分散型仮想工場の構築*

(第1報, 生産管理法の特性評価)

藤井 進^{*1}, 貝原俊也^{*1}, 森田 浩^{*2}Development of Distributed Virtual Factory
(1st Report, Evaluation of Manufacturing Management Methods)Susumu FUJII^{*3}, Toshiya KAIHARA and Hiroshi MORITA^{*3} Kobe University, Dept. of Computer and Systems Engineering,
1-1 Rokko-dai, Nada-ku, Kobe-shi, Hyogo, 657-8501 Japan

In this paper, Distributed Virtual Factory (DVF) concept is proposed as a distributed simulation environment which is to be used as a tool to evaluate the performance of large and complex manufacturing systems. DVF will be constructed on a distributed computer network system developed for the computer integrated manufacturing. The relation between the structure of the distributed simulation environment and the model of manufacturing system is firstly described and then basic requirements for the virtual factory are presented. Time bucket method is applied as a synchronization mechanism to construct a distributed simulation system and DVF effectiveness is discussed with respect to manufacturing management method analysis by a case study.

Key Words: Distributed Manufacturing Simulation, Synchronization, Virtual Factory, Management

1. はじめに

近年の製造業においては、変種変量生産形態への移行、グローバル環境に対応できるコアコンピタンスの確立など様々な課題が挙げられる。これらの環境下における新しい生産形態として CIM やヴァーチャル・エンタープライズ(VE)⁽¹⁾などが注目されており、現在、その構成法・運用法などに関する研究が進められている⁽²⁾⁽³⁾。このような状況下で生産システムはますます大規模化・複雑化・広域化しており、その設計・運用には多額の資源投入が必要となるため事前の定量的評価が重要となってきた。一般に動的で非線形な外乱項が存在する生産システムの評価法として、従来よりシミュレーションが広く利用され、その有効性が認められている⁽⁴⁾。

しかし CIM 環境下における生産システムでは、複数のサブシステムが物流・情報流の両面で密接に結合されており、サブシステム間で搬送される物やネットワークを通じてやりとりされる情報がシステム全体の挙動に大きな影響を与える。従来のシミュレーションは、工場全体を対象としたシミュレーションモデル作成の困難さからサブシステムを対象として行われることが多く、サブシステムが相互に与える影響を反映しているとはいえない。そのようなローカルなシステムを対象としたシミュレーションでは工場全体規模の評価を行うには不十分であり、大規模工場全体を対象としたシミュレーションシステムの構築が望まれている⁽⁵⁾。

このような要求に対する解決法として、本論文では分散型仮想工場の概念を新たに提案する。これは、従来より各部門の意思決定に利用されてきたシミュレータ群を CIM の基盤となる情報ネットワーク上で統合し、分散シミュレーション技術を用いて工場全体規模のシミュレーションを行うものである。さらに、工場レベルの生産管理システムも統合することにより、生産計画に沿ったより詳細なシミュレーションを行うことも可能となる。その結果、分散型仮想工場の有効性の一つとして、異なる生産管理法を用いた場合の生産システム全体の挙動を定量的に把握することが可能となると考えられる。

そこで本論文では、まず分散型仮想工場の基本的な考え方とその構成法について論じた後、ケーススタディとして実規模レベルの工場モデルを取り上げ、その生産管理法の違いによる生産システム全体の特性

* 原稿受付 2000年3月23日。

^{*1} 正員, 神戸大学工学部(〒657-8501 神戸市灘区六甲台町1-1)。^{*2} 正員, 神戸大学大学院自然科学研究科。

E-mail: fujii@cs.kobe-u.ac.jp

性評価が明確化できることを検証し、分散型仮想工場の有効性について明らかにする。

2. 分散型仮想工場

2・1 基本的な考え方

従来のシミュレーションでは、単一の設備機器や局所的な生産工程を対象としたものがほとんどであり、他のシステムからの影響は確率分布などを用いて簡略化している場合が多い⁶⁾。しかし、CIM 環境下での生産システムにおいては、個々のサブシステムの挙動が相互に関連しており、従来からの手法では CIM 環境下における大規模・複雑な生産システムの十分な評価は困難である。そのため、意思決定に他のエリアで発生する情報を必要とする場合は、各エリアにおける意思決定がエリア相互に与える影響を考慮し、統一的な環境で正確な情報を与えることが望ましい。

これらの課題を解決するため、分散型仮想工場 の概念を提案する。分散型仮想工場とは、CIM 環境下の生産システム全体の評価を対象としたものであり、図1にその概念を示す。

Fig. 1 The concept of Virtual factory

これは、従来各エリアでそれぞれ用いられてきたシミュレータ群を CIM の基盤である情報ネットワーク上で統合し、統合されたシステムに分散シミュレーション技術を導入した分散型シミュレーションシステムである。

このコンセプトにより工場全体を対象とした大規模モデルの構築が比較的容易に行えたとともに、モデルの中に工場全体レベルの生産管理システムを統合することで、工場全体規模の意思決定をシミュレ

ーションに反映することが可能となり、より現実的な仮想工場が実現できる。

2・2 物流モデリング

2・2・1 エリア内の物流モデリング

CIM 環境下での生産システムでは、コンピュータシステムの分散処理環境が整備されており、エリアレベルのシミュレータが活用されている。分散仮想工場では、一から工場全体のシミュレータを作るのではなく、これら既存のシミュレータ群を最小限の改造により統合することで、工場全体を対象としたシミュレーションモデルを比較的容易に構築することを試みる。

まず、各エリアのシミュレータは独立した個々のプロセッサ上に構築されているものとする。またシミュレータ上に構築されるシミュレーションモデルは既存のモデル、新規開発のモデルを問わないものとする。一般的な生産シミュレータにおいて、対象システム外からのワークの搬入は source ノードからの動的要素の出現、対象システム外へのワークの搬出は sink ノードからの消滅としてモデリングされる。

2・2・2 エリア間の物流モデリング

複数エリアのシミュレータ間でのワークの搬送を適切にモデリングするためには、エリア間を結ぶ一つのシミュレーションプロセスとして図2に示すような広域搬送システムのシミュレータを構築すればよい。このようなモデリングにより、複雑な搬送システムによってエリア間搬送を行っている場合においても、搬送の手続きを簡略化せずにシミュレーションモデルを構築することが可能となる。

この広域搬送システムのシミュレーションプロセスを以下では搬送プロセスと呼ぶ。搬送プロセスは既存の搬送システムシミュレータを利用するか、または新規に構築すればよい。なお、搬送プロセスにおいても sink, source の概念はエリアシミュレータと同様である。

このように搬送プロセスを介したエリアシミュレータ間の統合化は図3のように示される。なお、ワーク W_i をエリア i から j に搬送するための手続きは以下のとおりであり、その概念図を図4に示す。

- (1) $TR_i = TR_j$: ワーク W_i は時刻 TS_i にエリア j へ搬送要求を出す。搬送プロセスは搬送要求を同時刻 TR_i に受取る。
- (2) $TS_i = TS_j$: 搬送プロセスは割り当て規則に従い搬送台車を割り当て、割り当てられた台車はエリア i へ移動する。ワーク W_i は到着した台車に搭載され、時刻 $TS_i = TS_j$ にエリア j へ出発する。
- (3) $TA_i = TA_j$: 搬送台車はエリア j に到着し、時刻 $TA_i = TA_j$ にワーク W_i をアンロードする。

- (4) TP_j : エリア j に到着したワーク W_s を工程に投入する準備を行い, 時刻 TP_j に完了する.
ここで, 図4の記号は以下のとおりである.

$$TW_i = TS_{is} - TR_{is} = TS_{is} - TR_{is} \dots\dots\dots (1)$$

$$TT_{ij} = TA_{js} - TS_{is} = TA_{js} - TS_{is} \dots\dots\dots (2)$$

$$TD_j = TP_{js} - TA_{js} \dots\dots\dots (3)$$

Fig. 2 Integration of manufacturing simulators

Fig. 3 Transportation model

Fig. 4 Transportation procedure

2・3 生産システムに対する仮定

まず通常の生産管理システムでは, 複数エリアの生産活動をタイムバケットと呼ばれる一定の時間間隔 BT を単位として管理するのが一般的である⁷⁾.

ここで, 対象とする大規模生産システムのモデルを構築する際, その特徴を考慮するとエリア間搬送について以下のような仮定を設けることができる.

仮定1: 各エリアにはワークの搬入・搬出用バッファがあり, その容量は十分に大きいものとする.

仮定2: タイムバケット期間内に必要なワークはタイムバケット開始時点でエリアに搬入されており, 到着エリアに搬入されたワークがそのタイムバケット期間中に使用されることはないものとする.

仮定3: 仮定2の議論や搬送に要する時間 TT_{ij} が比較的最長いことを考慮すると, あるタイムバケット中に搬送要求がでたワークは, そのタイムバケット中に搬送先で使用されることはないものとする.

実際の生産システムにおいては, 上記の仮定が常に満足されるとは限らない. しかし逆にこれらの仮定を満たすように BT を設定すると考えることにより, 各エリアの挙動は BT ごとに独立と考えられる.

2・4 同期機構

2・4・1 生産システムと同期機構

上述の広域搬送をシミュレーションモデルとして実装する場合, あるシミュレーションモデル上の sink ノードにおいて消滅したワークのデータを移動先のシミュレーションモデルに転送しなければならない. 分散イベントシミュレーションにおいては, このような情報の転送をワークの到着イベントの登録という形で実現する. このとき, 異なるエリアのシミュレータにおけるイベント生起順序の一貫性を保証するための同期機構が必要となる. 同期機構は, 各シミュレータに到着するメッセージの順序関係を制御し, イベントをシミュレーション時刻が非減少になるように実行することを保証する.

同期機構としては, 従来より種々のアルゴリズムが提案され, 代表的なものとしてコンサパティブ法⁸⁾やタイムワープ法⁹⁾などが挙げられる.

しかし, 既存シミュレータの統合化を前提として分散仮想工場の実現を考える場合, これらの手法は既存シミュレータの大幅な内部変更が必要になり現実的であるとは言えない. また, 生産システムに固有な特徴として上述のような仮定を設定することが可能であり, この仮定を最大限に利用した同期機構を提案することは有効であると考えられる.

このような観点から筆者らは, 従来より生産システムの特徴を活かした生産システムシミュレーションの同期機構としてタイムバケット法を提案し, その特徴について多くの報告をしてきた^{(10), (11), (12)}. なお本論文では, 次項にその概念を示す.

2・4・2 タイムバケット法

図5にタイムバケット法の概念を示す。タイムバケット法では、シミュレーション時刻を一定のタイムフェーズに分解し、それをタイムバケットと呼ぶ。このとき、タイムバケット法におけるタイムバケットは生産管理の概念におけるタイムバケットとは必ずしも一致しない。この同期機構では、タイムバケットの終点が同期点となり、各プロセスはその時点でのみメッセージを交換する。

Fig. 5 Basic concept of Time Bucket Mechanism

図5は、搬送プロセスが1タイムバケット分のシミュレーション処理を終了し、一時停止している状態における他のエリアシミュレータのローカルシミュレーションタイム LVT_i を示している。ここで、 TB_k は第 k 番目のタイムバケットを、 STB_k 、 ETB_k はそれぞれ TB_k の開始と終了のシミュレーション時刻を表す($ETB_k = STB_{k+1}$)。タイムバケット法では各シミュレータはそれぞれ独立に1タイムバケット分のシミュレーションを行い、処理を中断する。また、搬送に関するメッセージの交換は、全シミュレータの1タイムバケット分の処理終了を確認後に行われる。

タイムバケット法ではバケットサイズを大きくするほど処理効率が向上する。このためバケットサイズは、2.3 で示した仮定を満たす範囲でできる限り大きくすることが望ましい。なお、エリア間の搬送時間が一定あるいは確率的に決定できると仮定される場合は、図6に示すようにエリア間の搬送を時間遅延のみでモデル化でき、この方法を簡易バケット法と呼ぶ。なお、タイムバケット法の詳細については、文献(10)、(11)、(12)などを参照にされたい。

タイムバケット法を採用した場合、既存のシミュレータにおけるモデル変更は最小限で抑えられる。これは、タイムバケット法が分散型同期法であるにも関わらず、生産システム固有の特徴を利用して、シミュレーション時刻のロールバックやその実現のための過去の内部状態保存といった通常の同期法が必要とする複雑な機構を排除しているからである。

Fig. 6 Simple Time Bucket Mechanism (STBM)

3. ケーススタディ

本章では、本論文で提案する分散型仮想工場の考え方を実規模レベルの工場モデルに適用し、大規模工場を対象にして生産管理法の違いによる生産システム全体の特性評価が明確化できることを検証する。

3・1 対象とする工場モデル

まず生産管理方式として引き取り型生産方式を用いた場合の、対象とする工場モデルの概観およびエリア間の物流・情報流を図7に示す。対象モデルは表1に示すような7エリアからなる大規模工場である。この工場における生産概要は以下の通りである。

生産品目：民生用機械製品5種(A～E)

生産形態：加工はロット生産、組立ては1個流し

稼動形態：加工・組立てとも24時間稼動

Fig. 7 Factory structure (Pull management)

Table 1 Area structure

No.	Area	Function
1	Factory Management	Production Planning Outside Order
2	Storage & Material Handling System	Storage of materials and parts Transportation
3	Processing A	Processing
4	Processing B	Processing, Cleaning, Measure
5	Assembly Line	Assembly, Product storage
6	Distribution Center	Distribution to outside
7	Cooperative Factory	Materials and parts supply

提案法の概念に従い、図7に示される各エリアのシミュレータはそれぞれ異なる7台のプロセッサ上に実装した。また図中で実線は物の流れを、点線は情報の流れをそれぞれ表し、後工程からの要求に従って物が後工程へ送られていることが分かる。

ここで、各プロセッサの仕様と利用したシミュレーション言語について、エリアごとに表2に示す。

Table 2 Configuration

Area No.	EWS	Simulation software
1	Axil: Axil Server 400	C, Microsoft ACCESS 97
2	SUN: Spark Staion 4	Smpl
3	SUN: Spark Staion 4	SLAM II, Fortran, C
4	SUN: Spark Staion 4	SLAM II, Fortran, C
5	SUN: Spark Staion 4	SLAM II, Fortran, C
6	SUN: Spark Staion 4	Smpl, C
7	SUN: Spark Staion 4	Smpl, C

次に、押し出し型生産方式を用いた場合のエリア間の物流・情報流を図8に示す。

Fig. 8 Factory structure (Push management)

押し出し方式では、生産管理コンピュータからの指示に従って物が後工程に送られることが分かる。

本ケーススタディで扱う製品は製品A～製品Eの5種類であり、102の部品から構成される。それぞれの製品の部品構成を表3に示す。部品のうちk01～k20, m01～m30の50種類は内製品として加工工程で処理され、その他は外注とする。

Table 3 Bill of materials

A	B	C	D	E
a10 a20 a30	b10 b20	c10 c20	d10 d20	e10 e20
k01 k19 k06 k20 k11 k16	k02 k19 k07 k20 k12 k16	k03 k17 k08 k18 k13 k19 k16 k20	k04 k18 k09 k19 k14 k20 k17	k05 k19 k10 k20 k15 k18
m01 m26 m06 m29 m11 m30 m16 m19 m20 m21	m02 m22 m07 m26 m12 m27 m16 m29 m17 m30 m19 m20	m03 m20 m08 m23 m13 m26 m16 m27 m17 m28 m18 m29 m19 m30	m04 m24 m09 m27 m14 m28 m17 m29 m18 m30 m19 m20	m05 m28 m10 m29 m15 m30 m18 m19 m20 m25
d32 d08 d33 d09 d01 d10 d02 d11 d03 d12 d04 d05 d07	d30 d19 d31 d13 d14 d15 d16 d17 d18	d28 d22 d29 d23 d16 d24 d17 d25 d18 d26 d19 d27 d20 d21	d32 d33 d01 d02 d03 d04 d05 d06	d30 d13 d31 d14 d07 d15 d08 d16 d09 d17 d10 d18 d11 d19 d12
p01 p04 p02 p05 p03	p01 p04 p02 p05 p03	p01 p04 p02 p05 p03	p01 p04 p02 p05 p03	p01 p04 p02 p05 p03

また、今回の実験で想定した受注パターンを図9に示す。この受注パターンでは、各製品に対して(A:B:C:D:E)=(6:4:3:2:1)の割合となり、一日当たりの受注量も10個/日～50個/日と大きな変動を持つものを想定している。

Fig. 9 Order pattern

3・2 生産管理方式とシステム特性

本節では、大規模工場モデルに分散型仮想工場の考え方を適用することにより、生産管理方式の違いが生産システム全体の特性に与える影響について明確化できることを検証する。

3・2・1 製品在庫量に関する特性評価

まず、対象工場において組立工程の後ろに位置する製品倉庫を対象に、図10(a)は引き取り型生産方式、図10(b)は押し出し型生産方式のそれぞれにおける製品Aの在庫変動を示す。

Fig.10(a) Inventory in product storage by pull management

Fig.10(b) Inventory in product storage by push management

これらの図から、共に4, 7, 10, 17, 20, 23, 30日目の初期段階で製品Aの在庫量が0になっているが、いずれの場合も、その日の後半に製品が組み立てられて製品が補充されていることがわかる。製品倉庫における平均在庫量(図中の点線)は、それぞれ25個と23個であり、押し出し型生産方式は引き取り型生産方式に比べ平均在庫量は92%となっている。このことから、たとえ需要変動が激しく変化する場合であっても、当初の計画に従う受注が行われた場合は、押し出し型生産方式の方が製品在庫量を安定させることができることが分かる。これは、計画主導型の押し出し型生産方式と、基準在庫量遵守を基本とする引き取り型生産方式との特徴による違いと考えられ、分散型仮想工場により工場全体を対象とした両手法の特性差について定量的に把握することが可能となることわかる。

3・2・2 部品在庫量に関する特性評価

次に、対象とする工場のほぼ中央に位置する部品倉庫について在庫量の変動を調べた。なお、対象とする部品は、全ての製品に共通する部品である部品k20を取り上げた。図11(a)は引き取り型生産方式、図11(b)は押し出し型生産方式の結果をそれぞれ示す。

Fig.11(a) Inventory in part storage by pull management

Fig.11(b) Inventory in part storage by push management

ここで、引き取り型生産方式におけるk20の平均在庫量は72個、押し出し型生産方式では220個であった。部品k20に関しては、引き取り型生産方式の方が少ない在庫量で安定した変動を示す。これは、k20の安全在庫量が適切に設定されていることを示し、また製品種類間の需要変動差に対する影響を受けにくい共通部品では、引き取り型生産方式がうまく機能することを示した結果といえる。

3・2・3 機械稼働状況に関する特性評価

表4(a),(b)に、加工エリアAにおける引き取り型生産方式、押し出し型生産方式のそれぞれを用いた場合の機械稼働状況について示す。さらに、表5(a),(b)には、同様に加工エリアBにおけるそれぞれの生産方式についての機械稼働状況を示す。

Table 4(a) Simulation report in Processing A (Pull)

Machine	Utilization	Queue length (Max)	Waiting time
HMC1	0.6229	0.7984 (5)	153.30
HMC2	0.7380	1.3308 (4)	445.66
HMC3	0.5391	0.9914 (4)	357.93
HMC4	0.2869	0.3576 (4)	245.19
VMC1	0.4753	0.0078 (1)	2.61
VMC2	0.1686	0.0000 (1)	0.00

Table 4(b) Simulation report in Processing A (Push)

Machine	Utilization	Queue length (Max)	Waiting time
HMC1	0.8411	0.2033 (4)	111.20
HMC2	0.5856	0.2083 (3)	132.31
HMC3	0.5168	0.2501 (4)	203.87
HMC4	0.2554	0.1733 (3)	170.19
VMC1	0.5886	0.1478 (3)	98.22
VMC2	0.1797	0.1008 (3)	87.12

表4(a)を見ると、引き取り型生産方式を用いた場合のHMC(横型マシニングセンタ)群の平均稼働率は、54.7%、平均仕掛数は0.87個、最大仕掛数は5個となっている。一方、押し出し型生産方式では、それぞれ55.0%、0.21個、4個となっており、引き取り型生産方式よりもやや安定した生産が実現されていることが分かる。

同様に加工エリアB(表5)に着目すると、引き取り型生産方式を用いた場合、NC群の平均稼働率は59.7%、平均仕掛数は0.34個、最大仕掛数は4個である。また、押し出し型生産方式の場合は、それぞれ63.0%、0.16個、8個となっている。この結果、装置稼働率の比較において、同一のシミュレーションラン中でありながら加工エリアAとBはまったく異なった状況であることがわかる。

これらの複雑な結果は、提案法が工場全体を統合化しシミュレーション評価することが可能になったため明らかにすることができた現象である。今回のケースでは、生産方式が異なることにより製品在庫や部品在庫の挙動が相互に影響を与えながら複雑に

変化し、その変化に対応して部品の生産計画がまったく異なるものになった結果、このような差異が発生したと考えられる。

Table 5(a) Simulation report in Processing B (Pull)

Machine	Utilization	Queue length (Max)	Waiting time
NC1	0.7538	0.2615 (3)	73.83
NC2	0.7479	0.2308 (4)	65.17
NC3	0.7200	0.6982 (3)	179.53
NC4	0.7200	0.3641 (4)	108.47
NC5	0.6426	0.3328 (3)	104.17
NC6	0.5620	0.4282 (4)	162.26
NC7	0.3633	0.2634 (4)	104.40
NC8	0.2647	0.1089 (3)	61.89
CLEAN	0.0389	0.0000 (1)	0.00
MESURE	0.3226	0.0241 (3)	1.94

Table 5(b) Simulation report in Processing B (Push)

Machine	Utilization	Queue length (Max)	Waiting time
NC1	0.7560	0.6377 (8)	170.05
NC2	0.7576	0.2581 (7)	88.51
NC3	0.7589	0.1114 (3)	41.85
NC4	0.7951	0.0689 (4)	33.84
NC5	0.6417	0.0882 (4)	49.47
NC6	0.5922	0.0652 (3)	46.20
NC7	0.4743	0.0584 (3)	50.47
NC8	0.2778	0.0221 (4)	79.98
CLEAN	0.0411	0.0000 (1)	0.06
MESURE	0.3563	0.0221 (4)	3.29

3・2・4 実験結果の考察

以上、ケーススタディにより分散型仮想工場の有効性について検討を行った。その結果、提案法により、大規模工場内のいたるところに点在するバッファ内の仕掛状況や各装置の稼働・待ち状況などについて、工場全体を有機的に統合した形で、生産方式の違いによる生産システムの特長評価を定量的に行えることが明らかとなった。

4. 結論

本論文では、まず CIM 環境下でのシミュレーションに対する要件について考察した後、大規模生産システム全体を統合的に評価するための新たなシミュレーション環境として分散型仮想工場の提案を行い、その基本的な考え方と構成法について物の流れを中心に論じた。次に、ケーススタディとして実規模レベルの工場モデルに対して提案法を適用し、引き取り型・押し出し型といった代表的な2つの生産管理法のもとで生産システム全体の特長評価を行っ

た。その結果、大規模システム内に点在するバッファ内の仕掛状況や各装置の稼働・待ち状況などについて、生産方式の違いによる生産システム全体での定量評価が行えることを確認し、提案法の有効性について明らかにした。

このような生産システムの統合的評価は、同期機構を有する分散型稼働工場概念を用いることにより始めて可能になるものであり、提案手法は、CIMやグローバルな VE 環境下におけるシステム評価手法として、今後ますます重要になると考えられる。

文 献

- (1) K. Iwata and S. Fujii, New Manufacturing Era - Adaptation to Environment, Culture, Intelligence and Complexity, Summary of the Questionnaire of the 29th CIRP International Seminar on Manufacturing Systems, 1997.
- (2) H.Afsarmanesh and L.M.Camarinha-Matos, The Virtual Enterprise Concept, Kluwer Academic Publishers, London, 1999.
- (3) T. Kaihara, Supply Chain Management based on Market Mechanism in Virtual Enterprise, Infrastructures for Virtual Enterprises, Kluwer Academic Publishers, Boston, pp.399-408, 1999.
- (4) 藤井 進, 製造システムにおけるシミュレーションとその応用, 情報処理, Vol.33, No.3, pp.240-247, 1992.
- (5) 日比野浩典, 他, 生産システムシミュレータの統合化に関する研究, IMS Projects 2000 (CD-ROM), 2000.
- (6) 森戸晋, 他, システムシミュレーション, 朝倉書店, 2000.
- (7) 太田雅晴, 生産情報システム, 日科技連, 1996.
- (8) R. M. Fujimoto, Parallel discrete event simulation, Communication of the ACM, Vol.33, No.10, pp.30-53, 1990.
- (9) R. M. Fujimoto, Parallel and distributed discrete event simulation: algorithms and applications, Proc. of 1993 WSC, pp.404-425, 1993.
- (10) S. Fujii and M. Hirano, A Basic Study on Distributed Simulation Model of Virtual Manufacturing system under CIM Environment, Proc. of The International Conference on Production Research, pp.281-282, 1993.
- (11) S. Fujii et al, Distributed Simulation Model for Computer Integrated Manufacturing, Proc. of 1994 WSC, pp.946-953, 1994.
- (12) S. Fujii, et al, Synchronization Mechanisms for Integration of Distributed Manufacturing Simulation Systems, International Journal of SIMULATION, SCS, Vol.71, No.3, pp.187-197, 1999.