively, at both the design specification stage and the implementation description stage.

2.2 IDEF0-BASED

IDEF0 is chosen as the basis of IDEF* methodology. In producing a complete system description model, an IDEF0 model is first built and other details are then added progressively onto the IDEF0 model so that:

- Functional description of the system can be achieved at any level of abstraction;
- Complete system description can be realised sequentially, and
- Models consistency can be maintained by using the same functions among the different models

Throughout the process, the IDEF0 model is used as a basis for the collection of relevant details. This helps to ensure that models resulting from the process will be compatible with the IDEF0 model, and that only function-related details are collected. This is in contrast with the traditional approach as mentioned in the §1. To meet these requirements of IDEF*, IDEF0 is extended syntactically and textually into IDEF0*.

2.3 INTEGRATION

It is integrated, meaning that it achieves four kinds of modelling integration simultaneously:

- Integration of five modelling processes: function, dynamics, information, resource, and organisation;
- Integration of three modelling stages: requirements definition, design specification, and implementation description;
- Integration of three modelling levels: generic, partial, and particular; and
- Integration of two modelling environments: computer-based enterprise engineering and enterprise operation.

The first two kinds of integration are achieved through an integrated modelling approach in which the different modelling processes share common information on business processes (see §.3, and §.4). The approach will help to reduce the time and effort needed to build the different models that are required for complete system description. It will also help to ensure consistency between these models, and will therefore allow system users and system developers to work together more easily than when using models that are built independently [Kim 1996].

The third kind of integration involves the use of a knowledge-based system to semi-automate the generation of IDEF0 models based on the concept of reference models (see §.5). The knowledge-based approach not only will greatly speed up the IDEF0 modelling process, but will also eliminate the problem of inconsistency that is associated with the traditional approach [Ang et al 1997; Luo 1997].

Finally, the last kind of integration involves extending the IDEF0 model, making it computer-processable not only for workflow analysis but also for workflow execution. This kind of integration

enables real-time implementation of operational changes, hence providing a very flexible operation environment.

3. INTEGRATION OF MODELLING PROCESSES.

This is achieved through an integrated modelling approach in which IDEF0 is chosen as the basis of modelling. Thus an IDEF0 model is first built and other details are then added progressively so that a complete system description can be realised sequentially. To fulfill these requirements, IDEF0 is extended syntactically and textually into IDEF0* in order to accommodate process, dynamic, information, resource, and organisation modelling.

3.1 EXTENSION OF IDEF0 TO INCLUDE PROCESS FLOW MODELLING

To facilitate visual analysis of the IDEF0 models for BPR (business process re-engineering), IDEF0 is extended syntactically so that it can also be used to describe the temporal precedence relationships between the functional activities. In other words, IDEF0* not only can be used to describe what the processes do, but also how they work, i.e. the behaviour aspects.

In IDEF0*, precedence relationships between functional activities are modelled with precedence links (denoted by bold arrows), which connect functional activities by means of *AND* (&), *OR* (O), and *Exclusive OR* (XOR) junction boxes (see Table 1). Figure 2 shows an IDEF0 diagram and its corresponding IDEF0* diagram showing the precedence relationships.

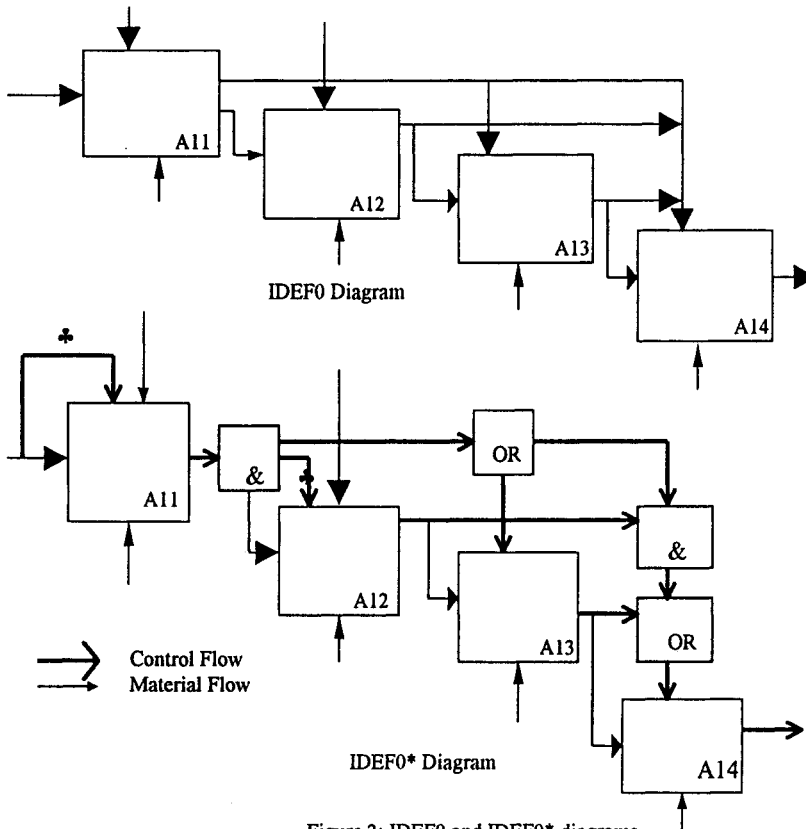


Figure 2: IDEF0 and IDEF0* diagrams.

As can be seen from Figure 2, the IDEF0* diagram is practically formed by superimposing the IDEF3 (PFD) diagram onto the IDEF0 diagram. In the IDEF0* diagram, additional precedence links (arrows marked *) are added to make explicit the fact that some material flows in the IDEF0 diagram are also control flows. This fact is implicit in the IDEF0 diagram.

3.2 EXTENSION OF IDEF0 TO INCLUDE DYNAMICS MODELLING

To facilitate performance analysis, dynamics parameters are incorporated into the IDEF0 model. They are incorporated as text files attached to the relevant functions or arrows of the IDEF0 model and contained within the Text or Glossary of the model. This has the advantages of not complicating the IDEF0 diagrams. Such performance-related parameters as arrival rate, quantity of each resource type, branching conditions or behaviour rules, etc. can be incorporated. The resulting IDEF0* model, which is basically an IDEF0-based IDEF2 model, can be simulated for workflow analysis [Shunk 1990, Popplewell et al 1994].

3.3 DERIVATION OF IDEF1x MODELS FROM IDEF0 MODELS

IDEF0 models describe the functions performed by business processes. A function transforms its inputs into outputs under the supervision of its controls by using its mechanisms. Although a function's inputs and outputs can be information or physical objects, they are represented only at the level of graphical labels and no actual information structures are attached to those labels. An IDEF1x model on the other hand graphically represents the information content and structure related to a business process or an enterprise. However, it is possible to derive an IDEF1x model from an appropriate IDEF0 model using the Glossary of the IDEF0 model as the entity pool for IDEF1x [Le Clair 1982, Smart et al 1996]. A procedure for the systematic derivation of IDEF1x models from IDEF0 models has been developed by the author [Ang 1997]. The procedure is adapted from the work of [Adam et al 1993].

3.4 EXTENSION OF IDEF0 TO INCLUDE RESOURCE MODELLING

The term resource is restricted to mechanisms in the sense of IDEF0, i.e. enterprise objects contributing to the realisation of the functionality of activities of business processes [Vernadat 1996]. In IDEF0*, description templates, which are subsets of those of the two CIMOSA resource modelling constructs: Capability Set and Resource [AMICE 1993], are used to deal with resource modelling (Figure 3). They are attached to the relevant mechanism arrows as text files and contained within the Glossary of the IDEF0 model. Again, they are input via pop-up boxes in the associated software tool being developed.

Capability Set		
Design Authority:		[Name of person with authority to design/maintain this particular Capability Set].
Capabilities		
Function Related:		[Capability elements]
Performance related:		[Capability elements]
Objective related:		[Capability elements]
Resource		
Design Authority:		[Name of person and department with authority to design/maintain this particular instance]
Capability Set:		[Name of Capability Set defining the provided capabilities of this Resource instance]
Class:		[Functional Entity or Resource Cell or Resource Set or Resource Component]
Operation Set:		[Functional operations valid for this Resource instance]
Object View:		[Name of Object View instance defining the characteristics (capacity, availability, location, etc.) of this Resource instance]

Figure 3: Description templates for resource modeling.

3.5 EXTENSION OF IDEF0 TO INCLUDE ORGANISATION MODELLING

In IDEF*, IDEF0 is also extended both syntactically and textually to deal with organisation modelling. Thus an IDEF0* diagram may have, in addition to the *AND*, *OR*, *XOR* junction boxes as mentioned in

§3.1, eight basic elements (see Figure 4) which are defined as follows. Here distinction is made between line authority (control) and staff authority (authority), and between the person who owns the process and the person who executes the process [Born 1994]. The person who executes a process is a resource termed Functional Entity in CIMOSA [AMICE 1995].

- Process: A series of steps or actions which transform an object from an initial state (input) to a final state (output).
- Process owner: The person or other agent responsible for the execution of a process.
- Organisation Cell (see below): The organisation cell to which the person is assigned.
- Staff authority (authority): The description, specification or justification of a process. It is shown entering the process box at the top right.
- Line authority (control): The conditions for activating process. It is shown entering the process box at the top left.
- Input: An object at the point of entry to a process.
- Output: An object at the point of exit from a process.
- Mechanisms: The resources used to execute the process.

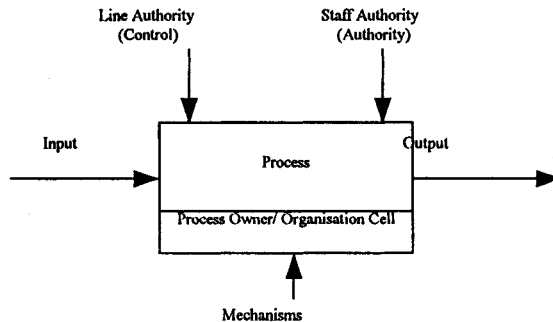


Figure 4: IDEF0* process notation

In addition, IDEF0* also uses two types of description templates for organisation modelling (Figure 5). They are attached to the relevant function boxes, again as text files, and contained within the Text of the IDEF0 model. These description templates are subsets of those of the two CIMOSA organisation modelling constructs, namely Organisation Unit and Organisation Cell [AMICE 1993].

Organization Unit	
Design Authority:	[Name of person and department with authority to design/maintain this particular instance]
<u>Function Description</u>	
Functional Entity:	[Name of functional entity associated to this organization unit]
Job Unit Description:	[Textual description of the decision-making or problem solving job]
Skill profile:	[List of capabilities/skills which are needed to fulfill the job]
Responsibilities:	[List of responsibilities given to this organization unit]
Authorities:	[List of authorities necessary to fill the job]
<u>Structural Description</u>	
Assigned to organization cell:	[Name of organization cell to which the organization unit belongs]
Organization Cell	
Design Authority:	[Name of person and department with authority to design/maintain this particular instance]
<u>Function Description</u>	
Cell Description:	[Textual description of the purpose of the organization cell]
Operational Authority/	
Responsibility:	[Organization unit responsible for the organization cell. Must be a person]
Process	
Authorities/	
Responsibilities:	[List of function view constructs on which this organization cell has responsibility and authority]
Information	
Authorities/	

Responsibilities:	[List of information view constructs on which this organization cell has responsibility and authority]
Resource Authorities/ Responsibilities:	[List of resource view constructs on which this organization cell has responsibility and authority]
Organization Level:	[One of: "Enterprise", "Plant", "Shop", "Cell", "Workcenter", or "Equipment"]

Figure 5: Description templates for organization modeling.

At this stage, the IDEF0* model contains sufficient information for the construction of an organisation chart (Figure 6).

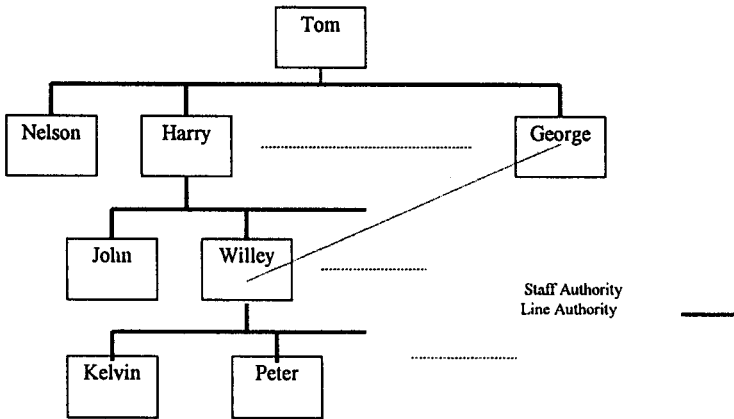


Figure 6: Organisation chart

4. INTEGRATION OF MODELLING STAGES

This section describes the integration of modelling stages of each of the four modelling processes.

4.1 FUNCTIONAL MODELLING

At the requirements definition stage, an IDEF0 model is first built. It is then extended into an IDEF0* model at the design specification stage to include details like time, resource, and behaviour of each enterprise activity and functional operation (resources in terms of functional entities providing the functional operations are also specified in the resource view using the description templates mentioned in §3.2). At the implementation description stage, the same IDEF0* model is used and enriched. In particular, ports for data communications are defined for each activity to indicate the exact entry points for receiving/sending inputs and outputs. Also, if messages are exchanged among activities, a communication channel and the format of messages must also be described. Finally, the IDEF0* model is supplemented by network diagrams and block diagrams to facilitate data communications system planning and facility layout planning, respectively.

4.2 INFORMATION MODELLING

Database design is more than just building entities and connecting them by relationships which only represent the static part of the information system. The dynamic part, i.e., the way data and information are going to be processed, must also be considered [Vernadat 1996]. In IDEF*, the static part is modelled by two constructs: Conceptual Schema and External Schema; while the dynamic part by another two constructs: Integrity Rule and Database Transaction. The modelling framework is compliant with the

three-schema approach proposed by ANSI (American National Standards Institute) [ANSI 1976], which advocates for the use of a global conceptual schema implemented in terms of an internal schema and presented to the system users via external schema.

In IDEF* methodology, IDEF1x models are derived from IDEF0* models. At the design specification stage, an ERA-based global conceptual schema is first produced (see §3.3 and Figures 4c & 4d). From the conceptual schema, external schemata can then be derived. In addition, the integrity rules that constrain the possible values taken by attributes can also be specified. These rules are expressed in terms of natural language of expressions. They can be derived from the declarative rules of business processes or they can be specified by users as separate detailed information requirements.

Once the conceptual schema and its external schemata have been produced, data and database transactions, i.e. data processing functional operations or queries on data, can be specified using IDEF0*. Like the M* Process Description Model [Vernadat et al 1989], the IDEF0* model mentioned §3.2 is a formal model which can be used to specify data manipulation operations to be executed on data, either by application programs, or directly by users in the form of data transactions.

Implementation description involves both logical database design and physical database design. In logical database design, the conceptual schema is mapped onto a relational schema using the conversion rules developed by [Elmasri et al 1993](see §3.3). The relational schema is then implemented as an internal schema in physical database design using the Structured Query Language (SQL). Physical database design also involves coding the integrity rules, data and database transactions of applications programs, and on-line queries using SQL for data manipulation and query. SQL is both the data description language, and the data manipulation and query language for well-known relational database systems like Oracle, DB2, Ingres, and Informix.

4.3 RESOURCE MODELLING

For resource modelling, the same IDEF0* model as for functional modelling (see §3.1) is used and extended to include two types of description templates, namely Capability Set and Resource (see §3.4). At the requirements definition stage, only the Capability Set template is used to express resource requirements from the standpoint of enterprise activities (i.e. required capabilities). At the design specification stage and the implementation description stage, both the Capability Set and Resource templates are used and are respectively qualified as specified and implemented.

The Capability Set in CIMOSA is an essential construct for supporting the principle of process and resource decoupling (see §6). The principle is based on the fact that activities, and therefore processes, require capabilities for their execution while resources provide capabilities to execute functional operations of activities, if they are available and have the necessary capacity.

4.4 ORGANISATION MODELLING

For organisation modelling, the same IDEF0* model as for functional modelling (see § 3.1) is again used and extended. As described in §3.5, it is extended to include additional graphical notations, and two types of description templates:, namely Organisation Units and Organisation Cells. The additional graphical notations together with the description templates are supposed to apply at the design specification modelling stage and at the implementation description modelling stage. This is especially true for designing or restructuring an organisation. However, they can be used at the requirements definition stage in the case of the modelling of an existing system.

5. INTEGRATION OF MODELLING LEVELS

Integration of the generic, partial, and particular modelling levels is achieved by means of a knowledge-based system for the automated generation of IDEF0 models based on the concepts of reference models. Reference models represent structures and practices that are typical of a particular industry or a set of companies within that industry. They can be customised to partial models or company-specific models and hence can be shared and reused. It has been reported by Baines [1989], Weaver [1995], and Vernadat [1996] that the use of reference function models can facilitate and accelerate the functional modelling process.

An operational prototype of the knowledge-based system has been developed in Gintic [Ang et al 1997, Luo 1977]. The work involved: 1) the development of a structured methodology for the automated

generation of IDEF0 models and its associated knowledge base; and 2) their implementation in computer software.

9. CONCLUSION

In this paper, a comprehensive modelling methodology IDEF* for manufacturing enterprise systems development has been proposed. The methodology, which is based on an integrated modelling approach, incorporates such robust modelling methodologies as IDEF0, IDEF3(PFD), and IDEF1x. It conforms to the CIMOSA modelling framework, and simultaneously achieves four kinds of integration, namely integration of modelling processes, integration of modelling stages, integration of modelling levels, and integration of modelling environments. When supported by appropriate software tools, it is expected to drastically reduce the time and effort of enterprise systems development.

REFERENCES

- Adam, N. R., and Ganopadhyay, A., Integrating Functional and Data Modeling in a Computer Integrated Manufacturing System, IEEE 9th International Conference on Data Engineering, 1993.
- AMICE Consortium, CIMOSA Open Systems Architecture for CIM, Springer Verlag, Berlin 1993.
- Ang, C.L., Luo M., and Gay Robert K.L., A Knowledge-Based Approach to the Generation of IDEF0 Models, International Journal of Production Research, Vol. 35, No 5, 1997.
- Ang C.L., Derivation of IDEF1x Models from IDEF0 models, In-house Technical Report WP/RIC-U/SD/ST/ACL/97/03, Gintic Institute of Manufacturing Technology, 1977.
- ANSI, ANSI/X3/SPARC, Study Group on Database Management Systems: Interim Report, 75-02-08, In: ACM SIGMOD Newsletter, FDT, Vol. 7, No. 2, 1975.
- Baines, R.W., and Colquhoun, G. J., The Use of a Generic IDEF0 Model to Capture Process Planning Functional Information, Proceedings of the Sunderland Advanced Manufacturing Technology Conference, Sunderland, 1989.
- Born, C., Process Management to Quality Improvement, Wiley, 1994.
- FIPS PUBS, Integration Definition for Function Modeling (IDEF0), Federal Information Processing Standards Publication 183, National Institute of Standards and Technology, USA, 1993.
- Kim J.I, Function, Information., Dynamics, and Organization Integrated Modeling Methodology for Enterprise Systems Integration, PhD Thesis, Arizona State University, 1996.
- Luo, M., A Knowledge-Based Approach to the Automated Generation of IDEF0 Models, PhD Thesis, in advanced preparation, 1997.
- Popplewell, K., et al, A Comprehensive Methodology for Factory Modeling, 1994.
- Smart, A., Maull, R., Childe, S., Bennett, J., and Weaver, A., Guidelines for Linking Process and Information Models - IDEF0 to IDEF1x, Working Paper WP/GR/J95010-7, 1996.
- Shunk, D.L., Creating an Integrated, Useful Systems Definition Technique, Optimization of Manufacturing Systems Design, Edited by Shunk, D. L., Elsevier Science Publishers B.V., 1993.
- Vernadat, F., DiLeva, A., and Giolito, P., Organization and Information System Design of Manufacturing Environments: the New M* Approach, Journal of Computer-Integrated Manufacturing Systems, Vol. 2, No. 2, 1989.
- Vernadat, F. B., Enterprise Modeling and Integration, Principles and Applications, Chapman & Hall, 1996.
- Wang, W., Popplewell, K., and Bell, R., An Integrated Multi-View System Description Approach to Approximate Factory Modeling, International Journal of Computer Integrated Manufacturing, Vol. 6, No.2, 1993.
- Weaver, A.M., Childe, S.J., Maul, R.S., and Smart, P.A., The Development and Application of a Generic "Order Fulfillment" Process Model, Proceedings of the Third International Conference on CIM, Singapore, July 1995.