A FORMAL APPROACH TO ENTERPRISE MODELING

Yoshiyuki Shinkawa
Department of Media Informatics, Ryukoku University
1-5 Seta Oe-cho Yokotani, Otsu, Shiga, Japan

Keywords: Enterprise modeling, model consistency, model evaluation, enterprise information systems.

Abstract: Model driven development for software systems provides us with many advantages in quality, productivity, or reusability. For accurate modeling, we have to create many kinds of models from various viewpoints. When applying model driven development to enterprise information systems, those viewpoints include not only software oriented matters but also business oriented matters. Such complexity in modeling often causes inconsistency between models. This paper presents a formal and systematic way to create consistent and integrated enterprise models that reflect those various viewpoints. Set theory, Colored Petri Nets (CPNs), and Unified Modeling Language (UML) are used for this formalism. In addition, the paper proposes a set theoretic approach to evaluating consistency between enterprise models. The consistency is discussed in traditional hierarchical organization and modern matrix organization.

1 INTRODUCTION

Today's large scale enterprise information systems are often built up through model based or model oriented approaches, so that various complicated requirements are implemented accurately into the information systems (Frankel, 2003).

Since late 1980's or early 1990's, there have been proposed many kinds of enterprise modeling frameworks which are equipped with the linkage to software development methodologies (Vernadat, 1996). Those frameworks include CIMASA (Kosanke, 1992), GRAI /GIM (Doumeingts et al., 1994), ARIS (Scheer, 1999), PERA/GERAM (Williams and Hong, 1998), and so on.

As there are many viewpoints or aspects of an enterprise, those frameworks claim to create multiple models according to the viewpoints, e.g., *function*, *resource*, *data*, and so on. However, those models are often tightly interrelated, and if they are built up independently by the isolated different groups, there could exist a lot of inconsistencies between them.

This paper presents a systematic approach to making those enterprise models consistent within an enterprise, and gives a formal way to evaluate the consistencies.

2 A BASIC MODEL STRUCTURE

An enterprise is a very complex reality which includes many resources, activities, processes, rules, regulations, constraints, objectives, missions, organizations, and so on. Therefore, enterprise models have to represent such a complex reality, and enterprise modeling becomes a complicated and difficult task. There are many consideration points that have to be taken into account in enterprise modeling.

This paper defines the three groups of them, namely, *modeling lifecycle*, *decision levels*, and *kind of interest*. Those groups are referred to as *axes* that compose an orthogonal coordination system, and multiple *values* or *entries* are defined on those axes.

The *modeling lifecycle* axis represents a modeling process which consists of the *requirement analysis* phase, the *conceptual modeling* phase, and the *implementation* phase.

The *decision level* axis represents managerial hierarchy in an enterprise, which consists of the *strategic* level, the *management control* level and the *operational* level.

The third axis, *kind of interest* axis, looks differently depending on which phase is focused on. In the *requirement analysis* phase, this axis is composed of the following business oriented matters, that is, *re-*

Shinkawa Y. (2004).
A FORMAL APPROACH TO ENTERPRISE MODELING.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 663-668
DOI: 10.5220/0002626306630668
Copyright © SciTePress

source, *organization*, *function*, *activity*, and *process*. On the other hand, in the *implementation* phase, the axis consists of more software or system oriented matters. The models in this phase are expressed as specifications written in appropriate specification tools. The paper adopts UML as a specification tool, and regards *kind of interest* axis as being composed of the *class diagram*, *activity diagram*, and *sequence diagram*.

In the *conceptual modeling* phase, the models should be neutral from the above two viewpoints, the business and software viewpoints. The paper regards the models in this phase as composing the transformation layer between the business oriented models and the software oriented models.

This framework resembles CIMOSA cube. However, CIMOSA cube has a symmetric model structure between the business view and the information system view, on the other hand, the proposed model has asymmetric model structure between them.

Since three dimensional model frameworks are difficult to view and understand, the framework in this paper is divided into two parts.

	Requirement Analysis	Conceptual Modelig	Implementation
Strategic	MS_{SR}	MS_{SC}	MS_{SI}
Management Control	MS_{MR}	MS_{MC}	MS_{MI}
Operational	MS_{OR}	MS_{OC}	MS_{OI}

Figure 1: Modeling Framework 1

One is the framework with *modeling lifecycle* axis and *decision level* axis, which is shown in Figure 1.

The modeling process usually follows the $MS_{Sx} \rightarrow MS_{Mx} \rightarrow MS_{Ox}$ path and $MS_{yR} \rightarrow MS_{yC} \rightarrow MS_{yI}$ path. However, as for the former path, that is, the *top down deployment*, enterprise modeling activities usually do not participate with it. This deployment is a part of *enterprise design* or *business design*. Therefore, this paper only deal with the latter path, and it assumes MS_{SR}, MS_{MR}, and MS_{OR} are modeled independently.

The other is the model framework with the *model lifecycle* axis and *kind of interest* axis, which is shown in Figure 2.

The relationships between the models in Figure 1 and Figure 2 are as follows.

$$MS_{xR} = \{RA_{xR}, RA_{xO}, RA_{xF}, RA_{xA}, RA_{xP}\}$$
$$MS_{xC} = CM_x$$
$$MS_{xI} = \{IM_{xC}, IM_{xA}, IM_{xS}\}$$

The entries in Figure 2, e.g., RA_{xR} and CM_x, are called *model constituents* in this paper.

Once the model framework is defined, the next steps are creating actual enterprise models which conform to the framework, and evaluating or validating those models. The next section discusses how the conforming enterprise models are created.

3 CREATING CONFORMING MODELS

According to the model framework discussed in the previous section, we have to create the following models.

1. Resource/Organization/Function/Activity/Process models for each decision levels, in the requirement analysis phase. Those are the models from a business view.
2. Concept models for each decision level, in the conceptual modeling phase.
3. Class diagrams, activity diagrams, and sequence diagrams for each decision level in the implementation phase. Those are the models from a information system view.

This section shows how those models are created.

3.1 Model Creation from a Business View

In order to create enterprise models from the business view rigorously, we first have to define the meaning of the terms *resource*, *organization*, *function*, *activity*, and *process*.

One of the ways to formalize those terms is to express them using the rough set theory (Pawlak, 1992).

The first two constituents can be derived using *equivalence relations*[1] in the following way (Shinkawa and Matsumoto, 2001).

1. The definition of *resource* (RA_{xR})
   - (a) Let U be a set of all the externally observable substances in an enterprise. This U is identical between the decision levels.
   - (b) A resource is a class of U in terms of set theory, which is defined by an equivalence relation over U. This equivalence relation represents a piece of knowledge on the resources in an enterprise.
   - (c) Assuming $\mathcal{R}_i(x)$ is an equivalence relation over U which corresponds to a piece of knowledge at the decision level x, the resources that are defined by $\mathcal{R}_i(x)$ is denoted by

[1]A relation $R \subseteq U \times U$ which is reflexive, symmetric and transitive.

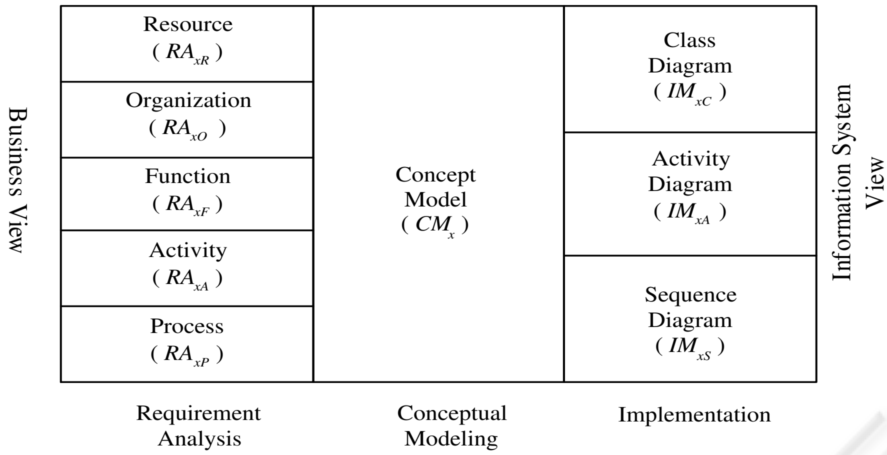

Figure 2: Modeling Framework 2

$U/\mathcal{R}_i(x) = \{U_1^{(i)}(x), \ldots, U_{m_i}^{(i)}(x)\}$
where $U/\mathcal{R}_i(x)$ represents the classification of the set U by the equivalence relation $\mathcal{R}_i(x)$. Each $U_j^{(i)}$ means a resource.

(d) The resource model constituent at the decision level x is a set of the above pieces of knowledge expressed in the equivalence relations, that is,
$RA_{xR} = \{\mathcal{R}_i(x)\}$

2. The definition of *organization* (RA_{xO})
   (a) Let V be a set of all the people in an enterprise. This V is identical between the decision levels.
   (b) In the similar way to defining the resource model constituent, the organization is formalized as
   $V/\mathcal{Q}_i(x) = \{V_1^{(i)}(x), \ldots, V_{n_i}^{(i)}(x)\}$
   where $\mathcal{Q}_i(x)$ is an equivalence relation over V at the decision level x.
   (c) The organization model constituent at the decision level x is denoted by
   $RA_{xO} = \{\mathcal{Q}_i(x)\}$
   as we did for RA_{xR}.

The third model constituent *function* (RA_{xF}) is defined as a transformation rule between resources. Such a transformation is formalized as an S-sorted function of many-sorted algebra (Astesiano et al., 1999). A function at the decision level x is denoted by

$$F_i(x):\ U_{i1}(x) \times \cdots \times U_{in_i}(x) \longrightarrow U_i(x)$$

where $U_{ij}(x)$ and $U_i(x)$ are the resources defined at the decision level x in the above way. The function model constituent RA_{xF} is a set of all the functions at the decision level x, that is,

$$RA_{xF} = \{F_i(x)\}$$

The fourth model constituent *activity* (RA_{xA}) is defined as a pair of a function and an organization that performs the function. At decision level x, an activity thus defined is denoted by a tuple of a function and an organization

$$A_i(x) = \langle F_j(x), V_k(x)\rangle$$

and the activity model constituent RA_{xA} is defined as

$$RA_{xA} = \{A_i(x)\}$$

The last model constituent *process* (RA_{xP}) is defined as a partially ordered sequence of activities. Such an activity is expressed as a set of tuples $\langle$ *preceeding activities*, *center activity*, *succeeding activities* $\rangle$. Therefore a a process at the decision level x can be denoted by

$$P_i(x) = \{\langle \overline{A}_{j1}^{(i)}(x), \ldots, \overline{A}_{jm_{ij}}^{(i)}(x), A_j^{(i)}(x), \underline{A}_{j1}^{(i)}(x), \ldots, \underline{A}_{jn_{ij}}^{(i)}(x)\rangle\}$$

and the model constituent RA_{xP} is denoted by

$$RA_{xP} = \{P_i(x)\}$$

where $\overline{A}_k^{(j)}(x)$ and $\underline{A}_k^{(j)}(x)$ represent a preceding activity and a succeeding activity of the activity $A_i^{(j)}(x)$ respectively. The m_{ij} or the n_{ij} is zero if $A_j^{(i)}(x)$ is an initial or a terminal node.

3.2 Model Creation from a Conceptual View

This paper uses Colored Petri Nets (CPNs) for conceptual modeling. CPNs can be conveniently used for expressing workflows and business processes (Aalst and Hee, 1995)(Aalst, 1998). In our case, the processes correspond to them, and the processes include all the model constituents by definition.

SciTePress Science and Technology Publications

CPNs are one of the enhancements of Petri nets, and formally defined as follows (Jensen, 1997).

CPN=(S, P, T, A, N, C, G, E, I) ,

where

S : a finite set of non-empty types, called color sets,

P : a finite set of places,

T : a finite set of transitions,

A : finite set of arcs $P \cap T = P \cap A = T \cap A = \emptyset$,

N : node function $A \to P \times T \cup T \times P$,

C : a color function $P \to S$,

G : a guard function $T \to$ expression,

E : an arc expression function $A \to$ expression and

I : an initialization function : $P \to$ closed expression.

In order to transform a process in RA_{xP} into a CPN model, we introduce the following basic rules (Shinkawa and Matsumoto, 2000). Let the original process be

$$P_i(x) = \{\langle \overline{A}^{(i)}_{j1}(x), \ldots, \overline{A}^{(i)}_{jm_{ij}}(x), A^{(i)}_j(x), \underline{A}^{(i)}_{j1}(x), \ldots, \underline{A}^{(i)}_{jn_{ij}}(x)\rangle\}$$

and let the resultant CPN model be

$$CPN_i(x) = \big(S_i(x), Pl_i(x), T_i(x), A_i(x), N_i(x), C_i(x), G_i(x), E_i(x), I_i(x)\big)$$

where x is a decision level.

1. $T(x)$ is a set of the activities that occur in the $P_i(x)$ as A, $\overline{A}$, or $\underline{A}$.
2. $Pl_i(x)$ is a set of the organizations that are included in A, $\overline{A}$, or $\underline{A}$ in the form of
$A_i(x) = \langle F_j(x), V_k(x)\rangle$,
where $V_k(x)$ represents an organization.
3. A tuple $\langle \overline{V}^{(i)}_{jk}(x), A^{(i)}_j(x)\rangle$, is a member of $N(x)$, iff $\overline{A}^{(i)}_{jk}(x) = \langle \overline{F}^{(i)}_{jk}(x), \overline{V}^{(i)}_{jk}(x)\rangle$ is one of the preceding activities of $A^{(i)}_j(x)$.
4. A tuple $\langle A^{(i)}_j(x), \underline{V}^{(i)}_{jk}(x)\rangle$, is a member of $N(x)$, iff $\underline{A}^{(i)}_{jk}(x) = \langle \underline{F}^{(i)}_{jk}(x), \underline{V}^{(i)}_{jk}(x)\rangle$ is one of the succeeding activities of $A^{(i)}_j(x)$.
5. The union of the above two sets of tuples,
$N(x) = \{\langle \overline{V}^{(i)}_{jk}, A^{(i)}_j(x)\rangle\} \bigcup \{\langle A^{(i)}_j(x), \underline{V}^{(i)}_{jk}\rangle\}$
determines the structure of the CPN model.
6. $S_i(x) = U/RA_{xR}$ and $Pl_i(x) = V/RA_{xO}$, where U/RA_{xR} means all the classes made by the equivalence relations included in RA_{xR}.
7. $A_i(x)$ is a set of arc names, and we can give arbitrary names to the arcs defined by $N_i(x)$.
8. $C_i(x)$ represents the associated colors to each place, and it means the required resource types for the transition that receive the tokens from the place. This association is imbedded in each
$A_i(x) = \langle F_j(x), V_j(x)\rangle$ and
$F_j(x) :\ U_{j1}(x) \times \cdots \times U_{jm_j}(x) \longrightarrow U_j(x)$.
$\big(U_{jk}(x) \in C_i(x)\big)$
9. $E(x)$, an arc expression function represents the function that each activity performs, therefore it can be derived from $A_i(x) = \langle F_j(x), V_k(x)\rangle$.
10. $G_i(x)$, a guard function, and $I_i(x)$, an initialization function, are derived from the constraint statements $\mathcal{C}$.

3.3 Model Creation from a Information System View

CPN models are too abstract to be implemented into enterprise information systems. In addition, most software developers, e.g., programmers, system designers, or systems engineers, seem not to be familiar with CPN. Therefore, those models should be transformed into the models written in more appropriate methods for software development. UML (Unified Modeling Language) is one of the most popular tools to design software and systems.

Although many kinds of diagrams are provided by UML, the most essential ones for enterprise modeling are *class diagram*, *activity diagram*, and *sequence diagram*. Some other diagrams, such as component diagrams, deployment diagrams, or state chart diagrams, are to be created after the enterprise modeling, while use case diagrams are considered to be embedded in CPN models.

The roles of places, transitions, arc expression functions, and guard functions discussed in the previous section suggests the following transformation rules from a CPN model to UML models.

1. A class is an entity that provides some functionality. This role is taken by a pair of an input place and a transition in our CPN models. Therefore, each pair of an input place and a transition compose a class, where arc expression functions are regarded as *methods*. The attributes in the class are defined by colors associated to the tokens that make the transition fire.
2. The role of an activity in UML is approximately the same as that of a transition in CPN models. Since activity diagrams show the execution sequences of those activities, they are formed by deleting the places from the CPN models, and connecting the transitions directly.
3. The role of a sequence diagram is to describe the message passing between objects. The places in

CPN models can be regarded as those objects that send and receive messages as color tokens. Therefore, sequence diagrams are formed by deleting the transitions from the CPN models, and connecting the places directly.

Since the symbols and notation of CPN and UML are different from each other, we have to transform them into appropriate forms that conform to UML. The above UML models are to be treated as the prototypes of IM_{xC}, IM_{xA}, and IM_{xS} in Figure 2. Those models should be refined according to the actual constraints of the information systems to be built.

4 MODEL CONSISTENCY

The previous discussion on model transformation does not take the model consistency along *decision level* axis into account. This section deals with the consistency along that axis.

4.1 Hierarchical Enterprise Structure

In the *requirement analysis* phase, all the model sets are expressed as a set of equivalence relations or sets of tuples, regardless of the decision levels. Since the *decision level* axis represents *generalization/specialization* as we discussed in section 2, the consistency along this path can be defined by relations of inclusion as follows.

1. *Resource* and *Organization*
   Let $RA_{xR} = \{\mathcal{R}_i(x)\}$ and $RA_{xO} = \{\mathcal{Q}_i(x)\}$ be the model constituents of resources and organizations respectively, where x represents a decision level. For model consistency between the decision levels,
   $$\forall \mathcal{R}_i(O) \in RA_{OR}$$
   $$\exists \mathcal{R}_j(M) \in RA_{MR}\ \exists \mathcal{R}_k(S) \in RA_{SR}$$
   $$[\mathcal{R}_i(O) \subset \mathcal{R}_j(M) \subset \mathcal{R}_k(S)]$$
   and
   $$\forall \mathcal{Q}_i(O) \in RA_{OO}$$
   $$\exists \mathcal{Q}_j(M) \in RA_{MO}\ \exists \mathcal{Q}_k(S) \in RA_{SO}$$
   $$[\mathcal{Q}_i(O) \subset \mathcal{Q}_j(M) \subset \mathcal{Q}_k(S)]$$
   must hold. Those conditions represent that the extent of resources or organizations in a higher decision level must be wider than that in a lower level.
2. *Function*
   Let $RA_{xF} = \{F_i(x)\}$ be a model constituent of functions. For consistency,
   $$\forall F_i(O) \in RA_{OF}$$
   $$\exists F_j(M) \in RA_{MF}\ \exists F_k(S) \in RA_{SF}$$
   $$[F_i(O) \subset F_j(M) \subset F_k(S)]$$
   must hold. This condition also represents the extent of functions between the decision levels.
3. *Activity*
   Let $RA_{xA} = \{A_i(x)\} = \{\langle F_j(x), V_k(x)\rangle\}$ be a model constituent of activities. For consistency
   $$\forall A_i(O) = \langle F_j(O), V_k(O)\rangle \in RA_{OA}$$
   $$\exists A_{i'}(M) = \langle F_{j'}(M), V_{k'}(M)\rangle \in RA_{MA}$$
   $$\exists A_{i''}(S) = \langle F_{j''}(S), V_{k''}(S)\rangle \in RA_{SA}$$
   $$[F_j(O) \subset F_{j'}(M) \subset F_{j''}(S) \bigwedge V_k(O) \subset V_{k'}(M) \subset V_{k''}(S)]$$
   must hold.
4. *Process*
   Let
   $$RA_{xP} = \{P_i(x)\} = \{\{\langle \overline{A}^{(i)}_{j1}(x), \ldots, \overline{A}^{(i)}_{jm_{ij}}(x), A^{(i)}_{j}(x), \underline{A}^{(i)}_{j1}(x), \ldots, \underline{A}^{(i)}_{jn_{ij}}(x)\rangle\}\}$$
   be a model constituent of processes. For consistency,
   $$\forall P_i(O) = \{\langle \overline{A}^{(i)}_{j1}(O), \ldots, \overline{A}^{(i)}_{jm_{ij}}(O), A^{(i)}_{j}(O), \underline{A}^{(i)}_{j1}(O), \ldots, \underline{A}^{(i)}_{jn_{ij}}(O)\rangle\} \in RA_{OP}$$
   $$\exists P_{i'}(M) = \{\langle \overline{A}^{(i')}_{j'1}(M), \ldots, \overline{A}^{(i')}_{j'm_{i'j'}}(M), A^{(i')}_{j'}(M), \underline{A}^{(i')}_{j'1}(M), \ldots, \underline{A}^{(i')}_{jn_{i'j'}}(M)\rangle\} \in RA_{MP}$$
   $$\exists P_{i''}(S) = \{\langle \overline{A}^{(i'')}_{j''1}(S), \ldots, \overline{A}^{(i'')}_{j''m_{i''j''}}(S), A^{(i'')}_{j''}(S), \underline{A}^{(i'')}_{j''1}(S), \ldots, \underline{A}^{(i'')}_{jn_{i''j''}}(S)\rangle\} \in RA_{SP}$$
   $$[(\overline{A}^{(i)}_{jk}(O) \subset \overline{A}^{(i')}_{i'j'}(M) \subset \overline{A}^{(i'')}_{i''j''}(S)) \bigwedge (A^{(i)}_{j}(O) \subset A^{(i')}_{j'}(M) \subset A^{(i'')}_{j''}(S)) \bigwedge (\underline{A}^{(i)}_{jk}(O) \subset \underline{A}^{(i')}_{i'j'}(M) \subset \underline{A}^{(i'')}_{i''j''}(S))]$$
   must hold.

The above conditions assure the model consistency along the *decision level* axis in the *requirement analysis* phase on the *modeling lifecycle* axis, The model consistency in this phase guarantee the model consistency in the *conceptual modeling* phase and the *implementation* phase, since all the models in those phases are derived consistently from the model constituents in the *requirement analysis* phase.

4.2 Non-Hierarchy Enterprise Structure

The model consistency discussed in the previous section assumes the simple hierarchical organization structure, and the *decision level* axis represents *generalization/specialization* process. In modern business environment, many enterprises adopt more efficient forms of organization for swift decision-making. A typical one of those new organization forms is *matrix organization*. A matrix organization is an organization which has *multi-dimensional* decision paths. The simplest form of the matrix organization is a two-dimensional organization, e.g., an organization with the functional decision path and the regional decision path.

In matrix organization, the *decision level* axis splits into the multiple axes corresponding to the decision paths defined in that organization. In addition, while the classical organization theory proposed three managerial levels or decision levels, today's complex enterprise organization includes variable number of those levels, more or less than three. From those discussions, the models of an enterprise with such an organization are formalized as follows.

1. Let the decision path axes be $X_1, \ldots, X_p$, where each path X_i includes q_i levels $\alpha_1, \ldots, \alpha_{q_i}$.
2. Each model set MS_{xy} shown in Figure 1 is extended to the multi-dimensional expression $MS_y(x_1, \ldots, x_p)$, whrer $x_i \in X_i$.
3. Each model RA_{xy} in Figure 2 is extended to the multi-dimensional expression $RA_y(x_1, \ldots, x_p)$.
4. Each model CM_x in Figure 2 is extended to the multi-dimensional expression $CM(x_1, \ldots, x_p)$.
5. Each model IM_{xy} in Figure 2 is extended to the multi-dimensional expression $IM_y(x_1, \ldots, x_p)$.

Each decision path axis X_i can be regarded as a totally ordered set $X_i = \{\alpha_{i1}, \ldots, \alpha_{iq_i}\}$, where

$$\forall j, k(j < k)[\alpha_{ij} \prec \alpha_{ik}]$$

holds. This total order "$\prec$" represents the *hierarchy* or *reporting line* along this decision path. The model sets in the requirement analysis phase can be expressd as

$$MS_R(\vec{x}) = \{RA_R(\vec{x}), RA_O(\vec{x}), RA_F(\vec{x}), RA_A(\vec{x}), RA_P(\vec{x})\}$$

where $\vec{x} = \langle x_1, \ldots, x_p \rangle$, and each $RA_y(\vec{x})$ represents the same objects as defined in Section 3.1.

5 CONCLUSION

In this paper, a formal approach to enterprise modeling and model consistency evaluation was presented. An asymmetric model framework between a business view and a system view was used to create and evaluate enterprise models. This framework is composed of three orthogonal axes that represent different types of concerning points for modeling.

The paper dealt with enterprise modeling within a single enterprise. However, the Internet technologies are enabling enterprise collaborations in a rapid pace. Therefore, it is required to enhance the proposed approach to modeling such enterprise collaborations as Supply Chain Management (SCM), e-business, e-marketplace, or virtual enterprises.

REFERENCES

Aalst, W. M. P. (1998). Three good reasons for using a petri-net-based workflow management system. In *Information and process integration in enterprises: Rethinking documents, pages 161–182, pp.161-182*. Kluwer Academic Publishers.

Aalst, W. M. P. and Hee, K. M. (1995). Framework for business process redesign. In *Proc. of 4th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '95), pp.36-45*. IEEE.

Astesiano, E., Kreowski, H. J., and Bruckner, B. K. (1999). *Algebraic Foundation of System Specification*. Springer.

Doumeingts, G., Chen, D., Vallespir, B., and Fenic, R. (1994). Grai integrated methodology (gim) and its evolutions: a methodology to design and specify advanced manufacturing systems. In *IFIP Transactions B: Computer Applications in Technology B-14 pp.101-117*.

Frankel, D. S. (2003). *Model Driven Architecture: Applying MDA to Enterprise Computing*. John Wiley & Sons.

Jensen, K. (1997). *Coloured Petri Nets Volume 1-3*. Springer.

Kosanke, K. (1992). Cimosa–a european development for enterprise integration part 1: An overview. In *Enterprise Integration Modeling (Edited by C.G. Petrie Jr), pp.179-188*. MIT Press.

Pawlak, Z. (1992). *Rough Sets : Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers.

Scheer, A. W. (1999). *ARIS–Business Process Modeling*. Springer.

Shinkawa, Y. and Matsumoto, M. J. (2000). Knowledge-based software composition using rough set theory. In *IEICE Trans. on Inf. and Syst, Vol.E83-D. No.4,pp.691-700*. IEICE.

Shinkawa, Y. and Matsumoto, M. J. (2001). Identifying the structure of business processes for comprehensive enterprise modeling. In *IEICE Trans. on Inf. and Syst, Vol.E84-D. No.2, pp.239-248*. IEICE.

Vernadat, F. B. (1996). *Enterprise Modeling and Integration*. Chapman and Hall, Upper Saddle River,NJ.

Williams, T. J. and Hong, L. (1998). Pera and geram enterprise reference architectures in enterprise integration. In *Proc. of Design and Information Infrastructure and Systems for Manufacturing, pp3-30*. Kluwer Academic Publishers.