

Title	分散環境における生産システムのコンカレントな設計 および運用に関する研究
Author(s)	谷水, 義隆
Citation	大阪大学, 2001, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/794
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

大阪大学博士論文

分散環境における生産システムのコンカレントな
設計および運用に関する研究

2001年9月

谷水 義隆

目次

第 1 章	緒論	1
1.1	研究の背景	1
1.2	研究の目的	3
1.3	本論文の構成	3
第 2 章	生産システムのモデルリポジトリ	9
2.1	緒言	9
2.2	生産システムのモデル化における問題点	9
2.2.1	生産システムモデル間の整合性管理	9
2.2.2	生産システムのモデル化に関する既存の研究	10
2.3	モデルリポジトリ	11
2.3.1	モデルリポジトリの定義	11
2.3.2	モデルの分類基準	12
2.4	生産システムモデルの原型の表現	13
2.4.1	単一のカテゴリに属する原型の記述	13
2.4.2	異なるカテゴリに属する原型の関係の記述	14
2.5	生産システムモデル間の整合性の管理	17
2.5.1	整合性の管理手順	17
2.5.2	整合すべきモデルインスタンスの検出	18
2.5.3	整合性管理の一例	19
2.6	結言	21
第 3 章	生産システムの設計プロセスの記述と分析	24
3.1	緒言	24
3.2	MSE プロセス	24
3.2.1	MSE プロセスの定義	24
3.2.2	MSE プロセスの記述に関する既存の研究	25
3.3	MSE プロセスの記述方法	26
3.3.1	設計活動モデル	26
3.3.2	生産システムモデル	27
3.3.3	MSE プロセスの記述ツール	27
3.4	MSE プロセスの分析	29

3.4.1	MSE プロセスの記述対象および記述の手順	29
3.4.2	現状の MSE プロセスの問題点	38
3.5	結言	38
第 4 章	生産システムの設計プロセス支援ツールの開発	41
4.1	緒言	41
4.2	MSE プロセス支援のための要求機能.....	41
4.2.1	既存の研究における MSE プロセスの支援	41
4.2.2	MSE プロセス支援ツールの基本機能.....	42
4.3	MSE プロセス支援ツールの開発.....	42
4.3.1	MSE プロセスの記述.....	42
4.3.2	設計変更の支援.....	45
4.4	ケーススタディ.....	49
4.5	結言	50
第 5 章	生産システムの運用のためのモデル	53
5.1	緒言	53
5.2	階層分散型生産システムのスケジューリングにおける問題点.....	54
5.2.1	生産スケジュールモデルの内容	54
5.2.2	生産スケジュール間の整合性管理	54
5.3	生産スケジュールモデルの構成要素	55
5.3.1	単一の生産スケジュールモデルの表現	55
5.3.2	複数の生産スケジュールモデル間の関係の表現.....	57
5.4	生産スケジュール間の整合性管理の手順.....	59
5.4.1	単一の生産スケジュールの作成.....	59
5.4.2	複数の生産スケジュール間の整合	60
5.5	生産スケジュールのモデリングシステムの開発.....	61
5.5.1	生産スケジュールのモデルリポジトリとモデリングシステム	61
5.5.2	ジョンソン・ルールによる生産スケジューリング技法	63
5.5.3	生産スケジュール間の整合性管理の一例	63
5.6	結言	65
第 6 章	階層分散型生産システムの運用におけるリスケジューリング	67
6.1	緒言	67
6.2	リスケジューリングに関する既存の研究.....	67
6.3	リスケジューリングプロセスの提案.....	68
6.3.1	2つのリスケジューリングプロセス.....	68

6.3.2	リスケジューリングプロセスの比較	71
6.4	リスケジューリングシステムの実装	74
6.4.1	リスケジューリングシステムの構成	74
6.4.2	遺伝的アルゴリズムによる生産スケジューリング技法	75
6.5	ケーススタディ	76
6.5.1	コンカレント・アプローチを用いたリスケジューリングの試行	76
6.5.2	数値実験による 2 つのリスケジューリングプロセスの比較	80
6.6	結言	81
第 7 章	ホロニック生産システムの運用におけるリアルタイムスケジューリング	84
7.1	緒言	84
7.2	HMS の概要	85
7.2.1	HMS の定義	85
7.2.2	HMS の基本構造	86
7.3	HMS における生産スケジューリング	88
7.3.1	リアルタイムスケジューリング	88
7.3.2	生産設備の代替を考慮したリアルタイムスケジューリングプロセスの提案	89
7.4	HMS のシミュレーションのためのモデルリポジトリ	91
7.4.1	HMS モデルの構成要素	91
7.4.2	HMS モデルによるリアルタイムスケジューリングプロセス	93
7.5	シミュレーションシステムの実装	94
7.5.1	シミュレーションシステムの基本構成	94
7.5.2	シミュレーションにおける時刻の進行方法	95
7.6	ケーススタディ	97
7.6.1	シミュレーションの対象とする生産システム	97
7.6.2	適用するディスパッチングルール	98
7.6.3	リアルタイムスケジューリングの試行	99
7.7	結言	103
第 8 章	結論と展望	106
8.1	結論	106
8.2	展望	109
謝辞	111
本論文の基礎となる発表論文	113

目次

Fig. 1.1 Relations among chapters of doctoral thesis	4
Fig. 2.1 Generation and utilization process of manufacturing system model	11
Fig. 2.2 Category of model repository	13
Fig. 2.3 Generation of application model	14
Fig. 2.4 Generation of models on different categories	17
Fig. 2.5 Class objects and instance objects.....	20
Fig. 2.6 Results of detected model instances to be consistent.....	22
Fig. 3.1 Manufacturing system engineering process	25
Fig. 3.2 Example of engineering activity sheet.....	28
Fig. 3.3 Example of IDEF0 notation.....	29
Fig. 3.4 Target manufacturing system for test case	30
Fig. 3.5 MSE process for machining line (1)	32
Fig. 3.6 MSE process for machining line (2)	32
Fig. 3.7 MSE process for machining line (3)	33
Fig. 3.8 MSE process for manual assembly line (1).....	33
Fig. 3.9 MSE process for manual assembly line (2).....	34
Fig. 3.10 MSE process for automated assembly line (1).....	34
Fig. 3.11 MSE process for automated assembly line (2).....	35
Fig. 3.12 MSE process for transportation system and line layout design (1)	35
Fig. 3.13 MSE process for transportation system and line layout design (2)	36
Fig. 3.14 MSE process for transportation system and line layout design (3)	36
Fig. 3.15 Example of information exchange on MSE process of test case.....	37
Fig. 4.1 Interface to describe engineering activities.....	43
Fig. 4.2 Interface to search output data	44
Fig. 4.3 Interface to describe output data	44
Fig. 4.4 Example of MSE processes	46
Fig. 4.5 Effects of designed data	48
Fig. 4.6 MSE process for case study	49
Fig. 4.7 Output data affected by re-design.....	51
Fig. 5.1 Schedules of hierarchical and decentralized manufacturing system.....	55
Fig. 5.2 Parallel and hierarchical production planning	58
Fig. 5.3 Consistency among production plans.....	61

Fig. 5.4 Model repository for production scheduling	62
Fig. 5.5 Example of production schedule models	64
Fig. 5.6 Consistency management of schedules	65
Fig. 6.1 Schedules on different layers.....	69
Fig. 6.2 Rescheduling process	70
Fig. 6.3 Hierarchy of schedules	71
Fig. 6.4 Sequential and concurrent approaches for rescheduling.....	72
Fig. 6.5 Target manufacturing system for case study	77
Fig. 6.6 Rescheduling process of target manufacturing system	79
Fig. 6.7 Results of case studies	81
Fig. 7.1 Holarchy	85
Fig. 7.2 Basic structure of holonic manufacturing system.....	87
Fig. 7.3 Holarchy of HMS	88
Fig. 7.4 Real-time scheduling process of HMS.....	90
Fig. 7.5 Model repository of HMS	92
Fig. 7.6 Real-time scheduling process of HMS models.....	93
Fig. 7.7 Basic structure of object oriented simulation system of HMS	95
Fig. 7.8 3-dimensional animation interface	96
Fig. 7.9 Holonic manufacturing system for simulation.....	97
Fig. 7.10 Simulation result of real-time scheduling of case 1	100
Fig. 7.11 Simulation result of real-time scheduling of case 5	101
Fig. 7.12 Simulation result of real-time scheduling of case 6	102
Fig. 7.13 Simulation result of real-time scheduling of case 7	102

表目次

Table 1.1 Characteristics of the research	5
Table 2.1 Relationship on different aspect	20
Table 2.2 Relationship on different abstraction.....	20
Table 2.3 Relationship on different scope.....	21
Table 2.4 Model instances	21
Table 4.1 Initial data for case study	49
Table 6.1 Production process for layer (1)	77
Table 6.2 Production process for layer (2)	77
Table 6.3 Production process for layer (3)	78
Table 6.4 Sequence of jobs for layer (1)	78
Table 6.5 Sequence of jobs for layer (2)	78
Table 6.6 Sequence of jobs for layer (3)	78
Table 6.7 Sequence of jobs for layer (1) after rescheduling process.....	80
Table 6.8 Sequence of jobs for layer (2) after rescheduling process.....	80
Table 6.9 Sequence of jobs for layer (3) after rescheduling process.....	80
Table 7.1 Experimental condition for simulation	99
Table 7.2 Total processing time at case studies	99
Table 7.3 Experimental condition of simulation for failure	101
Table 7.4 Total processing time at case studies for failure	101

第 1 章 緒論

1.1 研究の背景

消費者のニーズの多様化により、製品のライフサイクルは短縮化している。そのため、製造企業には、消費者の要求を満たす製品を素早く製造し、迅速に市場へ供給することが求められている。

これまでは、CAD(Computer Aided Design)や NC(Numerical Control) 工作機械といった対象製品の設計や生産プロセスの能力をコンピュータの支援により向上させることで、その要求を満たそうとしてきた。しかし、設計および生産プロセスの生産性向上だけではなく、製造企業の全ての活動を総合的に考え、ひとつの統合化された生産システムとして製造企業の能力を高めることが重要である。そのためには、対象製品の設計・生産プロセスの合理化だけでなく、生産システムの計画、設計、運用および制御のプロセスの合理化が必要になる。

コンピュータを用いて生産システムの計画、設計、運用および制御を行う場合には、対象となる生産システムのモデルおよびモデル化のためのアーキテクチャが必要となる。生産システムのモデルおよびモデル化のためのアーキテクチャは、製造企業を取り巻く環境の変化や、コンピュータのソフトウェアおよびハードウェア技術の進歩により変化してきた。

生産システムのモデル化は、1970 年代後半の ICAM(Integrated Computer Aided Manufacturing)プログラムで初めて行われた。ICAM では、IDEF(ICAM Definition)という図形的システム定義手法を用いて、航空機の製造活動における機能とその相互関係が記述された。ここで開発された機能モデル記述手法の IDEF0 [FIPS 93] は、現在でも広く利用されている。

1980 年代中頃には、コンピュータの支援により、生産システムの計画、設計、運用および制御を統合化する CIM(Computer Integrated Manufacturing)の概念が提案された [岩田 90] [人見 84]。日本工業規格 [JIS 90] では、CIM とは、生産に関するすべての情報をコンピュータネットワーク及びデータベースを用いて統括的に制御・管理することによって、生産活動の最適化を図るシステムと定義している。一般に、CIM とは、大型コンピュータによる大規模なデータベースを用いて、生産に関する全ての情報を集中的に管理し、生産現場の自動化設備をトップダウン的に制御する生産システムを表していた [福田 98]。製造企業では、独自のアーキテクチャに基づき、CIM システムを構築していった [岩田 92]。そこで、CIM のアーキテクチャを統一するために、各国の研究グループにより、標準的なモデルやアーキテクチャが提案された [FA 委員会 93] [Spur 96] [Williams 94a]。ISO(International Standardization Organization)は、生産システムにおける階層的制御構造をモデル化した 6 階層モデル [ISO 89] を提案した。これは、生産制御のための参照モデルであり、生産システムの具体的なモデリング手法を与えるものではなかった。その後、欧州の ESPRIT(European Strategic Program for Research and Development in Information Technology)プロジェクトでは、CIMOSA(CIM

Open System Architecture) [AMICE 92] [Didic 94] [大見 93] や GIM(GRAI Integrated Methodology) [Doumeingts 93] が, 米国では, PERA (Purdue Enterprise Reference Architecture) [Williams 93] [Williams 94b]が提案された. これらはいずれも, 生産システムの分析, 計画から制御プロセスまでの支援を目的とするライフサイクル指向のモデリング手法であった. すなわち, ビジネスプロセスを分析し, 生産システムの設計, 運用および制御を行うためのモデルを構築するだけでなく, そのモデルを利用して実生産システムの制御プロセスまでの支援を行おうとするものであった.

CIM は, 1980 年代までの少品種多量生産における効率性を重視した生産形態には適していた. しかし, 製造企業を取り巻く状況の変化により, 集中管理型の大規模な CIM は適さなくなってきた [福田 99]. 具体的には以下のような問題があげられる [瀬領 92] [FA Report 94].

- 生産形態が, 少品種多量生産から多品種少量生産へと移行してきた. そのため, 変化に対する柔軟性が乏しい集中管理型 CIM では, 対応が困難になってきた.
- 生産活動のグローバル化により, 複数の企業間における情報授受の必要性が高まっている. しかし, 構築された集中管理型 CIM の情報システムが企業毎に異なるため, 企業間で情報を授受することは容易でない.
- 景気の低迷により, 多額の資金を必要とする大規模な集中管理型 CIM の構築や運用は困難になってきた.

一方, 1990 年代中頃から, 処理能力が高く安価なパーソナルコンピュータが製造企業のあらゆる部門に普及するとともに, インターネットや TCP/IP (Transmission Control Protocol/ Internet Protocol) に代表される標準的なネットワーク技術が利用されるようになってきた [稲本 91]. これに伴い, 新たな CIM のアーキテクチャとして, オブジェクト指向技術を用いて, 分散環境のコンピュータで部門毎に開発したアプリケーションプログラムを, ネットワークを介して結合することにより, 種々の活動を統合化する手法が提案されている [福田 97]. 例えば, 米国の SEMATECH は, オブジェクト指向プログラミング手法で個々に作成したアプリケーションプログラムを, CORBA (Common Object Request Broker Architecture) [Siegel 96] [Ryan 96] を用いた共通のプラットフォーム上で結合することで, 各部門のアプリケーションを統合化する CIM アプリケーションフレームワーク [SEMATEC 95] を提案した. 集中管理を特徴とした 1980 年代の CIM に対して, これは分散的な CIM [福田 98] といえる. オブジェクト指向技術を用いる分散的な CIM では, 各アプリケーションプログラムがモジュール化されているため, 規模を大きくすることなく情報システムの開発が可能であるだけでなく, 必要に応じて情報システムやプログラムを部分的に修正することで柔軟な対応が可能である. また, CORBA など共通のアーキテクチャを用いることで, 異なる企業間での情報授受も比較的容易に行える. 以上のような特徴から, IFIP (International Federation of Information Processing) の GERAM (Generalized Enterprise Reference Architecture and Methodology) [GERAM 98] など, 企業モデルの標準化を検討している各機関においても, オブジェクト指向モデリング手法の適用が注目されている.

1.2 研究の目的

オブジェクト指向技術を用いる分散的な CIM のモデルおよびそのモデル化のためのアーキテクチャに関する研究は、まだ多くは行われておらず、十分ではない。そのため、製造企業のエンジニアが、実際に生産システムをモデル化するのは容易ではない。また、モデルの利用手法に関する提案は、ほとんど行われておらず、モデル化により得られる効果や利点が明確に示されていない。

集中管理を特徴とした 1980 年代の CIM では、生産システムの計画、設計から、運用および制御における種々の活動は、その階層構造にしたがってある手順で行わなければならなかった [CIM 委員会 91] [金子 92] [長江 93]。そのため、生産システムを計画してから、実際に運用されるまでに長い期間を要するという問題があった。一方、オブジェクト指向技術を用いる分散的な CIM では、種々の活動を分散環境で自律的に実行することができるため、逐次行っていた諸活動をコンカレントに行うことが可能になる。これにより、生産システムの計画から制御までに要する時間を短縮することができるだけでなく、変更に対して柔軟な対応が可能になると考える。しかし、生産システムの計画から制御における複数のアプリケーションをコンカレントに行う方法について提案した研究はない。

そこで、本論文では、分散環境において、生産システムの設計から運用までの過程における複数のアプリケーションをコンカレントに実行するために、オブジェクト指向技術を用いた生産システムの系統的なモデル化のアーキテクチャを提案するとともに、そのモデルの利用手法を提案し、コンカレントな分散処理についての効果を示す。

1.3 本論文の構成

本論文は Fig. 1.1 に示すように、本章を含めて、8 章で構成されている。本論文で提案する内容は、以下の 3 つに分類できる。

- (1) 生産システムのモデル化。
- (2) 生産システムの設計フェーズにおける生産システムのモデルとその利用手法。
- (3) 生産システムの運用フェーズにおける生産システムのモデルとその利用手法。

各章における概要を以下に示す。

まず、第 2 章では、生産システムのモデルの作成と利用の手法を提案するとともに、オブジェクト指向の考え方に基づき、表現形式の異なる生産システムのモデル間の整合性を管理する方法を提案する。

第 3 章および第 4 章では、生産システムの設計フェーズにおける生産システムのモデルとその利用手法について考察する。

第 3 章では、第 2 章で提案した生産システムのモデルを拡張して、生産システムの設計プロセスのモデル化手法を提案し、設計プロセスの記述ツールを提案する。次に、生産システムの設計実験を行い、その設計プロセスを記述し、分析して、生産システムの設計プロセスの問題点を明らかにする。

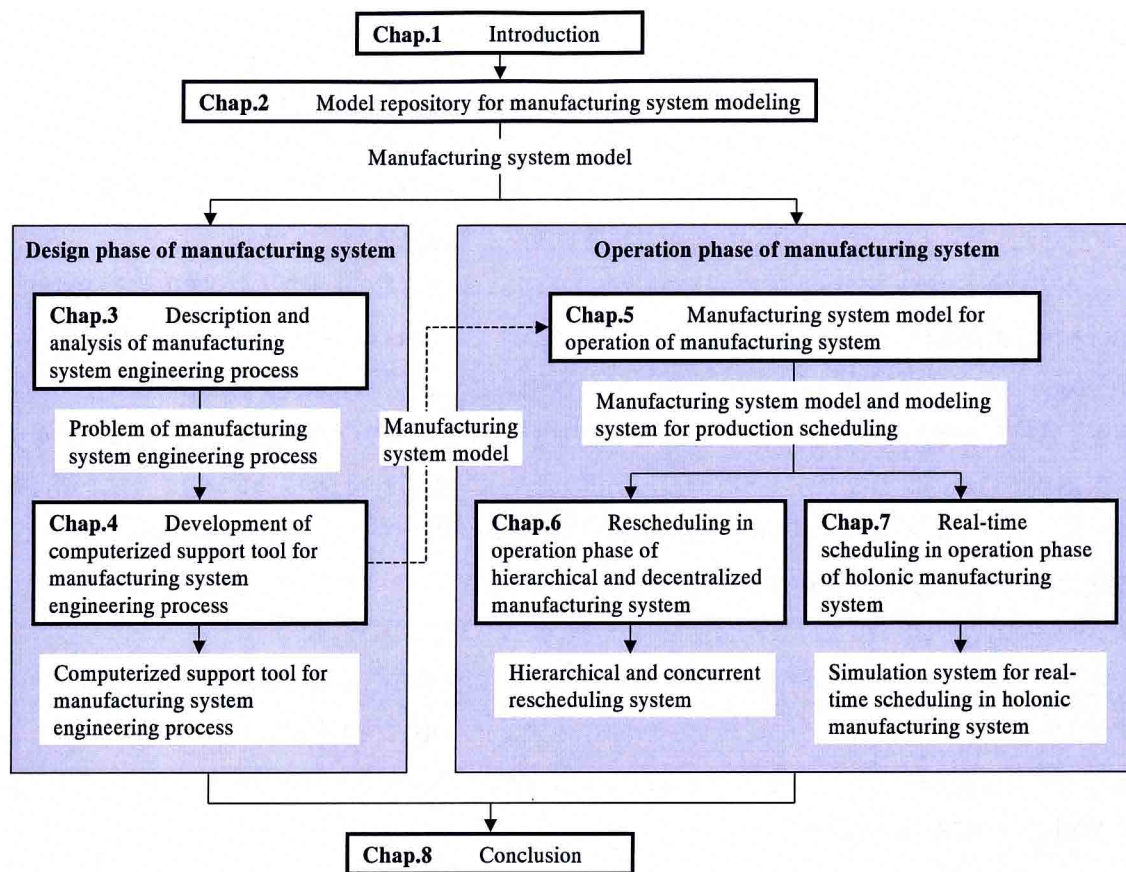


Fig. 1.1 Relations among chapters of doctoral thesis

第4章では、第3章で明らかにした問題点に基づき、設計プロセス支援ツールのプロトタイプを開発するとともに、それを簡単な生産システムの設計例に適用して、有効性を検証する。

第5章から第7章までは、生産システムの運用フェーズにおける生産システムのモデルとその利用手法について考察する。本論文では、領域を生産スケジューリングに絞って考察する。

第5章では、第2章で提案した生産システムのモデルの表現方法に基づき、生産スケジューリングのためのモデルの基本的な構成要素を提案する。次に、そのモデルを用いて、工場の生産スケジュールや、セルの生産スケジュールなど、階層的な複数の生産スケジュール間の整合性を管理するための方法を提案する。

第6章では、第5章で提案した生産スケジューリングのためのモデルの構成要素を用いて、階層的な複数の生産スケジューリングシステムで、コンカレントに生産スケジュールを変更(リスケジューリング)する方法を提案する。さらに、リスケジューリングシステムのプロトタイプを開発し、提案するリスケジューリング手法の有効性を検証する。

第5章および第6章では、各階層において、単一の生産スケジューリングシステムが、階層内の構成要素の生産スケジュールを決定している。しかし、近年、生産システム内の各構成要素が自身の意思決定を行う自律分散型生産システムが提案されている。そこで、第7章では、自律分散型生産システムの

ひとつであるホロニック生産システム(Holonic manufacturing system)を研究対象として、その生産スケジューリングについて考察する。まず、ホロニック生産システムにおけるリアルタイムスケジューリング手法を提案する。次に、第5章で提案した生産スケジューリングのためのモデルの構成要素を拡張して、ホロニック生産システムのシミュレーションシステムのプロトタイプを開発し、簡単な例を用いて、ホロニック生産システムのリアルタイムスケジューリングの特長について考察する。

第8章では、これらの結論について総括を行うとともに、今後の展望について考察する。

本研究は、生産システムモデルを核として、生産システム的设计および運用における諸活動の統合化の可能性を示すとともに、統合化による効果を具体的に示した点に新規性があると考えられる。従来の研究と比較した場合の本研究の特徴をTable 1.1にまとめる。

Table 1.1 Characteristics of the research

	既存の研究の問題点	本研究の特徴
第2章	<p>生産システム的设计から運用における複数のアプリケーションは、集中的に管理され、トップダウンで制御される。</p> <p>そのため、设计から製造までに時間がかかるだけでなく、変更に対する柔軟性が乏しい。また、新たなアプリケーションの追加や部分的な修正が困難である。</p>	<p>表現形式の異なる種々の生産システムモデルを整理する生産システムのモデルリポジトリを開発し、モデル間の整合性管理を支援するプロトタイプシステムを開発した。</p> <p>これにより、諸活動をコンカレントに行うことができ、设计から製造までの時間の短縮化や、変更に対する迅速で柔軟な対応が可能となる。また、新たなアプリケーションの追加や部分的な修正が容易になる。</p>
第3章	<p>生産システム的设计プロセスに関する研究は、あまり行われておらず、その特徴や問題点は明らかにされていない。</p> <p>そのため、製造企業で利用できる设计プロセス支援ツールを開発することは困難である。</p>	<p>製造企業で生産システム的设计に関与している技術者と、生産システム的设计を実験的に行い、その设计プロセスを記述、分析した。</p> <p>これにより、设计プロセスの特徴のひとつとして、设计活動の繰り返し処理を示した。また、设计期間の短縮化のための重要な問題として、设计情報の授受、设计変更、および事例の蓄積と利用の3つの問題を明らかにした。</p>
第4章	<p>设计プロセスを記述するためのツールはこれまでにいくつか存在するが、それを用いて、设计プロセスを十分に支援することはできない。</p> <p>そのため、生産システム的设计期間を短縮することは容易でない。</p>	<p>设计期間の短縮に重要な機能のうち、设计情報の授受の支援と、设计変更の支援を行うことができる设计プロセス支援ツールを開発した。</p> <p>これを用いて、複数の設計者間で设计情報を正確かつ迅速に授受することで、设计期間を短縮することができる。また、设计変更が連鎖的に及ぼす影響を迅速に求めることで、设计変更による设计期間の長期化を抑制できる。</p>

第 5 章	<p>生産スケジュールは、生産管理者により、階層ごとに作成、管理されている。</p> <p>そのため、ある生産スケジュールに変更が生じた場合、全ての階層において生産スケジュール間の整合性管理を迅速に行うことは容易でない。</p>	<p>生産スケジュールリングのためのモデルリポジトリとモデリングシステムのプロトタイプを開発した。</p> <p>これにより、階層的な生産スケジュール間の整合性管理が容易に行え、生産スケジュールの変更に対応できる。</p>
第 6 章	<p>階層的な生産スケジュールの作成後に、工作機械の故障など、生産現場の状況の変化が予測できた場合、全体の納期を満たすように、生産スケジュールを迅速に変更(リスケジューリング)しなければならない。</p> <p>従来の生産管理手法では、下位階層のリスケジューリング後に、上位階層のリスケジューリングを開始するため、リスケジューリングを完了するのに時間がかかるという問題がある。</p>	<p>第 5 章で開発したモデリングシステムの機能を拡張して、全ての階層で同時かつ並列に、リスケジューリングを行う、リスケジューリングシステムのプロトタイプを開発した。</p> <p>これにより、生産開始前に予測した生産現場の状況の変化に対し、階層的な複数の生産スケジュール間の整合性を保持しながら、迅速にリスケジューリングを行うことができる。</p>
第 7 章	<p>事前に適切な生産スケジュールを作成する従来の生産管理手法では、生産開始後に起こった生産現場の状況の変化に柔軟に対応することが困難であるという問題がある。</p>	<p>新たな生産システムのひとつとして提案されているホロニック生産システムについて、そのリアルタイムスケジュールリングによるシミュレーションシステムのプロトタイプを開発した。</p> <p>これにより、生産開始後に起こった生産現場の状況の変化に柔軟に対応して生産管理を行うことができる。</p>

参考文献

- [AMICE 92] ESPRIT consortium AMICE, CIMOSA AD 2.0, *ESPRIT Project 5288*, (1992).
- [CIM 委員会 91] CIM デザイン研究委員会, CIM 構築ガイドブック, 工業調査会 (1992), 159-203.
- [Didic 94] Didic M., CIMOSA model creation and execution for a casting process and a manufacturing cell, *Computers in industry*, 24 (1994), 237-247.
- [Doumeingts 93] Doumeingts G., ほか 2 名, GIM: GRAI integrated methodology, *Proc. of DIISM'93*, JSPE-IFIP WG 5.3 workshop, (1993), 453-464.
- [FA Report 94] FA Report 編集部, 特集「いま求められる CIM」, FA Report, 8-2, エンジニアリング・ジャーナル社 (1994), 6-20.
- [FA 委員会 93] 統合化 FA 調査研究委員会, 統合化 FA の標準化に関する調査研究 成果報告書, 国際ロボット・FA 技術センター (1993), 9-20.

- [FIPS 93] Announcing the standard for integration definition for function modeling (IDEF0), Federal Information Processing Standards Publications 183, <http://www.iedf.com/Downloads/pdf/idef0.pdf> (1993).
- [GERAM 98] IFIP-IFAC task force, GERAM: Generalized enterprise reference architecture and methodology, version 1.6.2 (1998).
- [ISO 89] ISO/TC184, Reference model for shop floor production standards, *ISO technical report 10314*, (1989).
- [JIS 90] 日本工業標準調査会, 日本工業規格 FA 用語, JIS B 3000, 日本規格協会 (1990), 2.
- [Ryan 96] Ryan, T. W., *Distributed object technology – concept & applications*, Prentice Hall (1996); 植野・多田 訳, 分散オブジェクトテクノロジー 概念とアプリケーション, プレンティスホール (1997).
- [SEMATEC 95] Freed, K., Implementation handbook for the computer integrated manufacturing (CIM) application framework specification 1.2, Technical transfer #95092971A-ENG, SEMATEC (1995).
- [Siegel 96] Siegel, J., *CORBA Fundamental and Programming*, Wiley Computer Publishing Group (1996).
- [Spur 96] Spur, G., Mertins, K. and Jochem, R., *Integrated enterprise modelling*, Beuth verlag gmbh (1996).
- [Williams 93] Williams T.J., The purdue enterprise reference architecture, *Proc. of DIISM'93*, JSPE-IFIP WG 5.3 Workshop, (1993), 317-337.
- [Williams 94a] Williams T.J., et al., Architecture for integrating manufacturing activities and enterprise, *Computers in industry*, 24 (1994), 111-139.
- [Williams 94b] Williams T.J., The purdue enterprise reference architecture, *Computers in industry*, 24 (1994), 141-158.
- [稲本 91] 稲本, CIM におけるこれからの制御技術, 計測と制御, 31-1 (1992), 162-165.
- [岩田 90] 岩田, 新しい生産システム—CIM, システム/制御/情報, 34-3 (1990), 123-127.
- [岩田 92] 岩田, ほか 11 名, 実践 CIM 構築法 ~導入への手引き~, 大阪科学技術センター (1992).

- [大見 93] 大見・福田, CIMのためのオープンシステムアーキテクチャ ESPRIT 研究報告・プロジェクト 688・AMICE・第1巻, 日刊工業新聞社 (1993).
- [金子 92] 金子・山下, CIMの実践とCIMインテグレータの役割, ファクトリ・オートメーション, 4 (1992), 59-65.
- [瀬領 92] 瀬領・渡部, CIMアーキテクチャ -CIMを支える情報技術体系-, 情報処理, 33-3 (1992), 213-221.
- [長江 93] 長江, ほか5名, オープンシステム指向のCIM構築法, 大阪科学技術センター (1993).
- [人見 84] 人見・中島・吉村・吉田, CIM総論-コンピュータによる設計・生産・管理, 共立出版 (1984).
- [福田 97] 福田, 生産システムのオープン化の動向, 精密工学会誌, 63-5 (1997), 613-616.
- [福田 98] 福田, エンタープライズモデル, 人工知能学会誌, 13-6 (1998), 3-9.
- [福田 99] 福田, 生産システムの軌跡 多量生産システムから高付加価値生産システムへ, 精密工学会誌, 65-1 (1999), 13-18.

第 2 章 生産システムのモデルリポジトリ

2.1 緒言

生産システムの計画, 設計, 実装, 運用および制御にいたるライフサイクルにおいて, コンピュータを用いた支援が行われるようになってきた. コンピュータを用いて, 生産システムを系統的に評価するためには, 評価の対象となる生産システムのモデル化が必要となる. 本論文では, 生産システムおよびその構成要素をモデル化したものを生産システムモデル(Manufacturing system model)と呼ぶことにする.

生産システムのライフサイクルを支援する場合, 表現形式の異なる多数のモデルが存在することになる. この時, これらの表現形式の異なる生産システムモデルの間の整合性を保証することが重要となる. 複数の生産システムモデル間の整合性を管理するには, ある一定の手順にしたがって生産システムモデルの作成と利用を行った上で, 生産システムモデル間の整合性管理の手法を適用することが必要と考える.

そこで, 本章では, まず, 生産システムモデルの作成と利用の手法を提案する. 次に, オブジェクト指向の考え方に基づいて, 表現形式の異なる生産システムモデルを構造化する方法を提案し, 構造化した生産システムモデル間の関係にしたがって, 生産システムモデル間の整合性を管理する方法を提案する. さらに, 生産システムモデル間の整合性管理のためのプロトタイプシステムを開発し, 簡単な例に適用する.

2.2 生産システムのモデル化における問題点

2.2.1 生産システムモデル間の整合性管理

生産システムのライフサイクルでは, 部門の異なる複数の設計者や生産管理者等により, 分散環境において複数の生産システムモデルが構築され, 利用される. 例えば, 工場内のレイアウト設計のためのモデル, ライン内の設備配置のためのモデル, ショップやセルの生産スケジューリングのためのモデルなどが異なる部門において構築され, 利用される. これら複数のアプリケーションをコンカレントに行うためには, 部門間での意思決定の調節に必要な情報を迅速に授受し, 生産システムモデル間の整合性を管理できることが必要である.

ネットワーク技術の普及により, 分散環境において情報を迅速に伝達することは容易になっている. また, 異なるアプリケーションソフトウェア間においても, 直接, 情報を授受する機能が準備されるようになってきた. しかし, そのソフトウェア上で記述される生産システムモデルは, 設計者や生産管理者等のモ

デルの利用者が任意に作成するため、その利用目的の違いにより、モデルの表現形式は異なり、構造化されていない。そのため、部門間で情報を自由に交換できないという *islands of information* の問題 [AMICE 92] が存在し、部門間での意思決定の調節に必要な情報の授受および生産システムモデル間の整合性管理は、多くの場合、人間を介して行われている。

必要な情報を迅速に授受し、生産システムモデル間の整合性を管理するためには、生産システムモデルの構造化の枠組みを構築し、それにしたがって生産システムモデルの作成と利用を行った上で、生産システムモデル間の整合性管理の手法を適用することが必要である。

2.2.2 生産システムのモデル化に関する既存の研究

第 1 章で示したように、ライフサイクル指向の生産システムのモデル化に関する研究として、CIMOSA [AMICE 92] [大見 93] や、GIM [Doumeingts 93] , PERA [Williams 93] などの研究がある。

CIMOSA は、欧州の ESPRIT (European Strategic Program for Research and Development in Information Technology) プロジェクトにおいて 1985 年から行われた、CIM の開放型システムアーキテクチャの開発に関する研究である。CIMOSA においては、生産システムモデルを構造化するための枠組みとして、生産システムモデルの一般性 (Genericity)、モデル化 (Models)、およびビュー (Views) の次元 (Dimensions) からなるモデリングフレームワーク (Modeling framework) が提案された。一般性の次元では、生産システムモデルの特殊性を表す *Generic, Partial, Particular* の 3 つのレベルが提案されている。モデル化の次元では、生産システムの要求仕様から実装までの過程を表す *Requirement definition, Design specification, Implementation description* の 3 段階が提案されている。また、ビューの次元では、生産システムをモデル化する視点として、*Function view, Information view, Resource view, Organisation view* の 4 つが提案されている。

GIM は、生産管理を対象とした GRAI Conceptual Model [Doumeingts 88] を、生産システム全体に拡張した CIM の設計法である。ここでも、CIMOSA と同様のモデリングフレームワークが提案されている。

PERA では、製造企業以外も含め、あらゆる企業のシステムを対象として、その目標設定から設計、構築およびそのメンテナンスまでのライフサイクル活動を支援する手順を提案した。ここでは、ライフサイクルを 7 つのレイヤ (Layer)、5 つのフェーズ (Phase)、および 2 つのビュー (View) に分けて、各段階で行うべき活動が提案されている。

これらの研究では、モデル化の視点が複数、考慮される点は優れていると考える。しかし、集中管理型の大規模な CIM を前提としているため、生産システムのライフサイクルにおける種々の活動は、その階層構造にしたがってある手順で行わなければならない。そのため、これらの手法に基づき、生産システムのライフサイクルにおける複数のアプリケーションを、分散環境でコンカレントに行うことは困難であると考えられる。

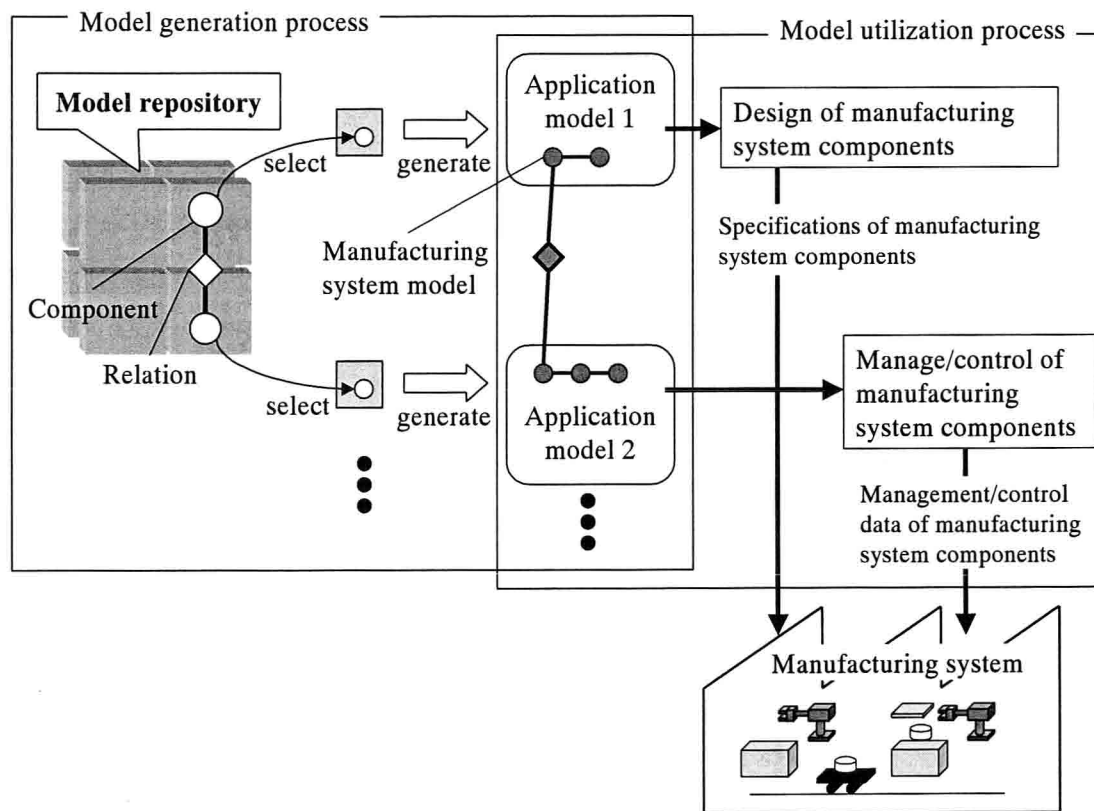


Fig. 2.1 Generation and utilization process of manufacturing system model

2.3 モデルリポジトリ

2.3.1 モデルリポジトリの定義

生産システムモデルの作成と利用のプロセスを、Fig. 2.1に示すように考える。すなわち、生産システムモデルは目的毎に構築するが、モデルの生成に用いる規範はあらかじめ定められている。ここで規範とは、モデルの構成要素、構成要素の属性、構成要素間の関係と制約など、生産システムモデルを生成する際に使用するモデルの構成要素とその関係を定義するものである。この規範は、モデルを使用する目的により異なるため、複数の規範が必要になる。ここでは、モデルの分類基準にしたがって規範を分類、整理したものをモデルアーキテクチャ(Model architecture)、このモデルアーキテクチャに基づいて作成されたモデルの構成要素の原型を集めたものをモデルリポジトリ(Model repository)と呼ぶ。ここで、原型とは、オブジェクト指向モデル(Object-oriented model)におけるクラス(Class)に相当するもので、あるモデルが持つべき特性を定義したものであり、これに属性値を与えることで特定の対象を表す生産システムモデルを構築できると考える。すなわち、対象のモデルの原型は、対象の名称、および対象の持つ属性と属性値の対を用いて表せる。それらを順に記号 X , p_i , a_i で表すと、以下のよう

になる。

$$(X \quad (p_1 \ a_1) \ (p_2 \ a_2) \ \dots \ (p_n \ a_n)) \quad (2.1)$$

Fig. 2.1に示すように、生産システムモデルの設計者は、モデルリポジトリの中から選択した生産システムモデルの原型に属性値を与えて生産システムモデル作成し、それを組み合わせることで、レイアウト設計や生産スケジューリングなど、ある特定の目的に適合する生産システムモデルの組み合わせを構築する。このようにして組み合わされた生産システムモデルの集まりをアプリケーションモデル (Application model)と呼ぶ。

2.3.2 モデルの分類基準

Fig. 2.1に示す生産システムモデルの生成および利用の手順を実現するには、生産システムモデルの構成要素をある基準で分類し、構成要素の原型をあらかじめ作成しておく必要がある。本章では、生産システムおよびその構成要素のモデルを以下の3つの基準で分類し、整理する。

(1) 視点

生産システムおよびその構成要素をモデル化する際に、着目すべき特性を表す。例えば、生産システムの生産機能に着目すると、加工設備、組立設備などが構成要素となり、生産制御機能に着目すると、生産設備のNC制御装置、セルコントローラなどが構成要素となる。この場合、ひとつのNC工作機械は、加工設備およびNC制御の視点からモデル化される。

(2) 抽象性

生産システムおよびその構成要素をモデル化する際に、どの程度詳細な属性を含めるかを表す。例えば、システム設計の初期段階では抽象的な加工設備の属性を持つモデルを作成し、詳細設計の段階ではマシニングセンタ (Machining center)、ターニングセンタ (Turning center) などの具体的な設備の属性を持つモデルを作成する。この場合、抽象的なモデルの属性は、具体的なモデルの属性に継承される。

(3) 範囲

モデルが考慮する空間的、時間的広がりを表す。例えば、工場全体の生産スケジューリングのためのモデルは工場全体と長期のタイムスパンを取り扱い、ショップフロアの生産スケジューリングのためのモデルは工場の一部であるショップフロアと短期のタイムスパンを取り扱う。

ここでは、Fig. 2.2に示すように、これらの3つの分類基準を座標軸と考え、これらの軸で表現される空間をカテゴリと呼ぶ。生産システムの構成要素をモデル化する際には、モデル化の対象が同一であ

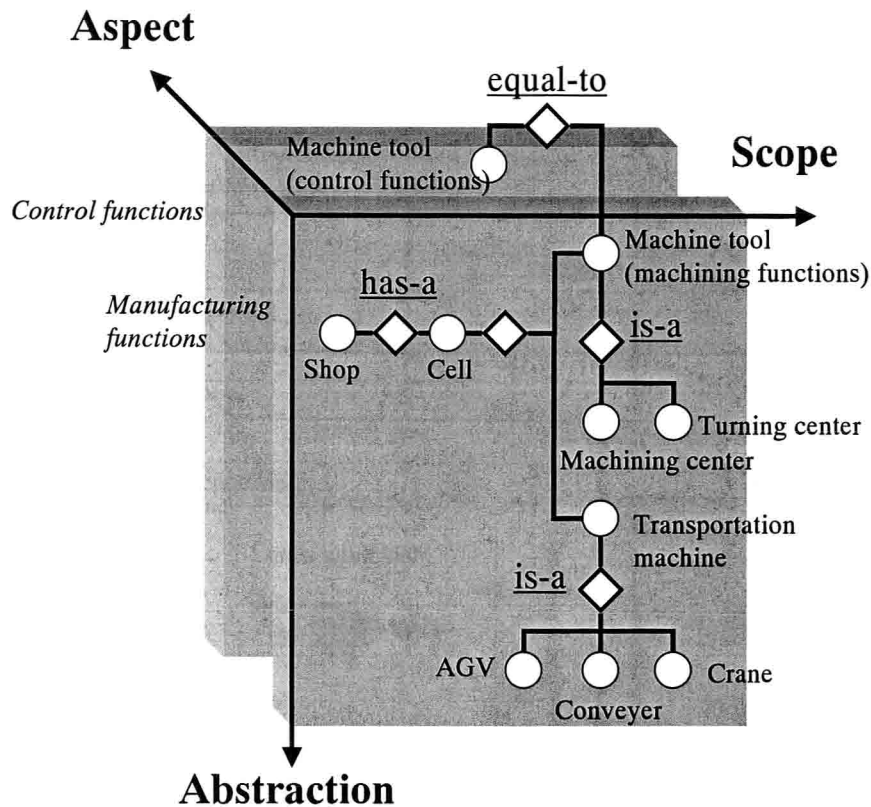


Fig. 2.2 Category of model repository

ってもカテゴリが異なる場合、モデルの内容が異なる。

2.4 生産システムモデルの原型の表現

Fig. 2.1に示すモデルリポジトリを構築する際には、視点、抽象性および範囲により定まるカテゴリにしたがって、生産システムの構成要素に関するモデルの原型を作成するとともに、異なるカテゴリ間の原型の間関係を記述しておく必要がある。

本章では、オブジェクト指向 [本位田 93] の考え方にに基づき、生産システムおよびその構成要素のモデルの原型を表現する。モデル化の対象には、機械製品の生産システムおよびその構成要素、すなわち、ショップフロア、エリア、セル、ステーション、工作機械、搬送装置、自動倉庫などの生産設備、素材、部品、組立品などの生産対象、生産設備の制御機能、生産スケジューリングなどの計画機能を考える。

2.4.1 単一のカテゴリに属する原型の記述

生産システムモデルの原型を、オブジェクト指向モデルのクラスとして記述する。以下、これをモデル

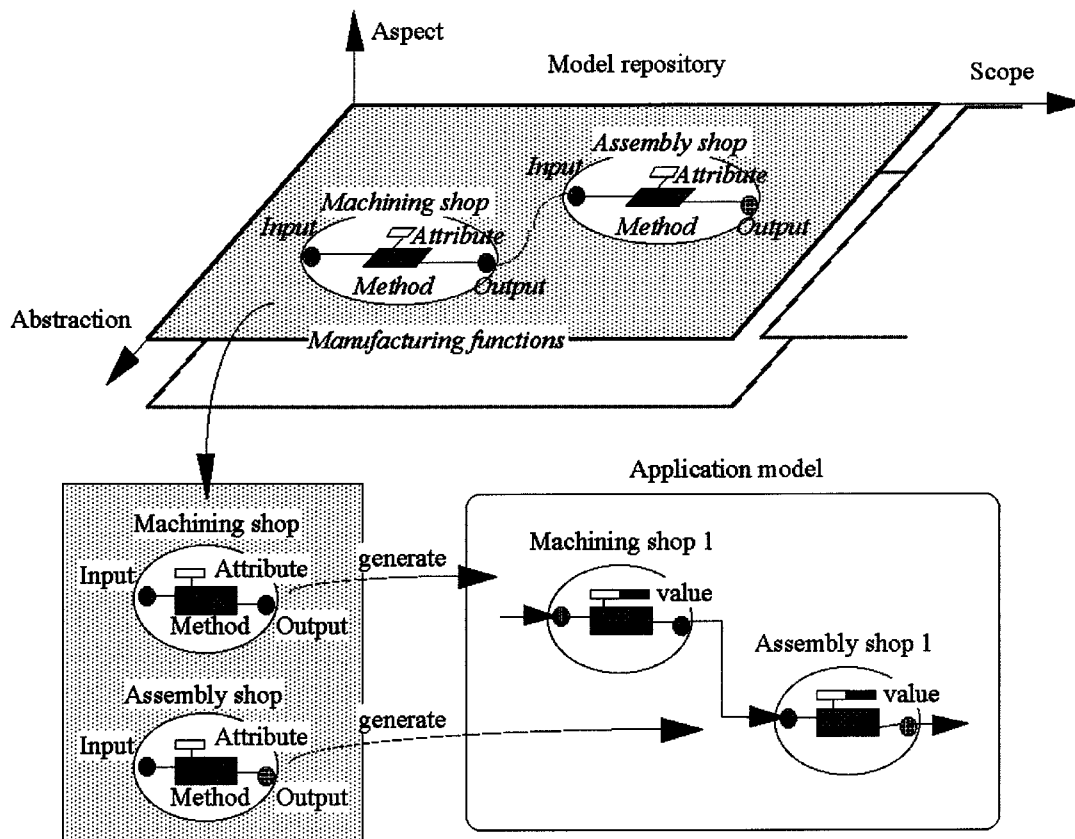


Fig. 2.3 Generation of application model

クラス(Model class)と呼ぶ。各モデルクラスには、名称、属性、およびメソッド(Method)を与える。メソッドには、入力に対して状態を変化させる手順や、出力を決定する手順を記述する。出力されるのは、属性値か、別のメソッドを呼び出すメッセージである。

生産システムモデルの設計者は、このモデルの原型の属性に値を与えることで、オブジェクト指向モデルのインスタンス(Instance)を作成する。以下、これをモデルインスタンス(Model instance)と呼ぶ。モデルインスタンスが生産システム全体またはその構成要素のモデルを表す。また、メッセージを授受するモデルクラス間の関係に基づいて、モデルインスタンスを接続し、複数のモデルインスタンスからなるアプリケーションモデルを構築する。Fig. 2.3に、単一のカテゴリに属するモデルクラスの例、およびこれらのモデルクラスから生成したモデルインスタンスを組み合わせるアプリケーションモデルを作成する例を示す。この例では、加工ショップと組立ショップのモデルクラスから、それぞれモデルインスタンスを生成し、それらを接続してアプリケーションモデルを作成している。

2.4.2 異なるカテゴリに属する原型の関係の記述

異なるカテゴリに属する生産システムモデル間の整合性の管理を容易にするため、モデルリポジトリ

のモデルクラス間に以下の関係を記述する。

(1) 視点の異なるモデルクラスの間

視点の異なるモデルクラスの場合、モデル化の対象が同一であっても異なる属性を持つ。例えば、2つのモデルクラスは、その名称 X, Y, 属性 p_i, q_i および属性値 a_i, b_i を用いて、以下のよう
に表される。

$$(X \ (p_1 \ a_1) \ (p_2 \ a_2) \ \dots \ (p_n \ a_n)) \quad (2.2)$$

$$(Y \ (q_1 \ b_1) \ (q_2 \ b_2) \ \dots \ (q_m \ b_m)) \quad (2.3)$$

ここで、X, Y の属性 p_i, q_i を元とする属性集合を A, B とする。

$$A = \{ p_1, p_2, \dots, p_n \} \quad (2.4)$$

$$B = \{ q_1, q_2, \dots, q_m \} \quad (2.5)$$

モデルクラス X, Y の視点異なる場合には、以下の関係が成り立つ。

$$A \neq B \quad (2.6)$$

ただし、集合 A, B が互いに素であるとは限らない。すなわち、A, B の共通部分が、必ずしも空集合になるとは限らない。

$$A \cap B \neq \phi \quad (2.7)$$

モデルリポジトリにおいて、同一のモデル化対象を表すモデルクラスの間には対応関係を与えておくことが必要である。本章では、この対応関係をモデルクラス間の二項関係で表現する。例えば、生産機能の観点から搬送装置を表すモデルクラスと制御機能の観点から搬送系の制御装置を表すモデルクラスの間、「搬送装置 = 搬送系の制御装置」の二項関係を与える。

同一の対象を異なる視点からモデル化した場合、そのモデルクラスの属性集合 A, B が互いに素でないならば、共通する属性の値は等しくなければならない。

モデル化の視点が異なる場合、モデルの抽象性のレベルをどのように分割するか、あるいはモデルの時間的および空間的な範囲をどのように分割するかが異なることがある。このような場合に対しては、現状では、同一の構成要素を表しているモデルクラスに対してのみ二項関係により対応関係を記述するものとする。

(2) 抽象性の異なるモデルクラスの間

抽象性の異なるモデルクラス間には、一方のモデルを抽象化すると他方のモデルに対応するという関係がある。ここでは、この関係を包摂と呼ぶ。モデルリポジトリにおけるモデルクラス間の包摂関係を、オブジェクト指向における is-a 関係で表現する。例えば、一般的な搬送装置を表すオブジェクトと具体的な AGV(Automated Guided Vehicle)の間に、「AGV is-a 搬送装置」の関係を与える。

抽象性の高いモデルクラスの属性は、低いモデルクラスに継承される。例えば、2 つのモデルクラス(X (p₁ a₁) (p₂ a₂) ... (p_n a_n)), (Y (q₁ b₁) (q₂ b₂) ... (q_m b_m))において、XがYよりも抽象性が高い場合、Xの属性集合 A は、Yの属性集合 B に含まれる。この継承される属性の値は等しくなければならない。

$$A \subset B \quad (2.8)$$

(3) 範囲の異なるモデルクラスの間

範囲の異なるモデルクラスの間には、一方のモデルが、空間的または時間的に他方のモデルに含まれるという関係がある。ここでは、この関係を包含と呼ぶ。モデルリポジトリにおけるモデルクラス間の包含関係を、オブジェクト指向における has-a 関係で表現する。例えば、加工ショップを表すオブジェクトとその中の加工設備を表すオブジェクトの間には、「加工ショップ has-a 加工設備」の関係を与える。

包含関係を持つオブジェクト間では、空間あるいは時間に関する属性の間に大小関係および包含関係が要求される。例えば、2 つのモデルクラス(X (p₁ a₁) (p₂ a₂) ... (p_n a_n)), (Y (q₁ b₁) (q₂ b₂) ... (q_m b_m))において、XとYの範囲が異なる場合、「X has-a Y」の関係が存在するならば、空間あるいは時間に関する属性のうち、面積や総処理時間などの数値を表す属性 p_i, q_iの値 a_i, b_iの間には大小関係が成り立つ。

$$a_i > b_i \quad (2.9)$$

また、大きさや加工時間など空間あるいは時間の領域や区間を表す属性の対(p_j, p_k), (q_j, q_k)に対する値の対(a_j, a_k), (b_j, b_k)には包含関係が成り立つ。

$$a_j < b_j \text{ かつ } a_k > b_k \quad \text{ただし } a_j < a_k \text{ かつ } b_j < b_k \quad (2.10)$$

Fig. 2.4に、カテゴリの異なるモデルクラスの例、および各カテゴリのモデルクラスから生成したモデルインスタンスを用いてアプリケーションモデルを作成する例を示す。この例では、加工ショップと AGV のモデルクラスを用いて、各アプリケーションモデルを作成している。加工ショップと AGV のモデルクラス間には、包含関係を表す has-a 関係が定義されているため、そのモデルインスタンス間には包含関係

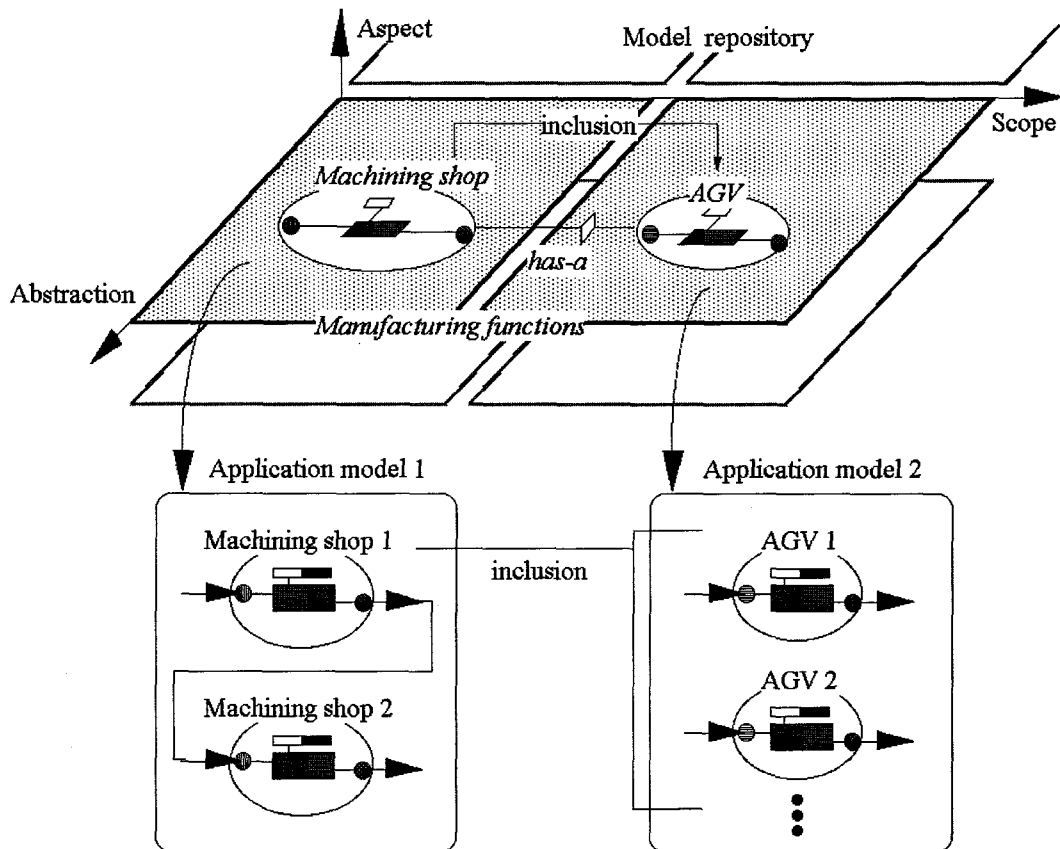


Fig. 2.4 Generation of models on different categories

が成立する。

2.5 生産システムモデル間の整合性の管理

2.5.1 整合性の管理手順

カテゴリの異なるアプリケーションモデルを同時に複数個作成する場合、これらのモデルインスタンスが持つ属性値の間に整合性が要求される。すなわち、同一の対象を表すモデルインスタンスの属性値が一致または整合していることが必要となる。本章では、モデルインスタンス間の整合性を以下の手順で管理する。

- (1) 整合させるべきモデルインスタンスの検出。
- (2) モデルインスタンスの属性値の比較。
- (3) モデルインスタンスの属性値の修正。

上記の内、(1)の処理、すなわち整合させるべきモデルインスタンスの検出が最も重要である。このため、以下では、この部分の処理手順を述べる。

ここで提案する手法では、基本的にオブジェクト指向モデリングにおける is-a 関係および has-a 関係に基づいて、整合させるべきモデルインスタンスの検出を行う。しかし、視点が異なるモデル間の整合性をも含めて考慮することが、本章では求められる。

2.5.2 整合すべきモデルインスタンスの検出

カテゴリの異なるモデルインスタンス間で、整合性が要求される場合として、以下の2つの場合を考慮することができる。

(1) 同一の対象を表す場合

一対のモデルインスタンスが、実生産システムにおける同一の対象を表現している場合である。視点または抽象性が異なるモデルインスタンスが、同一の対象を表現している可能性がある。

(2) 対象の間に包含関係がある場合

一対のモデルインスタンスが表現している対象が、実生産システムにおいて包含関係を持つ場合である。範囲が異なるモデルインスタンスの表現対象が、包含関係を持つ可能性がある。

以上のことから、整合性を要求されるモデルインスタンスの検出は、視点、抽象性および範囲が異なるモデルインスタンスに対して、以下の方法で行う。

(1) 視点が異なる場合の同一性の検出

モデルリポジトリにおいて、視点が異なるモデルクラスの間には、対象の同一性を表す二項関係が与えられている。このため、同一の対象を表現するモデルインスタンスを検出するには、まず、同一の対象を表すモデルクラスから生成されたモデルインスタンスを求める。次に、両者に共通の属性値を調べ、同一の対象を表しているか否かを確認する。

(2) 抽象性が異なる場合の同一性の検出

モデルリポジトリにおいて、抽象性が異なるモデルクラスの間には、is-a 関係が与えられている。このため、同一の対象を表現するモデルインスタンスを検出するには、まず、モデルインスタンスが生成されたモデルクラスの間には is-a 関係があるか否かを検証する。次に、is-a 関係がある場合、属性値間の継承関係を調べ、同一の対象を表しているか否かを確認する。

(3) 範囲が異なる場合の包含関係

モデルリポジトリにおいて、範囲が異なるモデルクラスの間には、has-a 関係が与えられている。

このため、モデルインスタンス間の包含関係を検出するには、まず、モデルインスタンスが生成されたモデルクラスの中に has-a 関係があるか否かを検証する。次に、has-a 関係がある場合、属性値間の大小関係と包含関係を調べ、モデルインスタンス間に包含関係があるか否かを確認する。

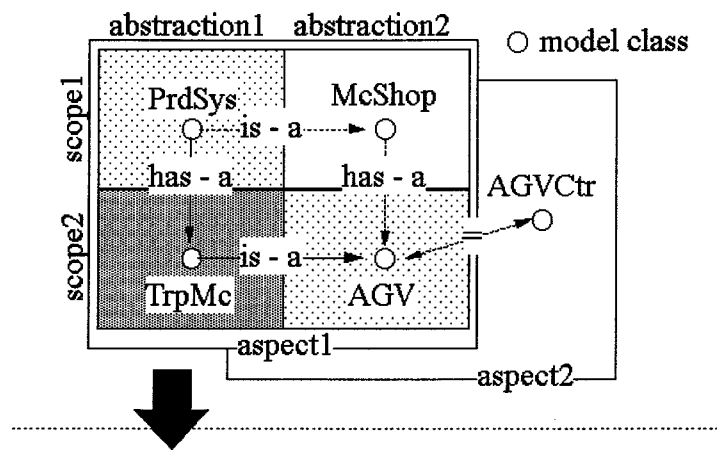
以上、(1)から(3)のプロセスにより、モデルインスタンス間の同一性および包含関係が求まる。しかし、同一のモデルクラスから多数のモデルインスタンスが生成されること、および属性値間に要求される関係をモデルクラスに全て記述することが困難なことより、現状では、これらのプロセスにより整合すべきモデルインスタンスの候補を求め、最終的な判断はモデル作成者が行うものと考えている。

2.5.3 整合性管理の一例

これまで述べてきた考え方にしたが、モデルリポジトリおよび生産システムモデルの間の整合性を管理するシステムのプロトタイプを、オブジェクト指向言語の Smalltalk を用いて開発した。ここでは、このプロトタイプシステムを用いて簡単なモデルインスタンスの作成、およびモデルインスタンス間の整合性の検証を行った結果について述べる。

まず、それぞれ 2 つの視点、抽象性および範囲についてカテゴリを設定し、5 つのモデルクラスを作成した。これらは、Fig. 2.5の上段に示すように、生産システム(PrdSys)、加工ショップ(McShop)、搬送装置(TrpMc)、AGV(AGV)および AGV の制御装置(AGVCtr)である。これらのモデルクラス間の関係を、Table 2.1からTable 2.3に示す。

Model repository



Application Models

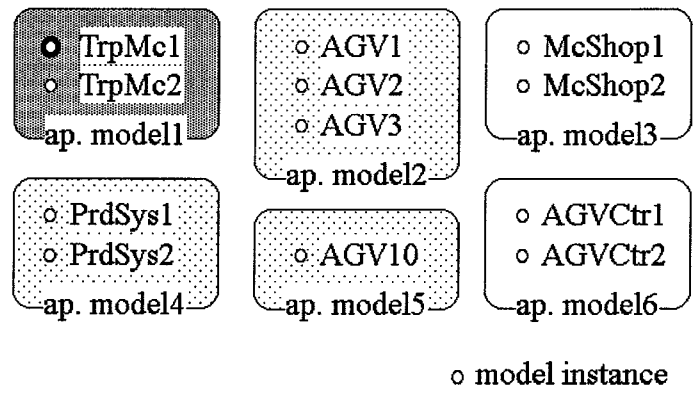


Fig. 2.5 Class objects and instance objects

Table 2.1 Relationship on different aspect

relationship	aspect 1	aspect 2
=	AGV	AGVCtr

Table 2.2 Relationship on different abstraction

relationship	abstraction 1 (super-)	abstraction 2 (sub-)
is-a	PrdSys	McShop
is-a	TrpMc	AGV

Table 2.3 Relationship on different scope

relationship	scope 1 (super-)	scope 2 (sub-)
has-a	PrdSys	TrpMc
has-a	McShop	AGV

Table 2.4 Model instances

models	model instances		
ap. model 1	TrpMc1	TrpMc2	
ap. model 2	*AGV1	AGV2	AGV3
ap. model 3	McShop1	McShop2	
ap. model 4	*PrdSys1	PrdSys2	
ap. model 5	*AGV10		
ap. model 6	AGVctr1	AGVctr2	

次に、これらのモデルクラスから、6つのアプリケーションモデルを作成するために、合計12のモデルインスタンスを作成した。これらのモデルインスタンスをFig. 2.5の下端およびTable 2.4に示す。

最後に、これらのモデルインスタンスの間で、整合すべきものの組み合わせを求めた。一例として、モデルインスタンス TrpMc1 と整合を取るべきモデルインスタンスの検出を試みると、まず、TrpMc1 のモデルクラス TrpMc と抽象性および範囲の観点から関係のあるモデルクラスとして AGV および PrdSys が求まる。次に、これらのモデルクラスから生成されたモデルインスタンスとして、AGV1, AGV2, AGV3, AGV10, PrdSys1, PrdSys2 が求まる。最後に、これらのモデルインスタンスの属性値を調べることで、最終的に、TrpMc1 は、AGV1 および AGV10 と同一であり、かつ PrdSys1 に包含されることが検出された。すなわち、Table 2.4の*印を付けたモデルインスタンスの間に対応があることがわかった。Fig. 2.6に、この検出結果を、プロトタイプシステムのインタフェースを用いてディスプレイ上に表示した例を示す。

2.6 結言

本章では、生産システムのモデルリポジトリにおいて生産システムモデルを分類する基準を提案し、その分類基準にしたがって生産システムモデル間の関係を示すとともに、表現形式の異なる複数の生産システムモデル間における整合性管理の方法を提案した。以下に本章の内容をまとめる。

- (1) 生産システムモデルの構成要素の原型を集めたモデルリポジトリを提案した。生産システムモデルの作成と利用の手順として、モデルリポジトリから選択した生産システムモデルの原型に属性値を与えて生産システムモデル作成し、それを組み合わせて、アプリケーションモデルを作成する方法を提案した。

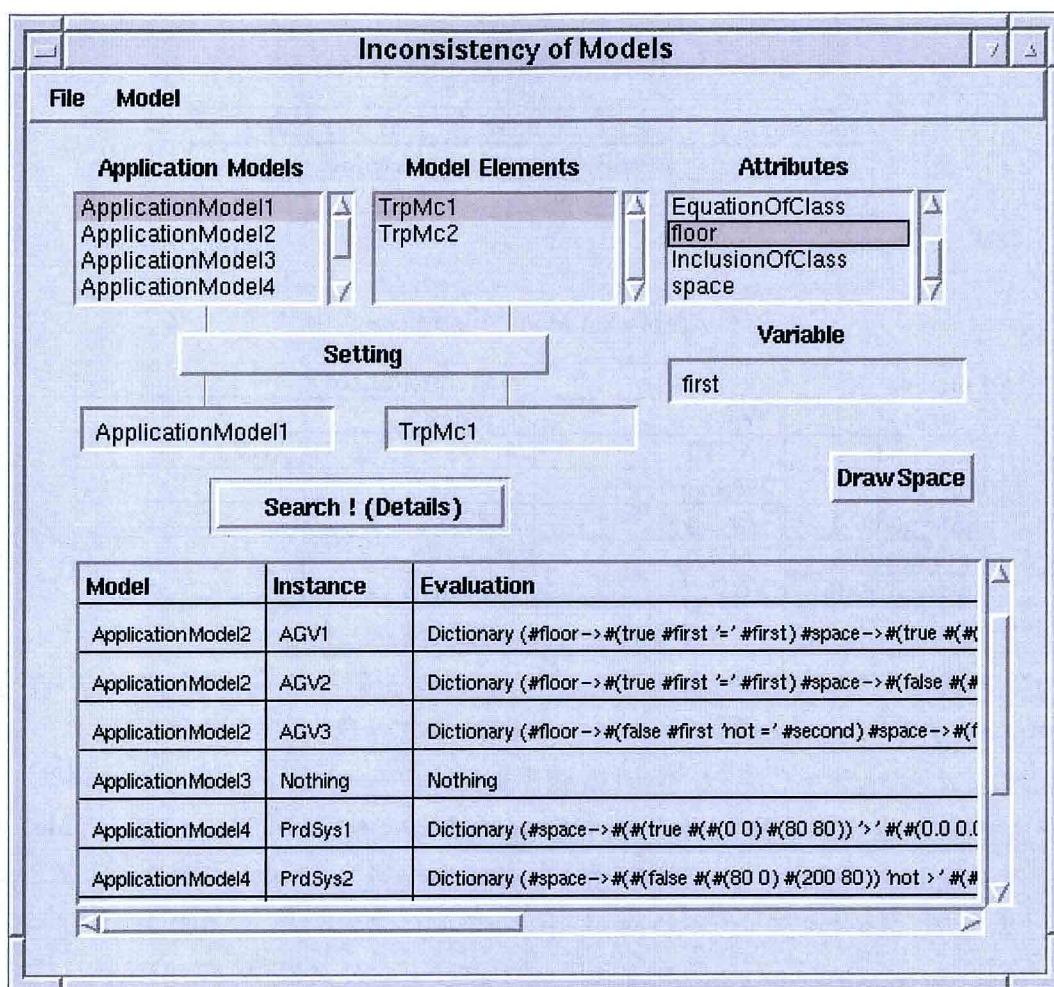


Fig. 2.6 Results of detected model instances to be consistent

- (2) モデルリポジトリにおいて、生産システムモデルの構成要素を分類するための基準として、視点、抽象性、および範囲の3つの軸を提案した。この3軸に基づいて、生産システムモデルの原型は、モデルリポジトリに構造化され、格納される。
- (3) オブジェクト指向の考え方に基づいて、生産システムモデルの原型をモデルクラスとして表現する方法を提案した。モデルクラスの属性に値を与えることで生成したモデルインスタンスを接続し、複数のモデルインスタンスからなるアプリケーションモデルを構築する。
- (4) (2)の3つの軸に基づくモデルクラスの間関係を提案した。ここでは、視点が異なるモデルクラスの間には対象の同一性を表す二項関係を、抽象性が異なるモデルクラスの間には is-a 関係を、範囲が異なるモデルクラスの間には has-a 関係を与える。
- (5) モデルリポジトリにおけるモデルクラスの間関係をに基づいて、アプリケーションの異なるモデルインスタンスの間で、整合性を保証すべきモデルの構成要素を検出する手法を提案した。すなわち、

視点, 抽象性および範囲の異なるモデルインスタンスにおける対象の同一性を検出する方法を提案した。

- (6) オブジェクト指向言語の Smalltalk を用いて, モデルリポジトリおよび生産システムモデル間の整合性を管理するプロトタイプシステムを開発した。さらに, このプロトタイプシステムを用いて, 簡単なモデルインスタンスの作成, およびモデルインスタンス間の整合性の検証を行った。ただし, 現状の検出方法は, 整合性が要求されると考えられる構成要素の候補を検出するだけであり, 最終的な判断は, モデルの作成者が行う必要がある。

参考文献

[AMICE 92] 第 1 章と同じ。

[Doumeingts 88] Doumeingts G., System analysis technique, *Computer-aided production management*, Springer-Verlag, (1988), 201 - 224.

[Doumeingts 93] 第 1 章と同じ。

[Williams 93] 第 1 章と同じ。

[大見 93] 第1章と同じ。

[本位田 93] 本位田・山城, オブジェクト指向システム開発, 日経BP社 (1993).

第 3 章 生産システムの設計プロセスの記述と分析

3.1 緒言

第 2 章では、生産システムの計画、設計、実装、運用および制御にいたるライフサイクルにおいて、表現形式の異なる複数の生産システムモデルの整合性を保証するために、モデルリポジトリを用いた生産システムモデルの作成手順、および整合性管理の手順を提案した。第 3 章および第 4 章では、生産システムの設計フェーズにおける生産システムモデルとその利用手法について考察する。まず、第 3 章では、生産システムの設計プロセスの問題点について考察する。

生産システムの要求定義から詳細設計までの設計過程を、ここでは、MSE プロセス (Manufacturing System Engineering Process) と呼ぶ。MSE プロセスを合理的かつ系統的に行うには、MSE プロセスにおける設計情報の記録、参照および適切な設計手順の指示などを、計算機を用いて統合的に支援するための MSE プロセス支援ツールが必要である。MSE プロセス支援ツールの開発には、実際の MSE プロセスの特徴や問題点の分析が必要である。しかし、MSE プロセスを分析した研究は数少なく、その特徴や問題点は十分には明らかにされていない。

そこで、本章では、まず、第 2 章で提案した生産システムモデルを拡張して、MSE プロセスのモデル化手法を提案し、MSE プロセスを記述するためのツールを提案する。次に、ある製品を製造する生産システムの設計実験を行い、その MSE プロセスを記述する。その結果を分析して、MSE プロセスの問題点を明らかにする。

3.2 MSE プロセス

3.2.1 MSE プロセスの定義

本章では、MSE プロセスを、生産システムの要求定義から詳細設計までの過程に含まれる複数の設計活動であると考え、ひとつの設計活動において、生産システムの設計者は、生産システム全体またはその構成要素の仕様の分析、設計案の作成、評価を行い、設計対象となる生産システム全体またはその構成要素の属性値を決定する。したがって、Fig. 3.1 に示すように、これら複数の設計活動の連鎖により、生産システム全体の設計を行う。

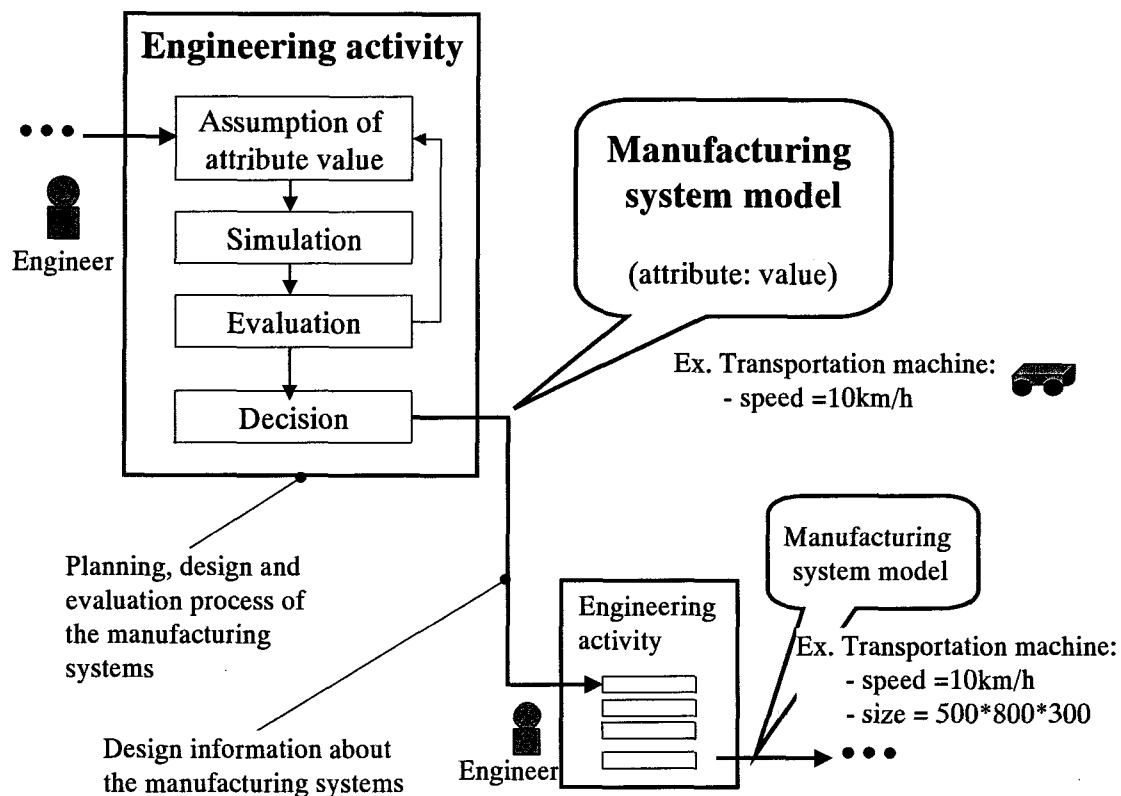


Fig. 3.1 Manufacturing system engineering process

3.2.2 MSE プロセスの記述に関する既存の研究

生産対象製品の設計プロセスの記述に関する研究は、これまでにいくつか行われている。窪田ら [窪田 96] や、田浦ら [田浦 99] , 荒井ら [荒井 98] [岡田 91] は、設計意図の伝達を目的として、製品およびその設計プロセスのモデルを提案した。仲ら [仲 97a] [仲 97b] は、設計行為のモデルを提案した。また、菊地ら [菊地 95] [菊地 96] は、設計および生産活動における情報の入出力関係を形式的に表現する形式仕様言語を提案した。これらの研究では、製品の設計プロセスに関するモデルを提案しているが、生産システムの設計プロセスについては十分に検討されていない。

一方、MSE プロセスの記述に関する研究としては、Mertins ら [Mertins 98] [Mertins 99] や小島ら [小島 96a] [小島 96b] 等の研究を挙げることができる。Mertins らは、生産システムの製造プロセスにおける論理的モデル(Logical model)と物理的モデル(Layout model)の2種類のモデルを提案した。しかし、実際の MSE プロセスの記述や分析は、十分に行われていない。小島らは、MSE プロセスに関する知識を設計者から抽出し、MSE プロセスを記述した。また、精機学会¹総合生産システム専門委員会 [IMS 委員会 80] や石村 [石村 93] は、生産設備の自動化や FA (Factory

¹ 現在:精密工学会

Automation)化を目的として、生産設備の設計プロセスを説明した。しかし、これらの研究では、実際の MSE プロセスに内在する問題が明示されていない。

実際の MSE プロセスを正確かつ十分に記述するためには、生産システムの設計を行うとともに、その MSE プロセスを記述する必要があると考える。そこで、本章では、ある製品を製造する生産システムの設計を実際に行い、その設計の進行と同時に、MSE プロセスの記述を行う。次に、これを分析して、現状の MSE プロセスにおける問題点を明らかにし、MSE プロセス支援ツールを開発する。

3.3 MSE プロセスの記述方法

MSE プロセスのモデルは、生産システムの設計活動を表現する設計活動モデルと、設計対象となる生産システム全体またはその構成要素を表現する生産システムモデルの 2 種類のモデルが存在すると考える。

3.3.1 設計活動モデル

MSE プロセスにおいて、生産システム全体またはその構成要素の仕様の分析、設計案の作成、評価および決定からなるひとつの設計活動をエンジニアリングアクティビティ(Engineering Activity)と定義し、以下、これを EA と表す。EA では、入力情報と評価手法に基づき、生産システム全体またはその構成要素の属性値を決定し、これが EA の出力情報になる。

本章では、EA を以下の情報により記述する。

(1) ID 番号および名称

EA を特定するための情報を表す。

(2) 設計者名または組織名

EA で出力情報を決定した人物、または組織の名称を表す。

(3) 設計の開始および終了時刻

EA において設計を開始した時刻および終了した時刻を表す。

(4) 入力情報

EA の入力情報を表す。これは、MSE プロセスの開始時に与えられる初期仕様、あるいは他の EA の出力情報である。

(5) 出力情報

EA の出力情報を表す。すなわち、この EA で決定した生産システム全体またはその構成要素の

属性値を表す。

(6) 評価手法

設計案である生産システム全体またはその構成要素の属性値を評価するための計算式やルールなどを表す。

複数の EA を考える場合、ある EA の出力情報が他の EA の入力情報となる関係、すなわち情報授受の関係を記述することが必要である。情報授受の関係を記述するために、入力情報には、その情報を出力した EA の ID 番号を与える。

3.3.2 生産システムモデル

工作機械やライン等、EA で設計対象となる生産システム全体またはその構成要素のモデルを生産システムモデルと呼ぶ。生産システムモデルの内容および構造については、第 2 章で示したとおりである。

MSE プロセスでは、設計者がひとつの EA において、生産システム全体またはその構成要素の状態や特性を表す属性値を決定し、それを生産システムモデルの属性値として記述する。この生産システムモデルが、EA の出力情報となる。また、この生産システムモデルが他の EA において参照される場合には、参照した EA の入力情報となる。

3.3.3 MSE プロセスの記述ツール

本章では、マイクロソフト社製ソフトウェア Microsoft Word で作成した EA シート(Engineering Activity Sheet)、および住友金属システム開発社製ソフトウェア Design/IDEF の IDEF0 を用いて、MSE プロセスを記述する。EA シートを Fig. 3.2 に、IDEF0 [FIPS 93] を Fig. 3.3 に示す。

EA シートには、EA の ID 番号と名称、設計者名、設計の開始および終了時刻、入出力情報、および評価手法を記述する。すなわち、1 枚のシートに、ひとつの EA に関する情報を詳細に記述する。これを全ての EA において繰り返すことで、MSE プロセスの全体を記述する。

また、IDEF0 では、複数の EA 間における情報授受の関係を記述する。ここでは、箱がひとつの EA を表し、左から箱へ入る矢印が EA への入力情報を、箱の右から出る矢印が EA からの出力情報を表す。

ID Number : M-1-3-1-1 Engineer : Engineer-B
 Starting time : 1998. AM/PM → Finishing time : 1998. AM/PM

Source of input data	Input data (Manufacturing system model)			Name of engineering activity	Output data (Manufacturing system model)		
	Object name	Attribute	Value		Object name	Attribute	Value
M-1-2-2-1	Machining equipment in the upper housing line	Average cycle time	7.0 minute	Determination of system specifications, 1st step; Determination of Number of the machine	Upper housing line	Number of the machine	4
Initial spec.	Upper housing line	Production volume	10000 pcs/M				
Initial spec.	Upper housing line	On-duty operation time / day	18 hour				
Initial spec.	Upper housing line	Off-duty operation time / day	1 hour				
Initial spec.	Upper housing line	Operation days / day	20 days				



Evaluation method:

From the expression;
 (Machining total time) = (Average cycle time) * (Amount of production)
 = 7 minute * 10000 = 70000 minute = 1166.7 hour

From the expression ;
 (Time of operation a month)
 = (On-duty operation time + Off-duty operation time) * (Operation days)
 = (18 hour + 1 hour) * 20 days = 380 hour

From the expression ;
 (Number of the machine) = (Machining total time) / (Time of operation a month)
 = 1166.7 / 380
 = 3.1 (Number of machine when 100% operating)
 < 4 (Operation rate; 76.8 %)

Fig. 3.2 Example of engineering activity sheet

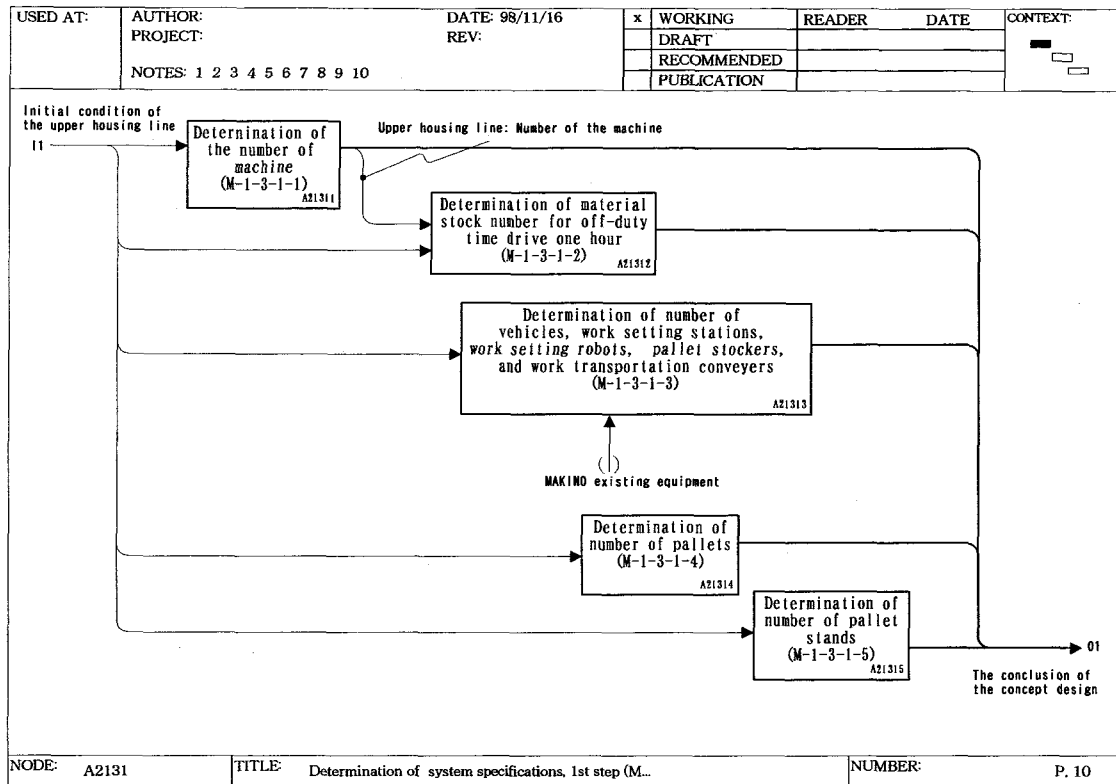


Fig. 3.3 Example of IDEF0 notation

3.4 MSE プロセスの分析

3.4.1 MSE プロセスの記述対象および記述の手順

実際の生産システムの設計を想定したテストケースとして、Fig. 3.4に示す自動車の回転機器部品を製造する加工および組立工場の設計を行い、その MSE プロセスの記述を試みた [谷岡 99]。

この工場は、複数の加工および組立ラインで構成される。製造する回転機器部品は、大きさおよび穴加工のタイプが異なる 20 種類のバリエーションがある。また、生産量は、1 万個/月、5 万個/月、10 万個/月と連続的に増大させる。そのため、異なる品種への対応と増産をあらかじめ考慮した工場の設計が要求される。

本テストケースでは、加工ライン設計、組立ライン設計、搬送システムおよびラインレイアウト設計が、異なる企業の設計者により行われると考え、設計の進行と同時に、その MSE プロセスを記述した。記述する設計情報を注出する方法として、エキスパートシステムの構築法 [Hart 89] を参考にした。以下に、本テストケースの記述の手順を示す。

(1) 各設計者は、他の設計者の設計結果を利用しながら、生産システムの各構成要素の設計を行うと

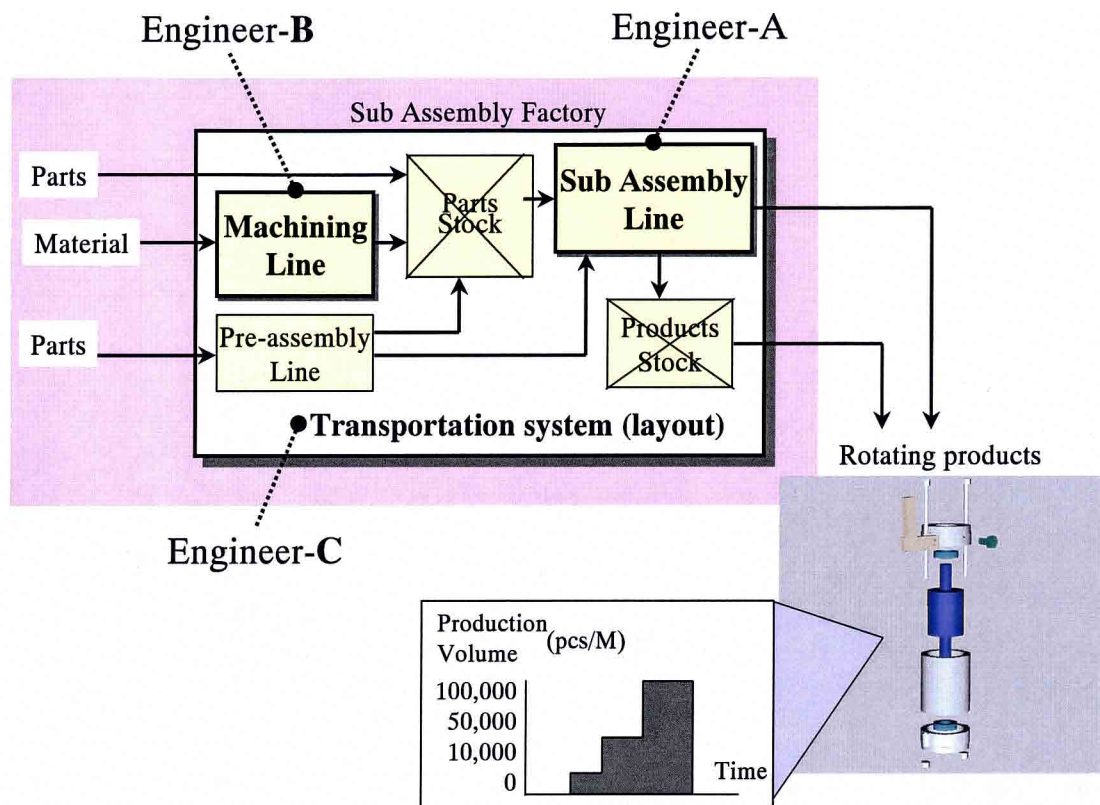


Fig. 3.4 Target manufacturing system for test case

もに、EA シート上に EA の情報を記述する。なお、ここで参加した設計者は、製造企業において生産システムの計画および設計に関与している技術者である。

- (2) 後日、会議を開き、設計者は MSE プロセスの記述者に、設計の経過を報告する。特に、個々の EA の出力情報を決定する際に利用した入力情報、その入力情報を出力情報として生成した EA の ID 番号、および評価手法について詳細に報告する。その際、入出力情報と評価手法の内容に矛盾がないかを確認する。
- (3) 会議終了後、設計者の報告により得られた情報と、会議を録画した VTR をもとにして、記述者は、IDEF0 上に MSE プロセスを記述する。
- (4) 後日、EA シートおよび IDEF0 の記述内容を全員で確認し、不足事項や誤りを追加、修正する。

以上の手順を数週間の間隔で繰り返し、設計の進行と同時に、MSE プロセスを記述した。その結果、約 180 個の EA、および EA 間における情報授受の関係を記述した。以下に、記述した MSE プロセスの概要を示す。

- (1) 製造する回転機器部品および設計する工場の初期仕様が、発注側から各設計者へ提示された。工場の初期仕様としては、工場の敷地面積や生産量の増大などの条件が提示された。
- (2) それらの情報を用いて、設計者は、加工および組立に関する所要時間の概算を求め、各ラインにおける機械の種類や数、搬送システムにおける倉庫の大きさ等を仮に決定し、サイクルタイムや面積などを評価することで、発注者側の要求を満たすことが可能か検討する。要求を満たすことができない場合は、発注者側との調整が必要になる。この段階で、組立ラインの構成は、生産量が1万個/月および5万個/月の時は作業者による手組みラインとし、10万個/月の時は機械による自動組立ラインとすることになった。
- (3) 要求を満たすことができると判断されたため、より具体的な設計が行われた。この段階では、加工方法や組立方法を決定した上で、熟練者の経験やデータに基づき、加工や組立の所要時間を詳細に求め、各ラインにおける機械の種類やライン構成、搬送システムにおける搬送設備やバッファ(Buffer)の容量、ラインレイアウトなどが具体的に決定された。

IDEF0 を用いて記述した本テストケースのMSEプロセスの一部をFig. 3.5からFig. 3.14に示す。Fig. 3.5からFig. 3.7は、加工ラインの設計プロセスの一部を表し、Fig. 3.5は全体の概要を、Fig. 3.6は検討段階の設計プロセスの一部を、Fig. 3.7は具体的な設計段階の設計プロセスの一部を表す。Fig. 3.8からFig. 3.11は、組立ラインの設計プロセスの一部を表す。Fig. 3.8とFig. 3.9は、手組みラインの設計プロセスの一部を、Fig. 3.10とFig. 3.11は、自動組立ラインの設計プロセスの一部を表す。Fig. 3.12からFig. 3.14は、搬送システムおよびラインレイアウト設計の設計プロセスの一部を表し、Fig. 3.12は全体の概要を、Fig. 3.13は検討段階の設計プロセスの一部を、Fig. 3.14は具体的な設計段階の設計プロセスの一部を表す。

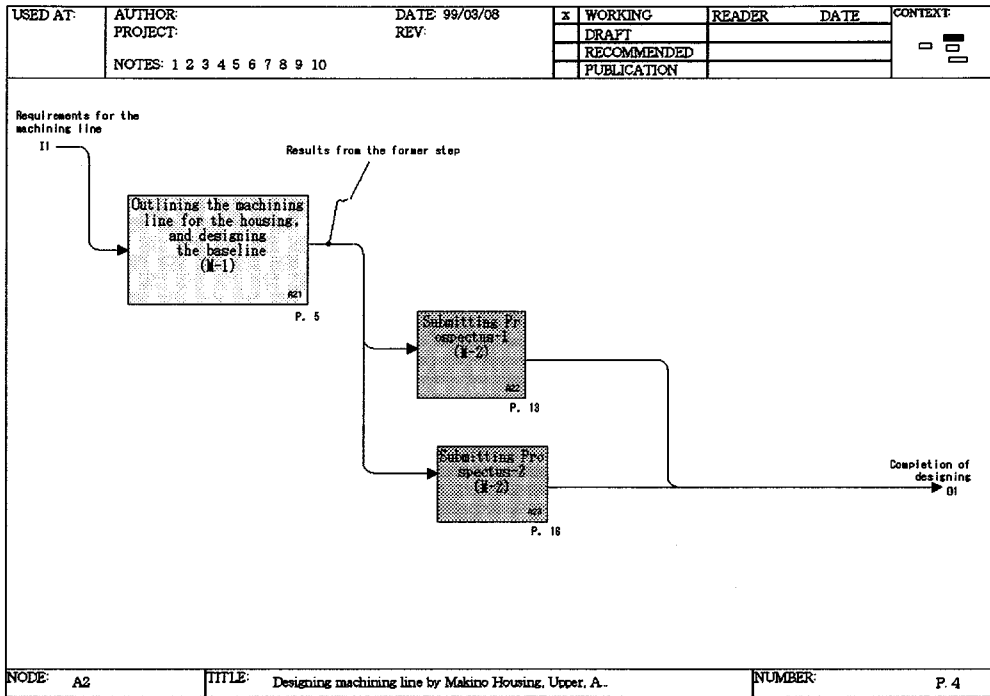


Fig. 3.5 MSE process for machining line (1)

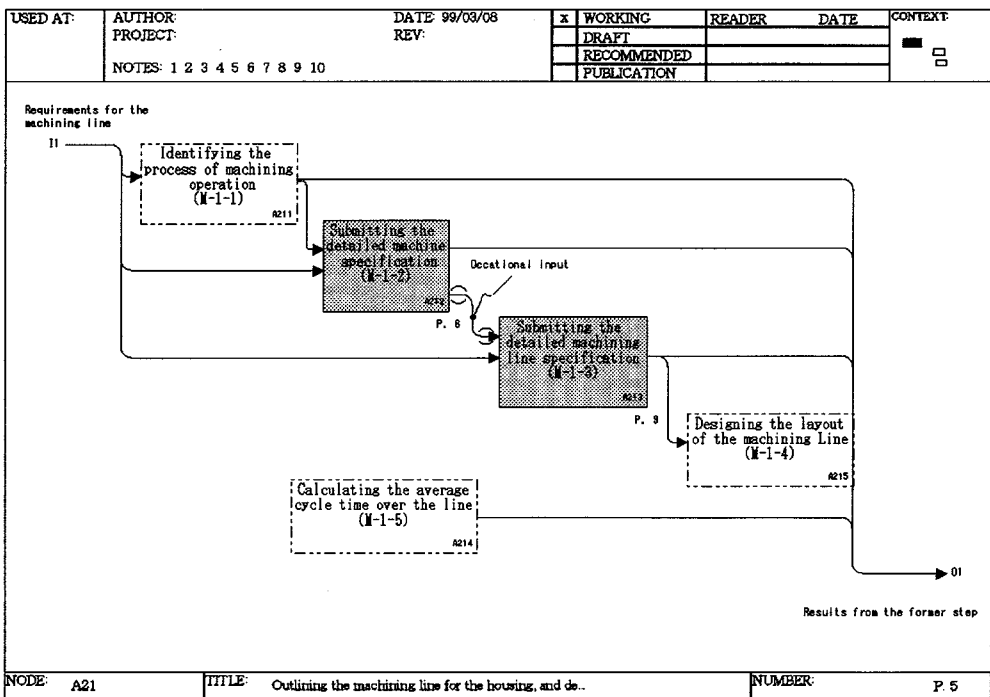


Fig. 3.6 MSE process for machining line (2)

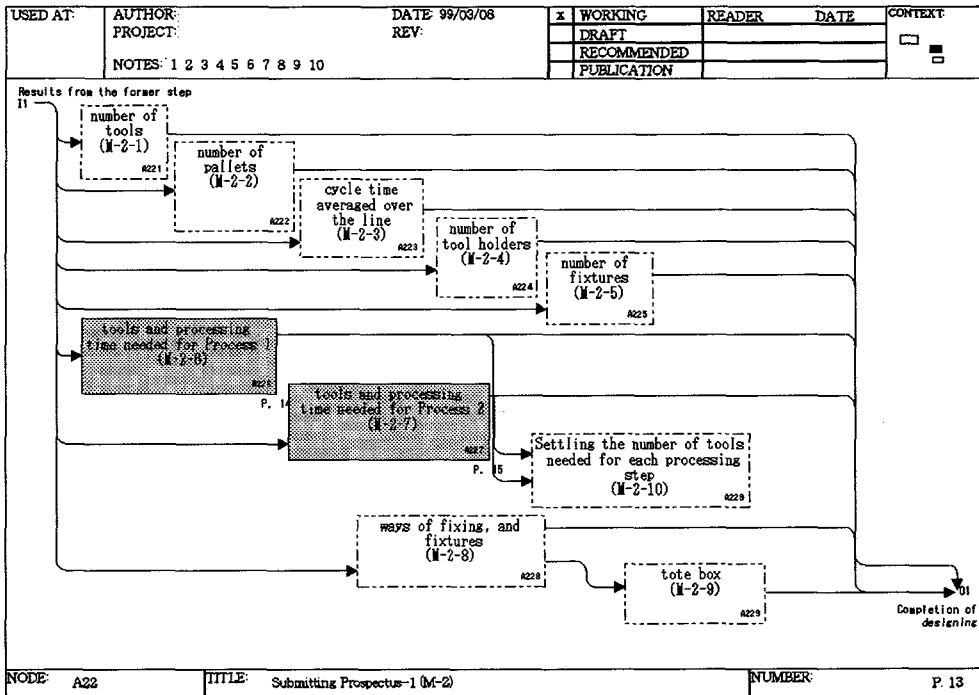


Fig. 3.7 MSE process for machining line (3)

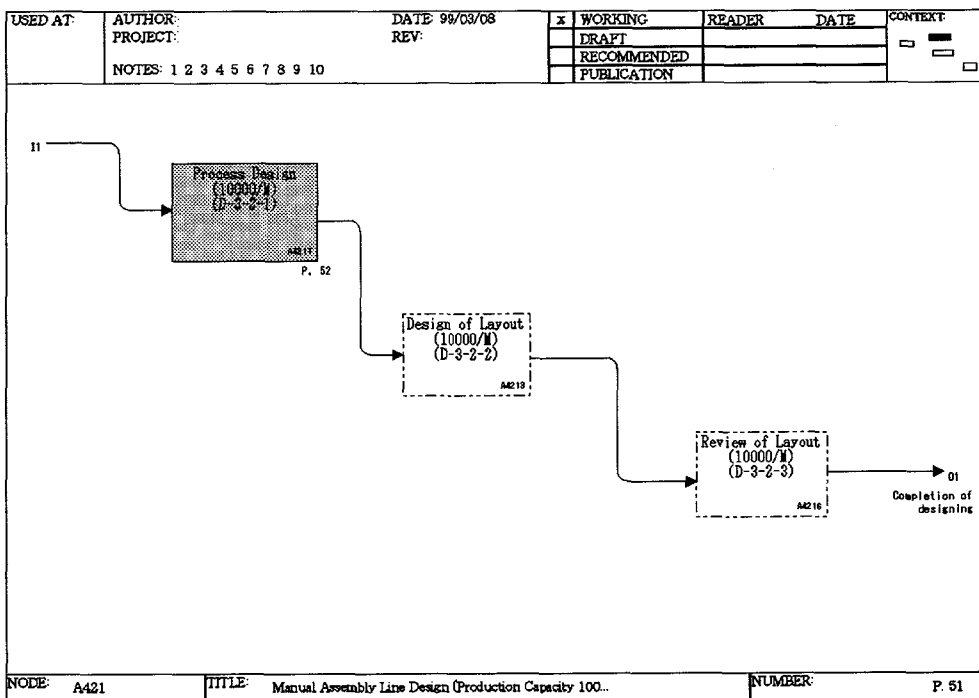


Fig. 3.8 MSE process for manual assembly line (1)

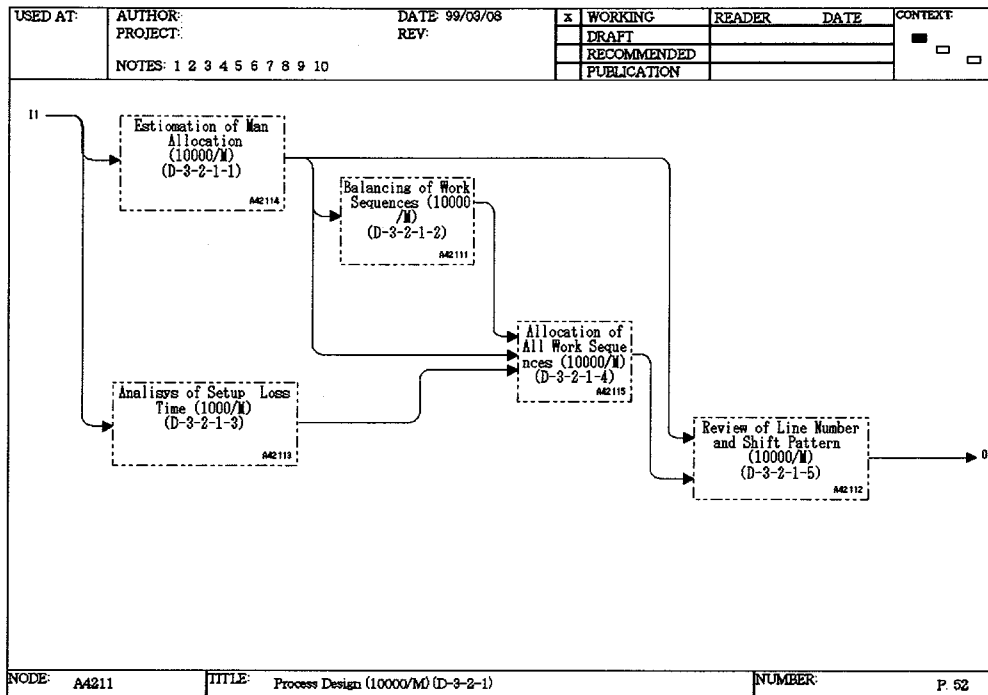


Fig. 3.9 MSE process for manual assembly line (2)

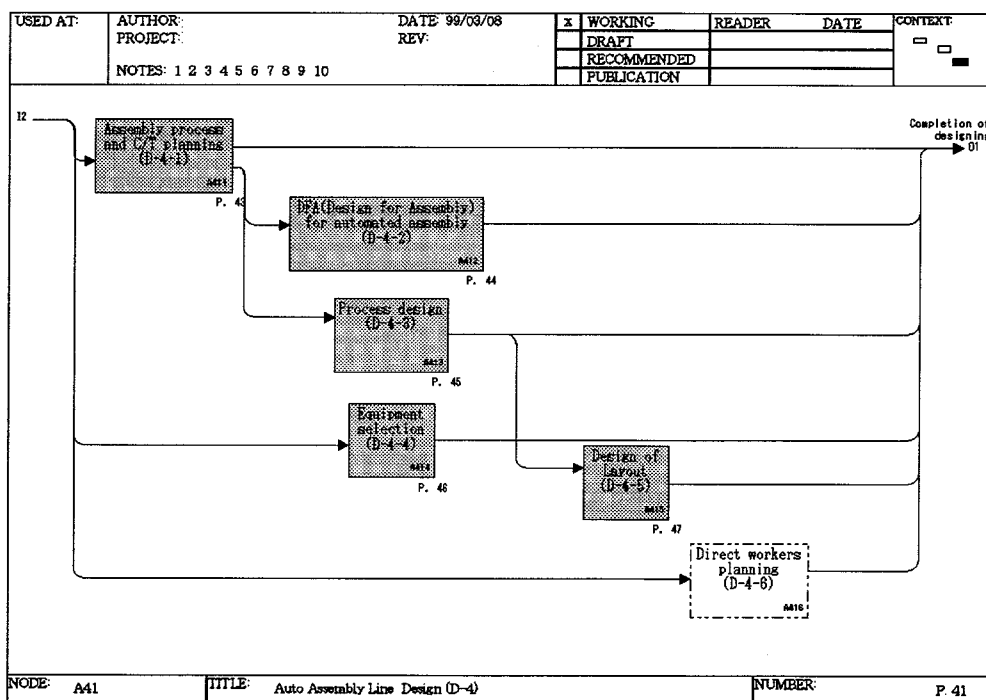


Fig. 3.10 MSE process for automated assembly line (1)

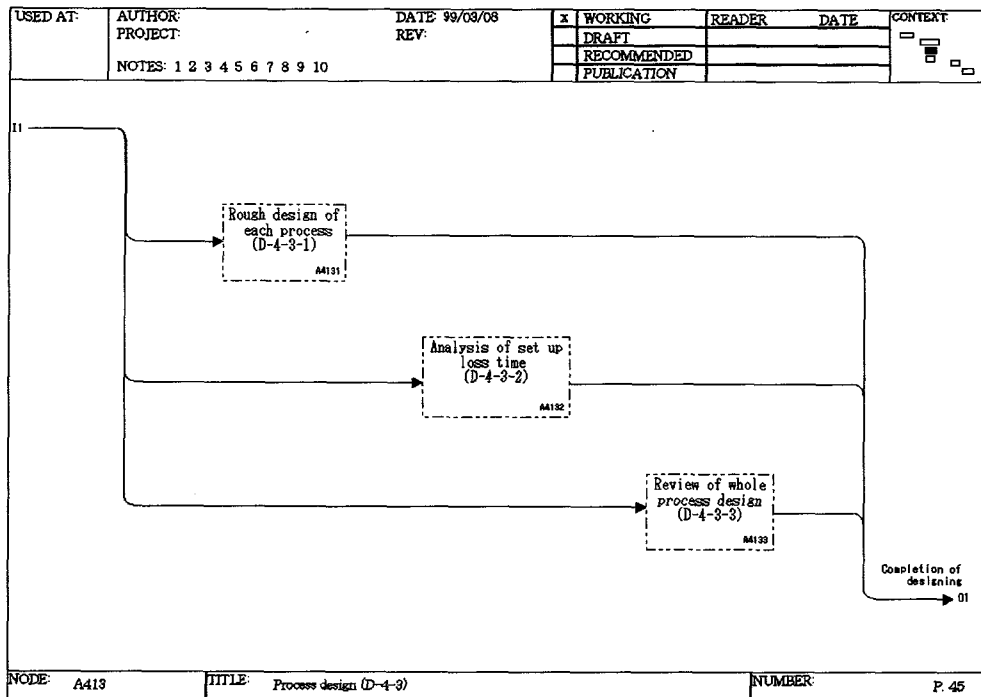


Fig. 3.11 MSE process for automated assembly line (2)

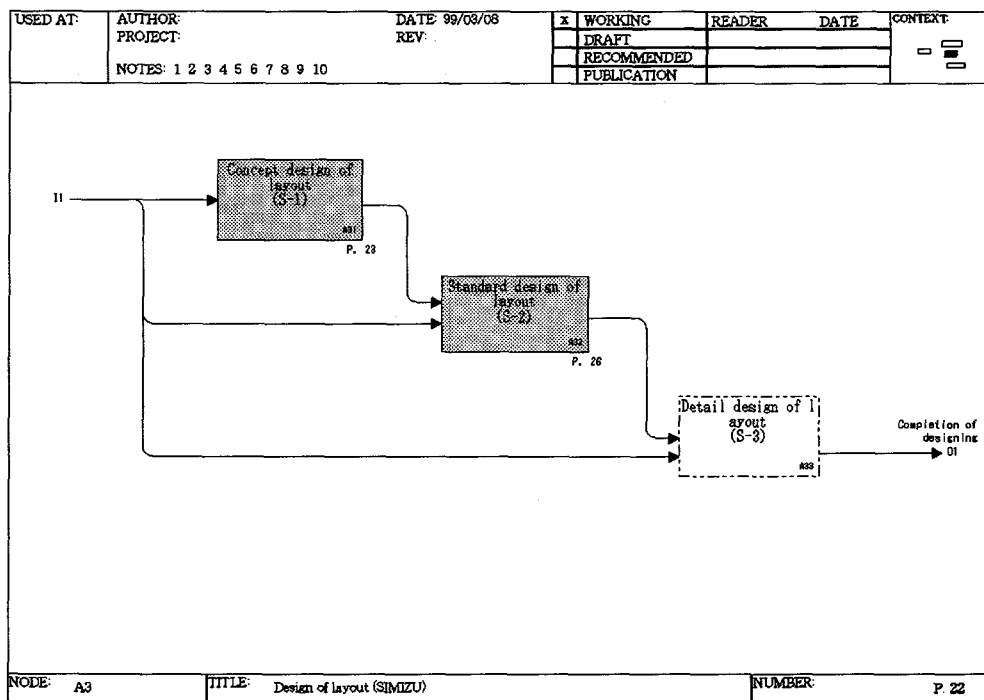


Fig. 3.12 MSE process for transportation system and line layout design (1)

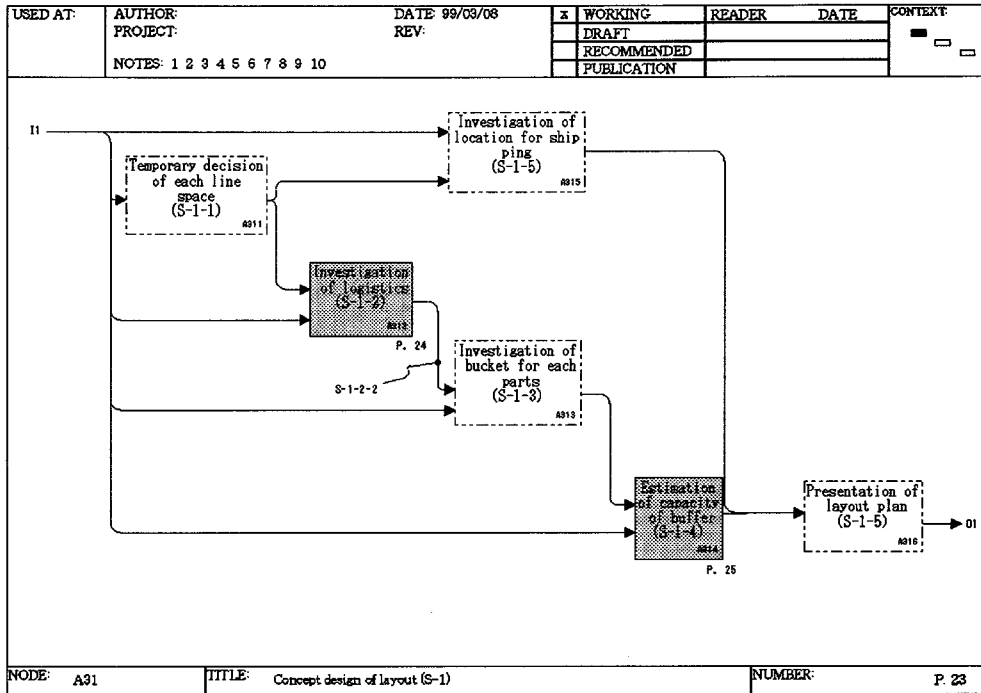


Fig. 3.13 MSE process for transportation system and line layout design (2)

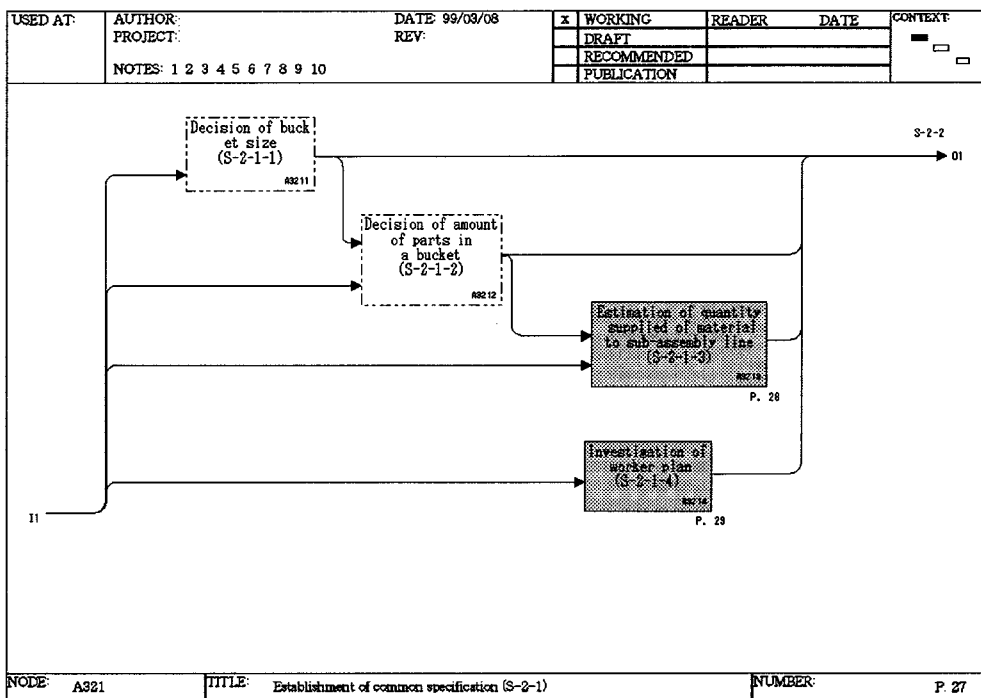


Fig. 3.14 MSE process for transportation system and line layout design (3)

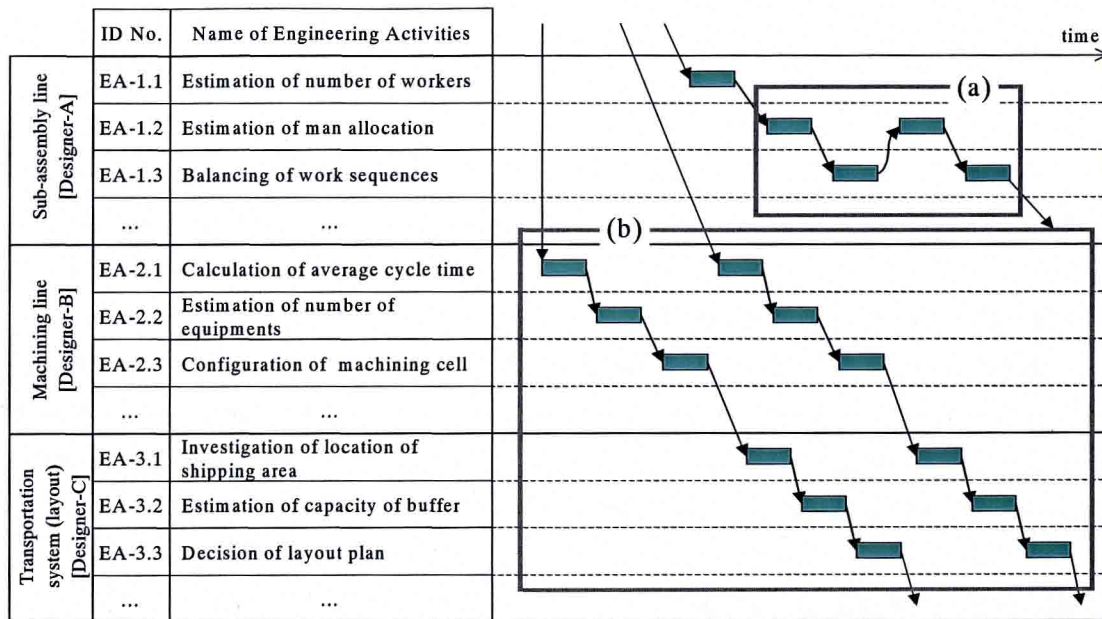


Fig. 3.15 Example of information exchange on MSE process of test case

EA 間における情報授受の関係の概略をFig. 3.15に示す。横軸は時間を、縦軸は各 EA を示す。図中の四角は個々の EA を、矢印は EA 間で授受される情報を表す。Fig. 3.15に示すように、MSE プロセスでは、異なる設計者が実行している EA 間で多くの情報が授受されている。また、設計変更、すなわち既に設計が終了した EA と同じ設計活動が繰り返し行われている。これには、以下の原因があると考える。

- (1) ある EA において、評価手法の条件を満たす出力情報を生成できない場合、入力情報として利用した他の EA の出力情報の値を変更する必要がある。そのため、Fig. 3.15 (a)のように、同一の EA (EA-1.2 および EA-1.3) が繰り返し行なわれている。
- (2) MSE プロセスの進行に伴い、精度の高い入力情報が得られるようになる。そのため、初期の EA で生成された出力情報の見直しが行われる。例えば、Fig. 3.15 (b)の場合、EA-2.1 の入力情報が時間の経過に伴い、精度が高くなるため、EA-2.1 から EA-3.3 までの EA が繰り返し行われている。
- (3) 入出力情報の転記ミスや、制約条件を考慮していないことがあった場合、同じ EA を再度、実行する必要がある。
- (4) 設計対象が同じであっても、その MSE プロセスに必要な時間は設計者の能力により異なる。例えば、熟練の設計者は、経験に基づき、重要な出力情報から決定することで、設計変更を少なくして

いる。

3.4.2 現状の MSE プロセスの問題点

具体的な生産システム設計における MSE プロセスの記述に基づいて、MSE プロセスを分析した結果、生産システムの設計期間の短縮化にとって、以下の問題が重要であることが明らかになった。

(1) 複数の設計者間での設計情報の授受

複数の設計者で設計を行う場合、設計者は、互いに出力情報を授受する。設計の終了や変更にあわせて迅速、かつ正確に情報を授受することは容易でない。また、生産システムモデル間の関係にもなう属性値間の制約条件は明示的に授受できない。設計者は、全ての制約条件を考慮して、情報の整合性を保持することが必要になる。

(2) 設計変更による出力情報の値の変更

複数の EA は、入出力情報の授受の関係による連鎖を形成する。そのため、ある EA での設計変更による出力情報の値の変更は、既に設計が終了した他の EA の出力情報に影響を及ぼす。設計変更があった場合、変更された情報の授受や出力情報の再評価が、MSE プロセスの広い範囲にわたって必要になるため、設計期間は長期化する。

(3) 事例の蓄積と利用

熟練の設計者は、経験に基づき、過去の設計事例から適切な生産システムモデルの属性値や設計活動モデルを参照して設計を行う。過去の設計事例を、設計者間で共有することは容易でない。そのため、経験の浅い設計者は、試行錯誤が多くなり、設計期間は長期化する。

3.5 結言

本章では、MSE プロセス支援ツールを開発するための準備として、実際の MSE プロセスの記述を行い、その分析を通して、MSE プロセスの問題点を明らかにした。以下に本章の内容をまとめる。

- (1) 第 2 章で提案した生産システムモデルを拡張して、MSE プロセスのモデル化手法を提案した。MSE プロセスのモデルは、生産システムの設計活動を表現する設計活動モデルと、設計対象となる生産システム全体またはその構成要素を表現する生産システムモデルの 2 種類のモデルが必要であることを示した。
- (2) MSE プロセスを記述するためのツールとして、Microsoft Word で作成した EA シートと、Design/IDEF の IDEF0 を用いる方法を提案した。EA シートを用いて、ひとつの EA に関する情

報を詳細に記述した上で、IDEF0を用いて、複数のEA間における情報授受の関係を記述する。

- (3) (2)の記述ツールを用いて、自動車の回転機器部品を製造する加工および組立工場の設計を行い、設計の進行と同時にそのMSEプロセスを記述した。その結果、約180個のEA、およびEA間における情報授受の関係を記述した。
- (4) (3)のMSEプロセスの記述を分析した結果、生産システムの設計期間の短縮化には、複数の設計者間での設計情報の授受、設計変更による出力情報の値の変更、および事例の蓄積と利用における問題を解決することが重要であることを明らかにした。

参考文献

- [FIPS 93] 第1章と同じ。
- [Hart 89] Hart, A., *Knowledge acquisition for expert systems*, McGraw-hill (1989);
神林 訳, エキスパートシステムのための知識獲得ハンドブック, 啓学出版 (1990).
- [IMS 委員会 80] 精機学会総合生産システム専門委員会 入力情報処理小委員会ワーキング・グループ, 自動化機械設計プロセスの現状, 精機学会総合生産システム専門委員会 昭和54年度活動報告書 (1980), 7-42.
- [Mertins 98] Mertins, K., Rabe, M. and Muller, W., Designing a computer-aided manufacturing systems engineering process, *Journal of material processing technology*, 76 (1998), 82-87.
- [Mertins 99] Mertins, K., and Jochem, R., *Quality-oriented design of business processes*, Kluwer academic publishers (1999).
- [荒井 98] 荒井・赤坂・若松・白瀬, CADにおける設計意図モデルと修正設計への適用, 日本機械学会論文集, 64-627, C (1998), 384-389.
- [石村 93] 石村, 生産システムのFA化設計, 日刊工業新聞社 (1993).
- [岡田 91] 岡田・荒井, 機械設計における設計意図モデルに基づくCAD, 日本機械学会論文集, 57-544, C (1991), 283-288.
- [菊地 95] 菊地・岸浪, 設計生産コミュニケーションの記述表現及び応用に関する研究(第1報)-基本概念及び形式仕様記述言語-, 精密工学会誌, 61-5 (1995), 712-716.
- [菊地 96] 菊地・岸浪, 設計生産コミュニケーションの記述表現及び応用に関する研究(第2報)-設計生産活動の統合モデル-, 精密工学会誌, 62-3 (1996), 388-392.

- [窪田 96] 窪田・田浦, エンジニアリング歴史ベースの研究, 精密工学会誌, 62-3 (1996), 377-382.
- [小島 96a] 小島・岩田・福田・松本, ファシリティのライフサイクルを考慮した生産モデルの設計手法に関する研究, 日本機械学会論文集, 62-597, C (1996), 410-417.
- [小島 96b] 小島, ファシリティのライフサイクルを考慮した生産モデルの設計・評価に関する研究, 大阪大学博士論文 (1996).
- [田浦 99] 田浦・他 7 名, グローバルデザインを支援する知的メディアの研究 -Activity Chain モデルの提案と応用-, 精密工学会誌, 65-2 (1999), 272-278.
- [仲 97a] 仲・小野里・岩田, 設計・生産業務の遂行における行為の分析とその構造化の枠組, 日本機械学会論文集, 63-616, C (1997), 332-338.
- [仲 97b] 仲・小野里・住岡・岩田, 設計・生産業務の遂行における行為の分析とその構造化の枠組, 日本機械学会論文集, 63-616, C (1997), 339-345.

第 4 章 生産システムの設計プロセス支援ツールの開発

4.1 緒言

生産システムの要求定義から詳細設計までの設計過程、すなわち MSE プロセス(Manufacturing System Engineering Process)を合理的かつ系統的に行うには、MSE プロセスにおける設計情報の記録、参照および適切な設計手順の指示などを、計算機を用いて統合的に支援するための MSE プロセス支援ツールが必要である。

第 3 章では、自動車の回転機器部品を製造する加工および組立工場の設計実験を行い、その設計の進行と同時に、MSE プロセスを記述した。さらに、それを分析して、MSE プロセスの問題点を明らかにした。

本章では、その問題点に基づき、MSE プロセス支援ツールのプロトタイプを開発するとともに、それを簡単な生産システムの設計例に適用して、有効性を検証する。

4.2 MSE プロセス支援のための要求機能

製造企業には、市場の要求を満たす製品を素早く製造し、迅速に市場へ供給することが求められる。そのため、製品の設計プロセスや生産プロセスだけでなく、MSE プロセスを短縮化し、製品の生産開始までの期間を短縮することが必要である。

第 3 章において、具体的な生産システム設計の MSE プロセスを分析した結果、生産システムの設計期間の短縮化にとって、複数の設計者間での設計情報の授受、設計変更による出力情報の値の変更、および事例の蓄積と利用の問題が重要であることが明らかになった。

4.2.1 既存の研究における MSE プロセスの支援

Mertins ら [Mertins 98] や、小島ら [小島 96a] [小島 96b] の研究では、MSE プロセスにおける情報授受の関係を IDEFO 表記のような図的形式で表現する。このような図的形式による記述は、情報授受の関係の全体を、設計者が視覚的に理解しやすいという利点がある。しかし、MSE プロセスが大規模になると、一目で全体の関係を理解するのは困難になる。そのため、ある設計情報の変更が影響を及ぼす EA の全てを迅速に特定することが困難になり、設計変更に長い時間を要するという問題がある。

荒井ら [荒井 98] [岡田 91] の開発した設計プロセス記述ツールでは、あらかじめ作成した生産対象製品の幾何形状モデルを修正した場合、生産対象製品の各部位に記述したコメントや評価式といった設計意図を考慮して、統合的な幾何形状モデルを自動的に生成することができる。窪田ら [窪田 96] は、生産対象製品とその設計プロセスの情報を記述する設計プロセス記述ツールを開発した。このツールでは、試行錯誤のプロセスも記述される。そのため、設計変更を検討する際に、試行錯誤のプロセスをたどることができる。また、田浦ら [田浦 99] は、他の設計者へ設計意図を伝達する手段として、ある設計者が利用したシミュレーションを、他の設計者が異なる計算機上からネットワークを介して再試行できる設計プロセス記述ツールを開発した。これら既存の研究における設計プロセス記述ツールは、記述した設計プロセスや設計意図を他の設計者が参照できるため、他の設計者が設計の経緯を理解するのに有効であると考えられる。しかし、設計変更が必要になった場合には、記述した設計プロセスや設計意図を、設計者が逐一参照して、どの情報を変更するのが最も適当であるかを判断しなければいけない。そのため、設計変更が必要な設計プロセスが大規模な場合には、変更する情報を迅速に選定することは困難となり、設計期間は長期化するという問題があると考えられる。

4.2.2 MSE プロセス支援ツールの基本機能

本章で開発する MSE プロセス支援ツールの基本機能を以下に示す。

まず、設計者が、MSE プロセスを十分に記述できる機能が必要である。さらに、設計変更に伴うエンジニアリングアクティビティ(以下、EA)の繰り返しを短縮するためのサポート機能が必要である。すなわち、ある EA の出力情報の値を変更しなければいけない場合に、同一の EA の繰り返しを減少させるためのサポート機能が必要である。

大規模な MSE プロセスでは、複数の EA が入出力情報の授受の関係による複雑な連鎖を形成する。そのため、ある EA での設計変更による出力情報の値の変更が、既に設計が終了した他の EA の出力情報に与える影響を認識することは容易でない。そこで、本章では、設計変更にともなう EA の繰り返しを短縮するためのサポート機能のひとつとして、まず、出力情報の設計変更が他の EA に及ぼす影響を迅速に調べる機能を準備する。

4.3 MSE プロセス支援ツールの開発

本章では、オブジェクト指向言語の Smalltalk を用いて、MSE プロセス支援ツールのプロトタイプを開発した。この支援ツールは、MSE プロセスを記述する機能と設計変更を支援する機能を持つ。

4.3.1 MSE プロセスの記述

Fig. 4.1に例示する入力インターフェースを用いて、第3章で示した EA の ID 番号と名称、設計者名、設計の開始および終了時刻、入出力情報、および評価手法を記述する。

入力情報は、初期仕様または他の EA の出力情報である生産システムモデルの属性値である。Fig. 4.2に例示するインタフェースは、他の EA の出力情報を検索し、表示する。設計者が、このインタフェースから選択した情報は、Fig. 4.1のインタフェースの入力情報へ自動的に挿入される。これにより、入力情報を出力情報と等しく設定することができ、情報の転記ミスを防止することができる。

評価手法として、設計者がルールやコメントを自由に記述することができる。また、出力情報の属性に関連する計算式をデータベースから選択することができる。

出力情報となる生産システムモデルの属性値は、Fig. 4.3のインタフェースを用いて決定する。第 2 章で示したモデルリポジトリを表すリストから、設計対象となる生産システムモデルを選択すると、その生産システムモデルの属性リストが表示される。設計者は、この EA で設計する属性を選択し、その属性値を決定する。その際、出力情報の属性に関連する計算式が評価手法としてデータベースに存在する場合には、その計算式と入力情報を用いて支援ツールが自動的に計算を行い、属性値を決定する。また、is-a や has-a など、モデルリポジトリにおける生産システムモデル間の関係に基づき、既に設計済みの生産システムモデルの中から、この EA で設計している生産システムモデルと整合すべき候補を検索する機能を持つ。設計者が、その候補の中から整合すべき生産システムモデルを特定すると、支援ツールは、その関係にもなう属性値間の制約条件を考慮して、生産システムモデル間の整合性を評価する。以上の手順で決定した生産システムモデルの属性値は、Fig. 4.1のインタフェースの出力情報に表示される。

The screenshot shows the 'Engineering Activity UI' window. At the top, there are buttons for 'Open AC', 'Save AC', 'New Model', and 'Delete Model'. Below these are input fields for 'ActivityID' (a02), 'Design Name' (組立ライン1の人工の決定), 'Designed Date' (2000.5.30), and 'Designer' (Mr.I). There are also 'Input Model' and 'Output Model' sections, each with a 'Refer To Model' button. The 'Input Model' list contains two entries: 'List (x01' 6 '組立ライン1' '総作業時間' 77)' and 'List (a01' 12 '組立ライン1' 'サイクルタイム')'. The 'Output Model' list contains one entry: 'List (a02' 13 '組立ライン1' '人工' 2)'. At the bottom, there is an 'evaluate' section with a text area containing the calculation formula: '計算式: (人工) = (総作業時間) / (サイクルタイム)'. The interface has a standard Windows-style window with a title bar and window controls.

Fig. 4.1 Interface to describe engineering activities

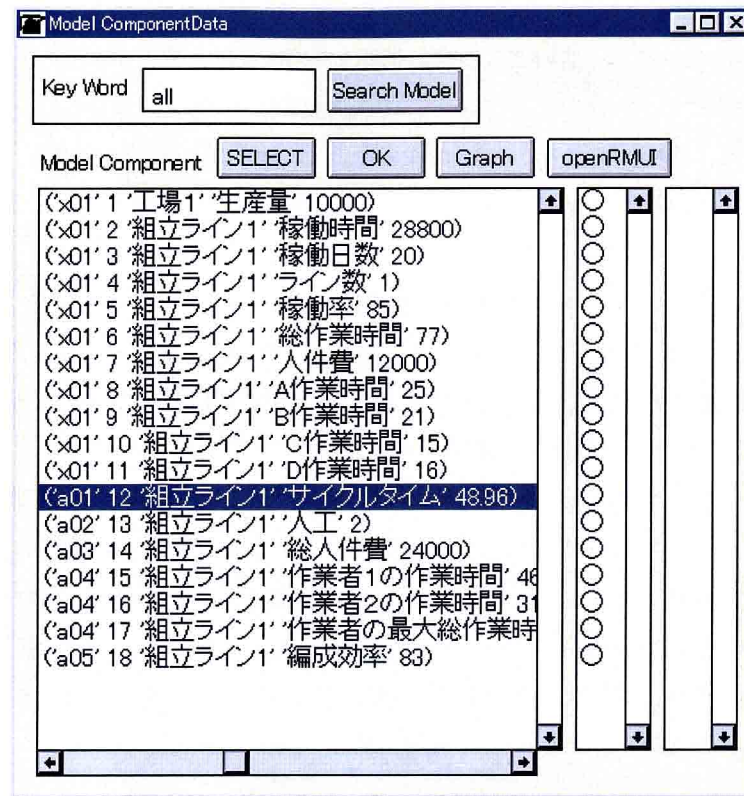


Fig. 4.2 Interface to search output data

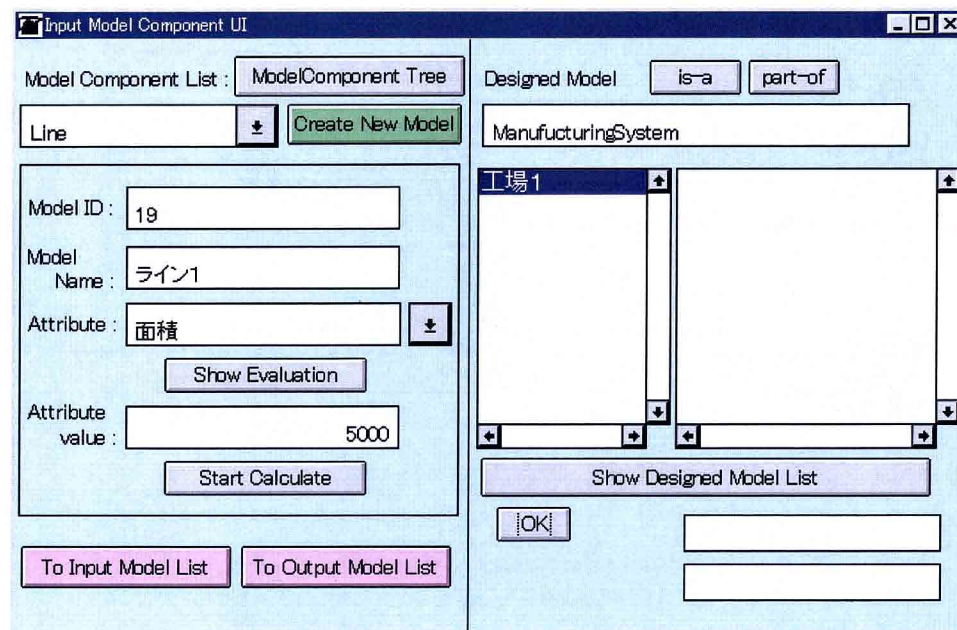


Fig. 4.3 Interface to describe output data

これらの EA の情報は全て、本支援ツールに保存される。これを全ての EA において繰り返すことで、MSE プロセスの全体を記述する。

なお、本支援ツールは、複数の設計者が、異なる計算機上からネットワークを介して、コンカレントに MSE プロセスを記述、参照することが可能である。

4.3.2 設計変更の支援

ある EA において、評価手法の条件を満たす出力情報を生成できない場合、入力情報として利用した他の EA の出力情報の値を変更しなければいけない。この場合、既に設計が終了した EA の出力情報に及ぼす影響が少ない情報を変更することが望ましいと考える。これにより、設計変更に伴う EA 間での情報の授受が少なくなるだけでなく、出力情報の再評価も少なくなるため、設計変更による設計期間の長期化を抑制することができる。

本章では、各 EA における情報の入出力関係に対して、ISM (Interpretive Structural Modeling) [Warfield 73] [赤木 91] の手法を適用し、ある EA の出力情報の値の変更が影響を与える他の EA の出力情報の総数を計算することで、出力情報の値の変更が MSE プロセス全般にわたって及ぼす影響を求める。以下に、その手順を示す。

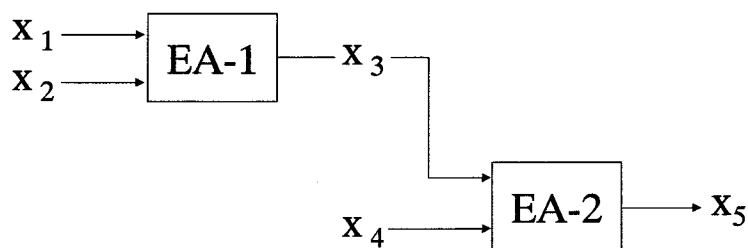
STEP1: 2 値行列 A の作成

まず、MSE プロセスにおいて、EA 間で授受される全ての情報に ID 番号を割り振り、それら全てを A の行と列に配置する。次に、EA 毎に入力情報と出力情報を見て、入力情報の ID 番号がある行と、出力情報の ID 番号がある列の行列成分を 1 にする。すなわち、ある EA において、情報 i が入力情報、情報 j が出力情報になっている場合、 A の (i, j) 成分を $a_{ij} = 1$ とする。これを全ての EA に対して行い、2 値行列 A を作成する。すなわち、 a_{ij} は、以下のように表せる。

$$a_{ij} = \begin{cases} 1 & (\text{入力情報 } i \text{ を持つ EA において出力情報 } j \text{ がある場合}) \\ 0 & (\text{入力情報 } i \text{ を持つ EA において出力情報 } j \text{ がない場合}) \end{cases} \quad (4.1)$$

$(i, j=1, 2, \dots, n)$

例えば、Fig. 4.4 に示す MSE プロセスの例では、 A は以下のようになる。



EA-i : Engineering activity
 X_j : Input and/or output data
 (Manufacturing system model)

Fig. 4.4 Example of MSE processes

$$A = \begin{matrix} & \begin{matrix} X_1 & X_2 & X_3 & X_4 & X_5 \end{matrix} \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (4.2)$$

STEP2: 隣接行列 B の作成

A に単位行列 I を加えて, 隣接行列 B を作成する. なお, 行列の各要素の計算には, ブール演算を用いる.

$$B = A + I \quad (4.3)$$

ブール演算による 2 進数の和は以下のとおりである.

$$\left. \begin{matrix} 0 + 0 = 0 \\ 1 + 0 = 1 \\ 0 + 1 = 1 \\ 1 + 1 = 1 \end{matrix} \right\} \quad (4.4)$$

よって, Fig. 4.4 の例では, B は以下のようになる.

$$B = A + I = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (4.5)$$

STEP3: 可到達行列 T の作成

ブール演算に基づき, B^2, B^3, \dots と B を掛け合わせていくと, 行列の要素の値が変化しなくなる. すなわち, $B^{p-1} = B^p$ となる. その B^p を可到達行列 T と呼ぶ.

$$B \neq B^2 \neq \dots \neq B^{p-1} = B^p \equiv T \quad (4.6)$$

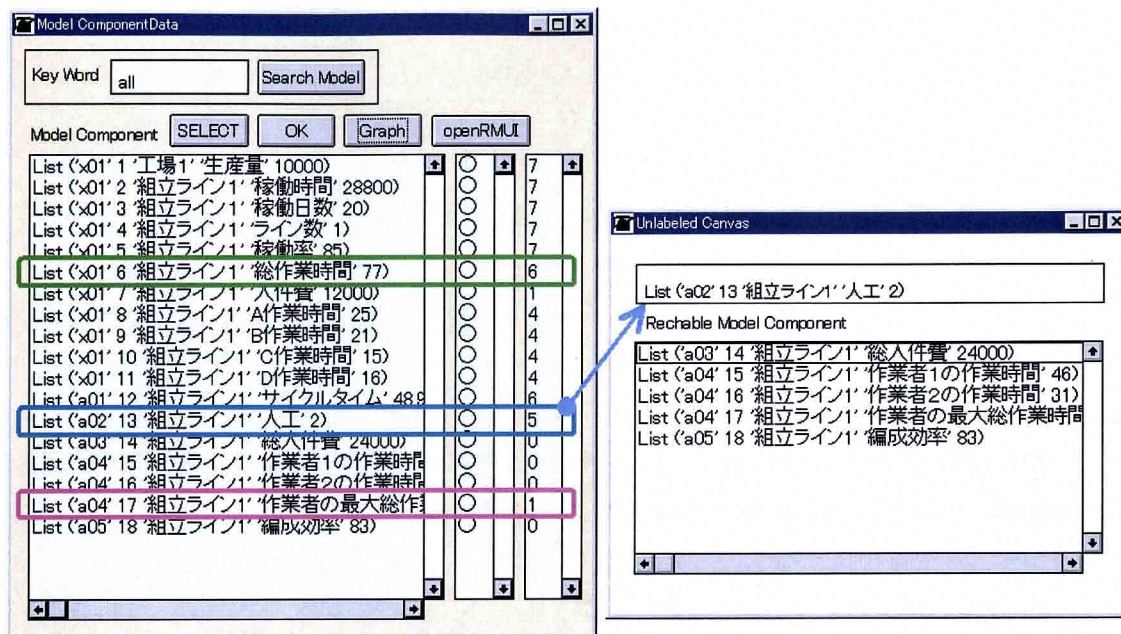
ブール演算による 2 進数の積は以下のとおりである.

$$\left. \begin{array}{l} 0 \cdot 0 = 0 \\ 1 \cdot 0 = 0 \\ 0 \cdot 1 = 0 \\ 1 \cdot 1 = 1 \end{array} \right\} \quad (4.7)$$

Fig. 4.4 の例で B の累乗を計算すると以下のようになる.

$$B^2 = B^3 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \equiv T \end{matrix} \quad (4.8)$$

T の i 行 j 列目に 1 がある場合, i 行目の情報の変更により, j 列目の情報が影響を受けることを表す. T には, B に含まれていなかった 1 が表れている. これは, EA 間における情報授受の連鎖により間接的に影響を受ける情報を表す. これにより, i 行目の列集合を見れば, MSE プロセス全体において, 情報 i の変更により影響を受ける全ての情報がわかる.



(a) Total number of output data affected by re-design (b) List of all output data affected by re-design

Fig. 4.5 Effects of designed data

STEP4: 影響を受ける出力情報の総数 r の計算

i 行目の情報の変更により影響を受ける情報の総数 r_i を以下の式で計算する.

$$r_i = a_{i1} + a_{i2} + \dots + a_{in} - 1 \quad (i=1,2,\dots,n) \quad (4.9)$$

Fig. 4.4の例の場合, x_1 から x_5 までの各情報の変更により影響を与える情報の総数 $r_i (i=1,2,\dots,5)$ は T より以下のようになる.

$$(r_1, r_2, r_3, r_4, r_5) = (2, 2, 1, 1, 1) \quad (4.10)$$

例えば, 評価手法の条件を満たすことができない EA が複数の入力情報を持つ場合には, この値 r が小さく, かつ, 他の設計者が影響を受けない情報から設計変更を検討すればよい.

Fig. 4.5(a) のインターフェースは, 設計変更が影響を及ぼす出力情報の総数を計算した結果の一例を示す. Fig. 4.5 (b) のインターフェースは, 影響を受ける全ての出力情報を表示する. 設計者は, これらを参照して, できるだけ影響の小さい情報から設計変更を検討することが可能になる.

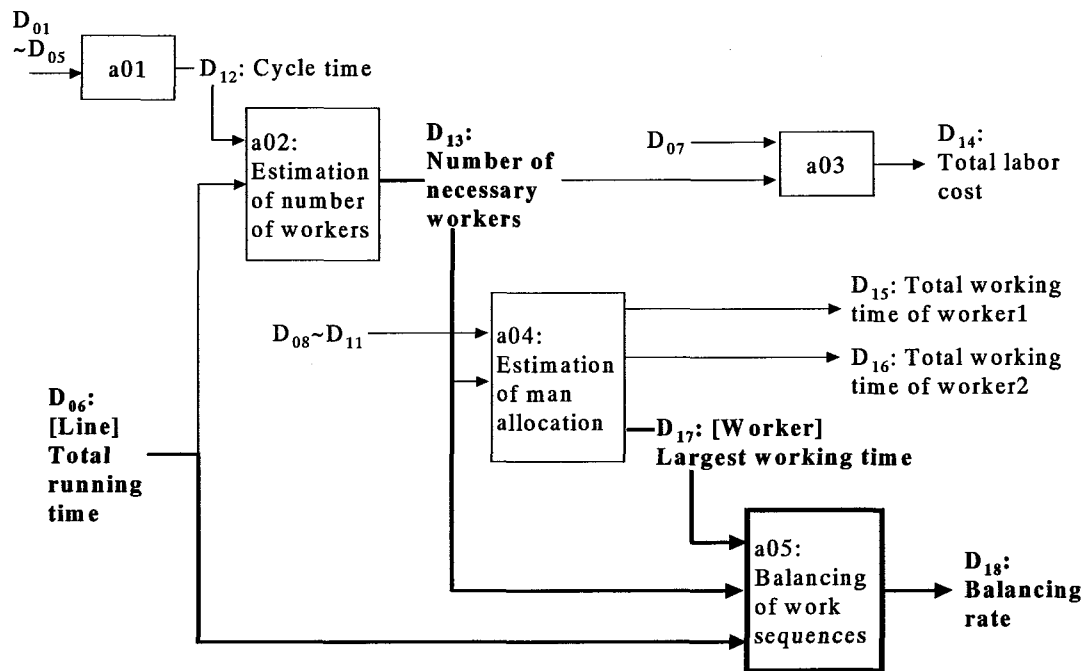


Fig. 4.6 MSE process for case study

Table 4.1 Initial data for case study

Number of data	Object name	Attribute	Value
D ₀₁	Factory 1	Production volume	10000 (pcs/M)
D ₀₂	Assembly line 1	Operation time	28800 (second)
D ₀₃	Assembly line 1	Operation days	20 (days)
D ₀₄	Assembly line 1	Number of lines	1
D ₀₅	Assembly line 1	Operation rate	85 (%)
D ₀₆	Assembly line 1	Total running time	77 (second)
D ₀₇	Assembly line 1	Labor cost	12000 (yen/person)
D ₀₈	Assembly process 1	Operation time	25 (second)
D ₀₉	Assembly process 2	Operation time	21 (second)
D ₁₀	Assembly process 3	Operation time	15 (second)
D ₁₁	Assembly process 4	Operation time	16 (second)

4.4 ケーススタディ

Fig. 4.6に示す MSE プロセスの例に対して、この支援ツールを適用した。まず、MSE プロセスの開始時に与えられる初期仕様として与える情報をTable 4.1に示す。

4.3節で示したFig. 4.1, Fig. 4.2およびFig. 4.3のインタフェースを用いて、5つのEAにおける設計を行った。まず、Fig. 4.1のインタフェースを用いて、EAのID番号、設計者名および設計の開始日

を入力する。次に、Fig. 4.3のインタフェースを開いて、この EA で設計する生産システムモデルとその属性リストを選択する。データベースから、出力情報の属性に関連する計算式を評価手法として引用する。その計算式で必要な情報をFig. 4.2のインタフェースを用いて検索し、入力情報として引用する。これにより、支援ツールが自動的に計算を行い、属性値を決定するとともに、Fig. 4.1のインタフェースの出力情報に記述する。例えば、1 つ目の EA (a01)の場合、生産システムモデルとして、組立ライン (AssemblyLine)を、その属性としてサイクルタイムを選択する。次に、サイクルタイムを含む計算式をデータベースから引用し、以下の式が得られる。

$$(\text{サイクルタイム})=(\text{稼働時間})\times(\text{稼働日数})\times(\text{稼働日数})\div(\text{生産量})\times(\text{稼働率}) \quad (4.11)$$

そこで必要となる情報、すなわち、組立ライン 1 の稼働時間、稼働日数、稼働日数、稼働率および工場 1 の生産量をFig. 4.2のインタフェースを用いて検索し、入力情報として引用する。支援ツールに計算を行わせた結果、サイクルタイムの値 49 秒が得られ、これを出力情報 D₁₂としてFig. 4.1のインタフェースの出力情報に記述し、保存する。

5 つ目の EA (a05)において、出力情報 D₁₈(編成効率)が評価手法の要求を満たすことができない。そこで、3つの入力情報に対して、4.3.2項で示した方法を適用し、設計変更が及ぼす影響を計算した。その結果、Fig. 4.5(a)に示すように、入力情報 D₁₇(作業者の最大作業時間)が他の EA の出力情報に及ぼす影響が最も小さいことがわかった。そこで、D₁₇ の値を決定した EA (a04)において、この値の変更を検討し、各作業者の作業工程の割当てを変更した。これにより、D₁₈ は評価手法の要求を満たすことができた。なお、出力情報の変更が他の出力情報に及ぼす影響関係をFig. 4.7に示す。これよりわかるように、D₁₇の変更は、D₁₈以外に影響を及ぼさないため、設計変更の作業を終了することができた。

4.5 結言

本章では、第 3 章で行った生産システムの設計実験における MSE プロセスの記述とその分析から得られたMSEプロセスの問題点に基づき、MSE プロセス支援のための要求機能を整理し、MSE プロセス支援ツールのプロトタイプを開発した。以下に本章の内容をまとめる。

- (1) 第 3 章で得られた MSE プロセスの問題点に基づき、MSE プロセス支援ツールの基本機能として、MSE プロセスを記述する機能と、設計変更を支援する機能が重要であることを示した。
- (2) オブジェクト指向言語の Smalltalk を用いて、MSE プロセス支援ツールのプロトタイプを開発した。この支援ツールは、(1)で示した 2 つの基本機能を持つ。
- (3) MSE プロセスを記述する機能においては、第 3 章で提案した MSE プロセスのモデルに基づき、MSE プロセスの情報を記述する。さらに、複数の設計者が、異なる計算機上からネットワークを介して、コンカレントに MSE プロセスを記述、参照することができる。また、他の EA の出力情報を検索

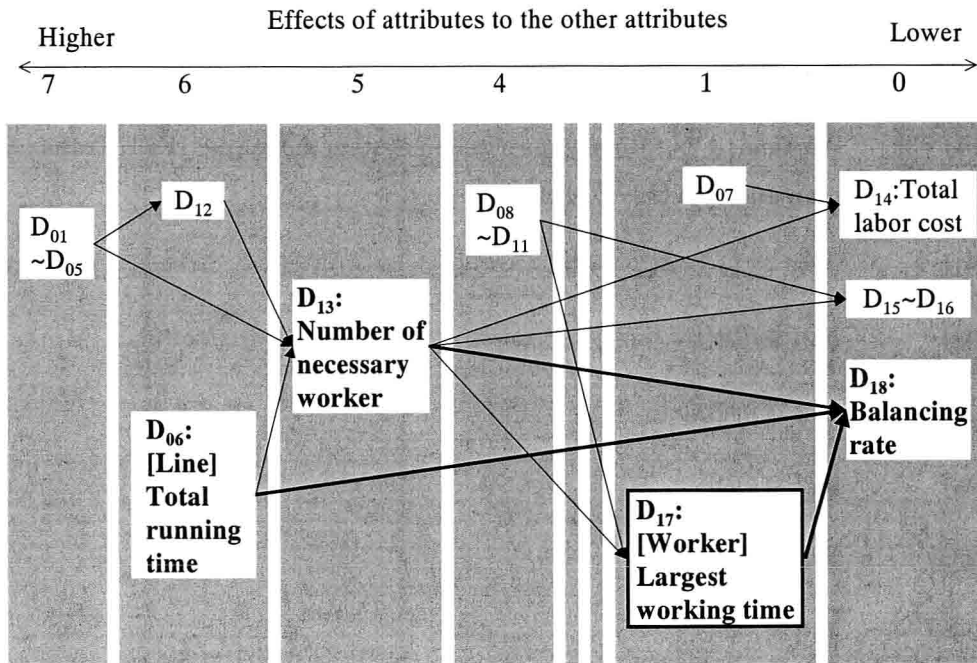


Fig. 4.7 Output data affected by re-design

する機能や、出力情報の属性値を自動的に計算する機能などにより、設計者の設計活動を支援することができる。

- (4) 設計変更を支援する機能においては、ISMの手法を適用して、あるEAの出力情報の値を変更した結果がMSEプロセス全般にわたって及ぼす影響を求める方法を提案した。これにより、設計変更の際に、既に設計が終了したEAの出力情報に及ぼす影響が少ない情報を変更することができ、設計変更による設計期間の長期化を抑制することができる。
- (5) 開発したMSEプロセス支援ツールを、簡単な生産システムの設計例に適用し、その有効性を検討した。設計変更の及ぼす影響が小さい情報から設計変更を検討することで、設計変更による設計期間の長期化を抑制できることを示した。

なお、本章で示したMSEプロセス支援ツールは、製造企業で生産システムの設計に関与している技術者の設計プロセスを記述、分析することで得られた要求機能に基づいて開発されている。そのため、製造企業で生産システムを設計する際に、本支援ツールを利用することは十分に可能であると考えられる。

参考文献

- [Mertins 98] 第 3 章と同じ.
- [Warfield 73] Warfield, J. N., Binary matrices in system modeling, IEEE Trans., System man and cyb. (1973), 441-449.
- [赤木 91] 赤木, 設計工学(上)－新しいコンピュータ応用設計－, コロナ社 (1991), 30-38.
- [荒井 98] 第 3 章と同じ.
- [岡田 91] 第 3 章と同じ.
- [窪田 96] 第 3 章と同じ.
- [小島 96a] 第 3 章と同じ.
- [小島 96b] 第 3 章と同じ.
- [田浦 99] 第 3 章と同じ.
- [谷岡 99] 谷岡 ほか 47 名, IMS9810 グローバル分散企業の設計・計画及び経営の為のモデリングとシミュレーション環境に関する研究成果報告書, IMS センター (1999).

第 5 章 生産システムの運用のためのモデル

5.1 緒言

本論文の目的は、オブジェクト指向技術を用いて、生産システムの設計から運用までの過程における複数のアプリケーションを分散環境でコンカレントに実行するための方法を提案することである。第 3 章および第 4 章では、生産システムの設計フェーズを研究の対象とし、生産システムの設計実験における設計プロセスの分析を行うとともに、生産システムの設計プロセス(MSE プロセス)支援ツールを開発した。設計フェーズで作成された生産システムモデルは、生産システムの運用フェーズにおいて利用されると考える。そこで、第 5 章以降では、生産システムの運用フェーズにおける生産システムモデルとその利用手法について考察する。本論文では、モデル化の対象を生産スケジューリング (Production scheduling) に絞って考察する。

生産スケジューリングに関する研究や開発は、遺伝的アルゴリズム(Genetic algorithm) [西川 91] [山田 93] やラグランジュ緩和法(Lagrangian relaxation) [黒田 98] [米田 97] などの解法に関する研究から、市販のソフトウェア開発 [錦戸 93] まで、様々に行われてきた。これによって、ロットや部品の生産ラインへの投入順序の決定における意思決定の効率化や最適化が実現されている。しかし、生産システムの運用フェーズには、工場レベルから生産設備レベルまで、複数の生産スケジュールが階層的に存在する。それらの生産スケジュールの作成は、一般に、トップダウン的に行われる。また、各階層における意思決定は、異なる人または部門で行われる。そのため、最終的に生産設備レベルの生産スケジュールを得るまでに時間がかかるだけでなく、生産スケジュールの変更が困難である等の問題がある。そこで、各部門での意思決定をできるだけコンカレントに行うことが望まれている。

複数の意思決定をコンカレントに行う方法として、他部門の意思決定に必要な情報をできるだけ早く提供する方法や、他部門で必要とされるであろう情報をあらかじめ準備しておく方法がある [岩田 96]。そのためには、生産スケジュールの作成者が、部門全体にわたる生産システムモデルの構造をあらかじめ理解していることが必要である。すなわち、生産スケジューリングにおいて作成する生産システムモデルを、明確に構造化することが必要である。

第 2 章で示したように、生産システムモデルの表現形式は、モデルの利用目的によって異なる。生産スケジューリングにおいては、工場レベルや生産設備レベルなどの階層の違いにより、モデルの構成要素の時間的、空間的な大きさが異なる。そこで、本章では、第 2 章で提案した生産システムモデルの表現方法に基づき、生産スケジューリングのためのモデルの基本的な構成要素を提案するとともに、そのモデルを用いて、階層的な複数の生産スケジュール間の整合性を管理する方法を提案する。さらに、オ

プロジェクト指向言語の Smalltalk を用いて、生産スケジューリングのためのモデルリポジトリとモデリングシステムのプロトタイプを開発し、簡単な例を用いて、生産スケジュールの変更に対する整合性管理の過程の一例を示す。

5.2 階層分散型生産システムのスケジューリングにおける問題点

5.2.1 生産スケジュールモデルの内容

本章では、生産スケジューリングに必要な生産システムモデルを、生産スケジュールモデル (Production schedule model) と呼ぶ。また、部品や組立品などの生産対象をジョブ (Job)、ジョブの各生産工程を実行することができるセルや工作機械などの生産資源をリソース (Resource)、リソースを用いてジョブの生産工程のひとつを実行する作業をオペレーション (Operation) と呼ぶ。生産スケジューリングとは、ジョブの生産工程と、リソースで生産工程を実行するために必要な生産時間の情報に基づき、リソースにおいてジョブを処理する順序と、リソースでジョブを処理する作業の開始時刻および終了時刻を決定することである。すなわち、生産スケジュールモデルの内容は、以下のように整理することができる。

(1) ジョブにおけるリソースの処理順序

各ジョブの生産工程をリソースの順序の形式で表現したものである。これは、生産工程設計により決定され、生産スケジューリングへの入力となる。

(2) オペレーションの処理時間

あるリソースを用いて、あるジョブのひとつの工程を処理する時間である。これは、生産工程設計により決定され、生産スケジューリングへの入力となる。

(3) リソースにおけるジョブの処理順序

各リソースにおけるジョブの処理順序を表す。これは生産スケジューリングにより決定される。

(4) オペレーションの開始時刻および終了時刻

各リソースで実行するオペレーションの開始時刻と終了時刻を表す。これは、生産スケジューリングにより決定される。

5.2.2 生産スケジュール間の整合性管理

生産スケジューリングでは、一般に、工場、ライン、およびショップの生産スケジュールなど、生産スケ

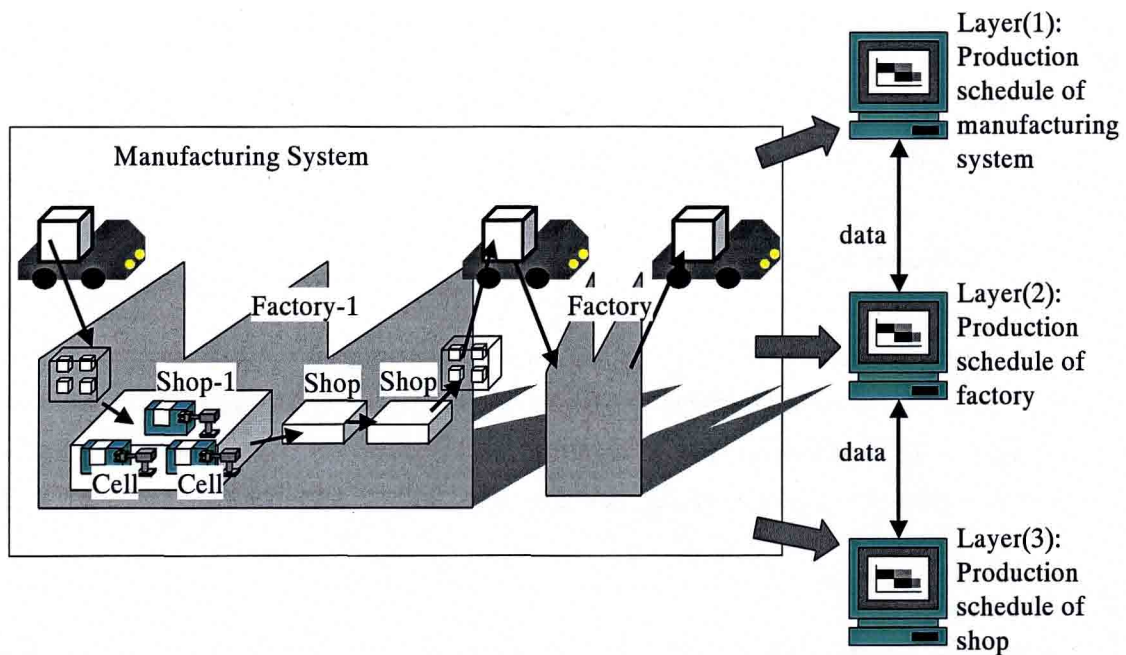


Fig. 5.1 Schedules of hierarchical and decentralized manufacturing system

ジュールモデルの構成要素の時間的、空間的な大きさが異なる複数のモデルが階層的に存在する。この階層的なモデルに対する意思決定が、異なる人や部門に分散している生産システムを、本章では、階層分散型生産システムと呼ぶ。階層分散型生産システムの生産スケジューリングでは、全ての生産スケジュール間の整合性を保持しながら、生産スケジューリングを行う必要がある。

例えば、Fig. 5.1では、ひとつの生産システムに対して、階層的な3つの生産スケジュールが存在している。階層1では、工場をリソースとする生産スケジュールを、階層2では、工場の中のショップをリソースとする生産スケジュールを、階層3では、ショップの中のセルをリソースとする生産スケジュールを作成している。これらの階層的な3つの生産スケジュールは、必ず整合していなければならない。そのため、階層分散型生産システムにおける生産スケジュールモデルの定式化および構造化が必要である。

5.3 生産スケジュールモデルの構成要素

5.3.1 単一の生産スケジュールモデルの表現

第2章では、オブジェクト指向の考え方にに基づき、生産システムモデルの原型を集めたモデルリポジ

トリを提案した。本章では、単一の生産スケジュールモデルを表現するために、モデルリポジトリにおけるオブジェクトの表現方法に基づき、以下の4つのオブジェクトを提案する。

(1) ジョブ

ジョブは、自身に作業を施すリソースの処理順序の情報を持つ。すなわち、ジョブ Jb_i がリソース Rs_1, Rs_2, \dots, Rs_m の順で作業を施される場合、ジョブのオブジェクトを次の形で表現する。

$$(Jb_i (SeqRs (Rs_1, Rs_2, \dots, Rs_m))) \quad (5.1)$$

ここで、 $SeqRs (Rs_1, Rs_2, \dots, Rs_m)$ は、リソースの順序を表すリストである。なお、本章では、1つのジョブは1組のリソースの順序を持つものとする。すなわち、ジョブの各工程を処理するリソースは決まっており、代替のリソースは存在しないものとする。また、リソースの順序関係は全順序関係であり、同一のリソースを2度以上、使用しないものとする。

(2) リソース

リソースは、ジョブの処理順序の情報を持つ。すなわち、リソース Rs_j がジョブ Jb_1, Jb_2, \dots, Jb_n の順で作業を施す場合、リソースのオブジェクトを次の形で表現する。

$$(Rs_j (SeqJb (Jb_1, Jb_2, \dots, Jb_n))) \quad (5.2)$$

ここで、 $SeqJb (Jb_1, Jb_2, \dots, Jb_n)$ は、ジョブの順序を表すリストである。

(3) オペレーション

オペレーションは、対象となるジョブ Jb_i 、実行するリソース Rs_j 、作業の開始時刻 st_{ij} 、終了時刻 ft_{ij} 、処理時間 pt_{ij} の情報を持つ。したがって、オペレーションのオブジェクトは、次のようになる。

$$(Op_{ij} (j Jb_i)(r Rs_j)(st st_{ij})(ft ft_{ij})(pt pt_{ij})) \quad (5.3)$$

オペレーションの開始時刻、終了時刻および処理時間の間には、以下の関係が存在する。

$$st_{ij} + pt_{ij} = ft_{ij} \quad (5.4)$$

この属性値間の関係を維持する手順をメソッドで表現する。

(4) オペレーションの順序関係

オペレーションの順序関係は、ジョブにおけるリソースの処理順序、およびリソースにおけるジョブ

の処理順序から生成する. すなわち, Jb_i において $Rs_1 < Rs_2$ ならば $Op_{i1} < Op_{i2}$ が, また Rs_j において $Jb_1 < Jb_2$ ならば, $Op_{1j} < Op_{2j}$ の順序関係を導出する. ここで, $A < B$ は, A が B に優先することを示す. このオペレーションの順序関係を次のオブジェクトで表す.

$$(\text{SeqOp}_{ij} (\text{prd} (Op_{ij1}, Op_{ij2}, \dots, Op_{ijk})) (\text{suc} Op_{ij})) \quad (5.5)$$

ここで, $\text{prd} (Op_{ij1}, Op_{ij2}, \dots, Op_{ijk})$ は, Op_{ij} の前に実行されるオペレーションの集合を示す.

先行するオペレーション $Op_{ij1}, Op_{ij2}, \dots, Op_{ijk}$ と後続のオペレーション Op_{ij} の終了時刻と開始時刻の間には, 以下の関係が存在する.

$$\max(ft_{ij1}, ft_{ij2}, \dots, ft_{ijk}) = st_{ij} \quad (5.6)$$

ここで, $\max(ft_{ij1}, ft_{ij2}, \dots, ft_{ijk})$ は, オペレーション $Op_{ij1}, Op_{ij2}, \dots, Op_{ijk}$ の終了時刻の最大値を示す. この属性値間の関係を維持する手順をオペレーションの順序関係のメソッドで表現する.

以上の 4 つのオブジェクトにより, 単一の生産スケジュールモデルを表現する.

5.3.2 複数の生産スケジュールモデル間の関係の表現

生産スケジューリングは, 長期の生産スケジュールから短期の生産スケジュールまでが階層的に作成される. さらに, 異なる人や部門においてコンカレントに作成されることが多い. Fig. 5.2は, 3 つの生産スケジュールを階層的かつコンカレントに作成する例を示す. この例では, 生産スケジュールの作成者 A は, ショップをリソースと, ショップ間を流れるロットをジョブと考慮して, 各ショップにおけるロットの投入順序を決定するための生産スケジュールモデルを作成する. 他方, 生産スケジュールの作成者 B および C は, ショップの構成要素であるセルをリソースと, ロットの構成要素であるワークをジョブと考慮して, 各セルにおけるジョブの処理順序を決定するための生産スケジュールモデルを作成する.

上記の例のように, 生産スケジュールの対象により, モデルの構成要素の時間的, 空間的な大きさが異なる. すなわち, 第 2 章で提案したモデルの分類基準において, 範囲の異なるモデルを取り扱うことになる. このような複数の生産スケジュールモデルを統合的に作成, 管理するためには, モデルの構成要素間に存在する包含関係を表現することが必要になる. このため, 以下の 3 つのオブジェクトを提案する.

(1) ジョブの包含関係

ジョブ間の包含関係の情報を持つ. ジョブ Jb_x が, より詳細なジョブ Jb_1, Jb_2, \dots, Jb_n に分解できる場合, すなわち, $\{Jb_x\} \supseteq \{Jb_1, Jb_2, \dots, Jb_n\}$ の包含関係が存在する場合, この包含関係を以下のオブジェクトで表す.

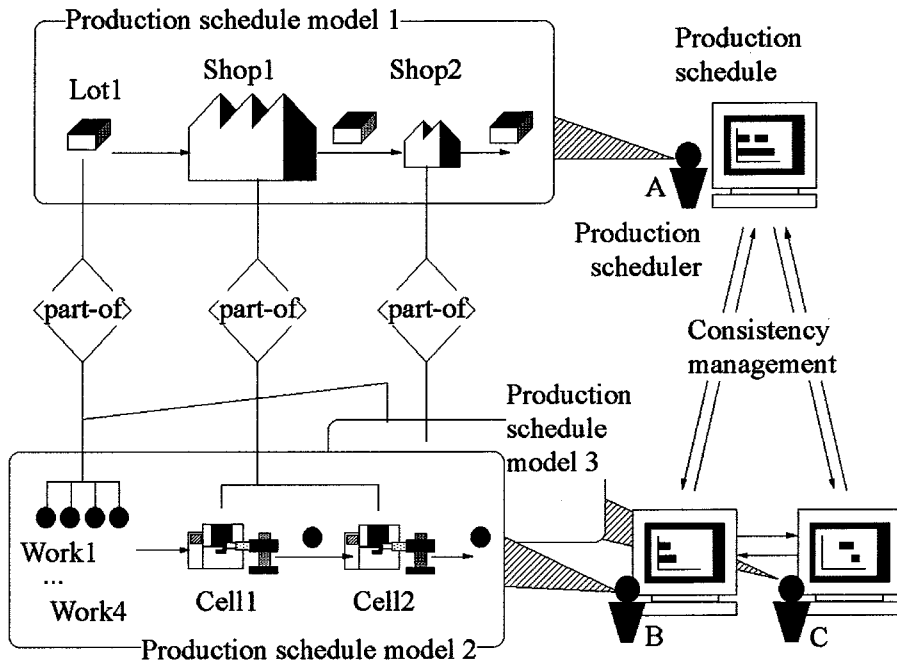


Fig. 5.2 Parallel and hierarchical production planning

$$(\text{IncJb}_x (\text{whole Jb}_x)(\text{part (Jb}_1, \text{Jb}_2, \dots, \text{Jb}_n))) \quad (5.7)$$

ここで、 $\text{part (Jb}_1, \text{Jb}_2, \dots, \text{Jb}_n)$ は、 Jb_x に包含されるジョブの集合を示す。

(2) リソースの包含関係

リソース間の包含関係の情報を持つ。上位のリソース Rs_y が、下位のリソース $\text{Rs}_1, \text{Rs}_2, \dots, \text{Rs}_m$ から構成される場合、すなわち、 $\{\text{Rs}_y\} \supseteq \{\text{Rs}_1, \text{Rs}_2, \dots, \text{Rs}_m\}$ の包含関係が存在する場合、この包含関係を以下のオブジェクトで表す。

$$(\text{IncRs}_y (\text{whole Rs}_y)(\text{part (Rs}_1, \text{Rs}_2, \dots, \text{Rs}_m))) \quad (5.8)$$

ここで、 $\text{part (Rs}_1, \text{Rs}_2, \dots, \text{Rs}_m)$ は、 Rs_y に包含されるリソースの集合を示す。

これら 2 つの包含関係の情報から、次に示すオペレーションの包含関係を表すオブジェクトを導き出すことができる。

(3) オペレーションの包含関係

オペレーション間の包含関係の情報を持つ。ジョブ Jb_x およびジョブ $\text{Jb}_1, \text{Jb}_2, \dots, \text{Jb}_n$ の間に、 $\{\text{Jb}_x\} \supseteq \{\text{Jb}_1, \text{Jb}_2, \dots, \text{Jb}_n\}$ の包含関係が存在し、リソース Rs_y およびリソース $\text{Rs}_1, \text{Rs}_2, \dots, \text{Rs}_m$ の間

に, $\{Rs_y\} \supseteq \{Rs_1, Rs_2, \dots, Rs_m\}$ の包含関係が存在する場合, オペレーション Op_{xy} およびオペレーション $Op_{11}, Op_{12}, \dots, Op_{nm}$ の間には, $\{Op_{xy}\} \supseteq \{Op_{11}, Op_{12}, \dots, Op_{nm}\}$ の包含関係が存在する. オペレーションの包含関係を表現するオブジェクトは, 以下のようになる.

$$(\text{IncOp}_{xy}(\text{whole } Op_{xy})(\text{part}(Op_{11}, Op_{12}, \dots, Op_{nm}))) \quad (5.9)$$

ここで, $\text{part}(Op_{11}, Op_{12}, \dots, Op_{nm})$ は, Op_{xy} に包含されるオペレーションの集合を示す.

包含関係にあるオペレーション Op_{xy} とオペレーション $Op_{11}, Op_{12}, \dots, Op_{nm}$ の開始時刻と終了時刻の間には, 次式の関係が存在する.

$$st_{xy} = \min(st_{11}, st_{12}, \dots, st_{nm}) \quad (5.10)$$

$$ft_{xy} = \max(ft_{11}, ft_{12}, \dots, ft_{nm}) \quad (5.11)$$

ここで, $\min(st_{11}, st_{12}, \dots, st_{nm})$ は, オペレーション $Op_{11}, Op_{12}, \dots, Op_{nm}$ の開始時刻の最小値を, $\max(ft_{11}, ft_{12}, \dots, ft_{nm})$ は, オペレーション $Op_{11}, Op_{12}, \dots, Op_{nm}$ の終了時刻の最大値を示す.

また, 式(5.4)より次式が得られる.

$$pt_{xy} = ft_{xy} - st_{xy} \quad (5.12)$$

よって, 式(5.10), (5.11)および(5.12)より, 次の関係式が得られる.

$$pt_{xy} = \max(ft_{11}, ft_{12}, \dots, ft_{nm}) - \min(st_{11}, st_{12}, \dots, st_{nm}) \quad (5.13)$$

以上, 式(5.10), (5.11)および(5.13)で示される属性値間関係を維持する手順を, オペレーションの包含関係のメソッドで表現する.

5.4 生産スケジュール間の整合性管理の手順

5.4.1 単一の生産スケジュールの作成

5.3節で提案したオブジェクトに基づいて生産スケジュールモデルを作成し, そのモデルを用いて, 工場, ショップ, またはセルなどを対象とする単一の生産スケジュールを作成する手順を以下に示す.

STEP1: ジョブ, リソースおよびオペレーションの入力

生産スケジュールの作成者が, ジョブ, リソースおよびオペレーションを表現するオブジェクトを生成する. さらに, ジョブにはその生産工程を実行するリソースの順序を, オペレーションにはその処

理時間を入力する。

STEP2: リソースにおけるジョブの処理順序の決定

生産スケジュールの作成者が、各リソースにおけるジョブの実行順序を決定する。この情報とジョブが持つリソースの順序から、オペレーションの順序関係を表すオブジェクトが生成される。

STEP3: オペレーションの開始時刻または終了時刻の決定

オペレーションの順序関係に基づいて、各オペレーションの開始時刻と終了時刻を生産スケジュールの作成者が決定する。この際、オペレーションの開始時刻を定めれば、式(5.4)によりその終了時刻が自動的に求められる。また、オペレーション間の順序関係に基づき式(5.6)を用いることで、後続するオペレーションの開始時刻を自動的に決定することができる。

5.4.2 複数の生産スケジュール間の整合

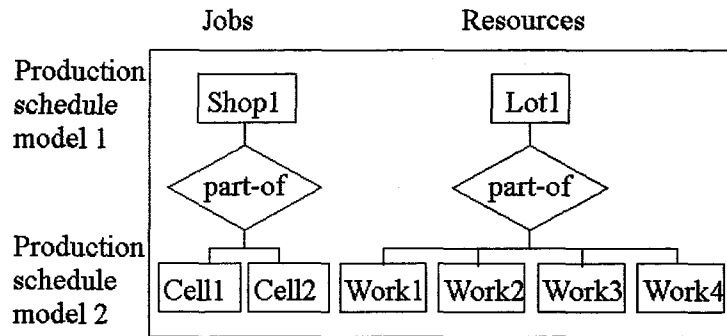
階層分散型生産システムにおいて、5.4.1項に示した手順で、工場、ショップ、セルなどの生産スケジュールをコンカレントに作成する場合、それらの生産スケジュール間の整合性を管理することが必要になる。そこで、5.3節で提案したオブジェクト間の関係式に基づき、階層的な複数の生産スケジュールモデル間の整合性を管理することで、生産スケジュール間の整合性を管理する手順を以下に示す。

STEP1: ジョブおよびリソースの包含関係の作成

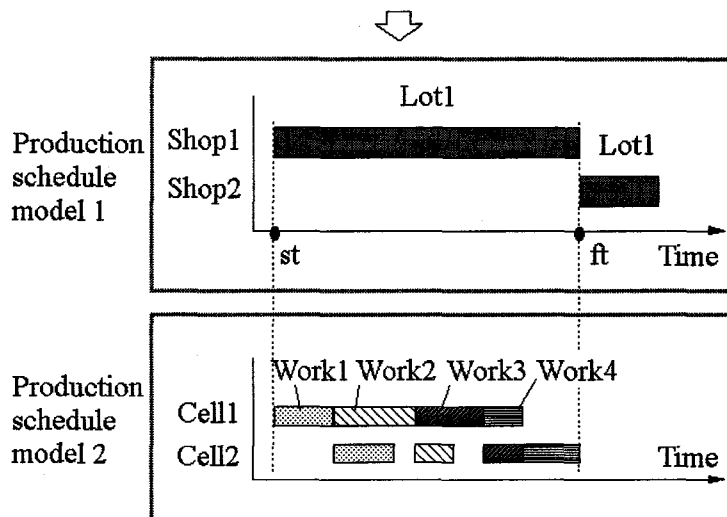
生産スケジューリングが、複数の階層でコンカレントに行われている場合、各生産スケジュールモデルのジョブおよびリソースの間には包含関係が存在する。例えば、Fig. 5.3(a)に示すショップとセルの包含関係、およびロットとワークの包含関係である。すなわち、モデル 1 のショップおよびロットは、より詳細なモデル 2 の中で、複数のセルおよびワークに分解されている。これらの包含関係を、生産スケジュールの作成者間で意見を調整し、ジョブおよびリソースの包含関係を表現するオブジェクトを作成する。これらの包含関係のオブジェクトにより、モデル 1 およびモデル 2 におけるオペレーション間の包含関係を表現するオブジェクトを自動的に作成することができる。

STEP2: 生産スケジュールの作成と整合性管理

各々の生産スケジュールの作成者は、5.4.1で述べた手順で、各々の生産スケジュールモデルを用いて、生産スケジュールを作成する。この時、包含関係を有するオペレーションの開始時刻および終了時刻の間には、式(5.10)および(5.11)の条件を満たすことが求められる。すなわち、Fig. 5.3 (b)の場合、モデル 1 のショップ 1 におけるオペレーションの開始時刻と終了時刻は、モデル 2 のセル 1 およびセル 2 の全てのオペレーションの開始時刻と終了時刻に一致することが必要になる。このため、モデル 1 および 2 の間で、オペレーションの開始時刻と終了時刻を交換しながら、式(5.10)、(5.11)および(5.13)の関係を満足するように生産スケジュールを作成する。これにより、複数



(a) Part-of relations about jobs and resources



(b) Relations among starting time and finishing time of operations

Fig. 5.3 Consistency among production plans

の生産スケジュール間の整合性を保証することができる。

5.5 生産スケジュールのモデリングシステムの開発

以上の考えに基づき、オブジェクト指向言語の Smalltalk を用いて、生産スケジューリングのためのモデルリポジトリとモデリングシステムのプロトタイプを開発した。以下に、モデルリポジトリとモデリングシステムの概要を示すとともに、生産スケジューリング、および生産スケジュール間の整合性管理の例を示す。

5.5.1 生産スケジュールのモデルリポジトリとモデリングシステム

5.3節で提案した7つのオブジェクトを生産スケジュールモデルの基本的な構成要素として、ロット、ワ

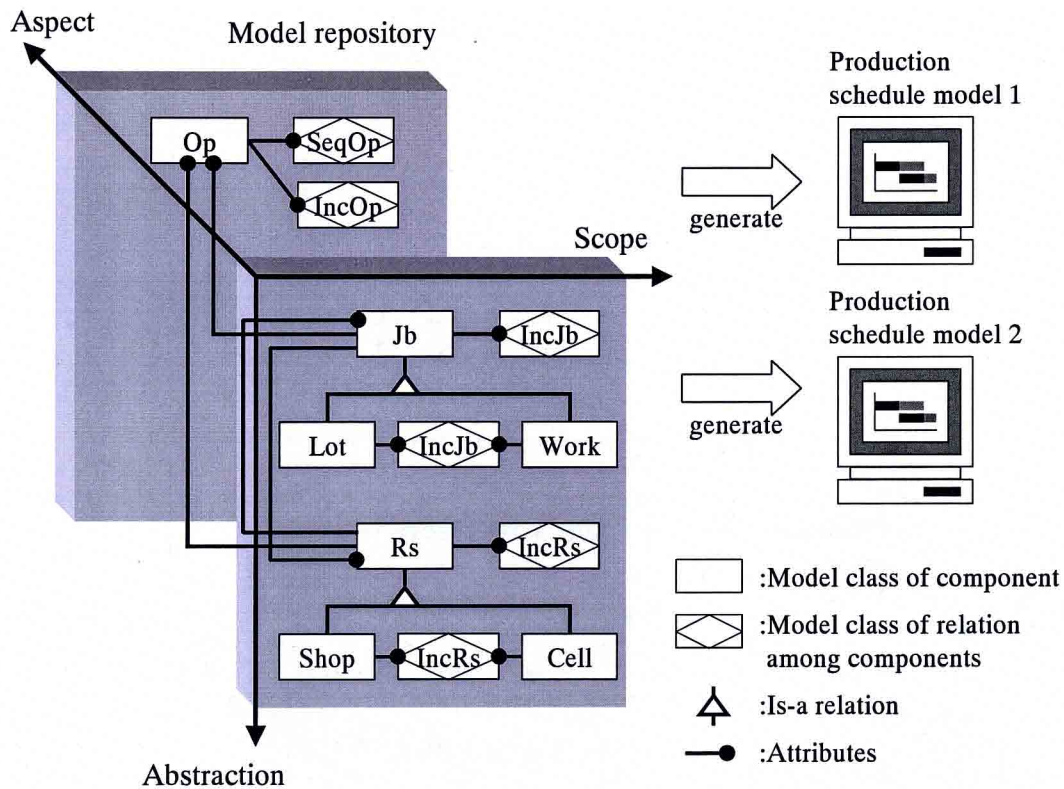


Fig. 5.4 Model repository for production scheduling

ーク、ショップ、セルなどの生産スケジュールモデルで使用される構成要素を、Fig. 5.4のように構造化し、生産スケジュールモデルのためのモデルリポジトリを作成した。生産スケジュールの作成者は、モデルリポジトリからジョブやリソース等の適当なモデルクラスを抽出し、そのモデルインスタンスを生成することで、生産スケジュールモデルを作成することができる。

さらに、本プロトタイプシステムでは、生産スケジュールの作成者によるモデル化や生産スケジュールの作成を支援するために、以下のプログラムモジュールを準備している。

(1) 生産スケジューリングプラットフォーム(Production scheduling platform)

作成した生産スケジュールモデルのジョブの処理順序に矛盾がないかどうかを調べ、その総処理時間を計算する機能を持つオブジェクトを実装した。

(2) 生産スケジューリングエンジン(Production scheduling engine)

リソースに配分されたジョブの最適な処理順序を決定し、生産スケジューリングプラットフォームや生産スケジュールの作成者に、最適な生産スケジュールを提供する機能を持つオブジェクトを実装した。すなわち、生産スケジューリングの解法を持つオブジェクトである。本章では、ジョンソン・ルール(Johnson's rule) [Johnson 54] の解法を持つオブジェクトを実装した。これを用いて、n 個の

種類の異なるジョブを生産する2工程のフローショップ型生産システムで、総所要時間最小となるジョブの処理順序を求めることができる。

(3) ネットワークインタフェース(Network interface)

異なる計算機上の生産スケジュールのモデリングシステム間で、整合性管理に必要な情報を、ファイルを介して授受するオブジェクトを実装した。

(4) ユーザインタフェース(User interface)

生産スケジュールの作成者とプロトタイプシステム間で必要な情報を授受するために、生産スケジュールの作成に必要な情報の入力を促すウィンドウを生成するオブジェクトや、作成した生産スケジュールをグラフィカルに表示するオブジェクトを実装した。

5.5.2 ジョンソン・ルールによる生産スケジューリング技法

ジョンソン・ルールは、2工程のフローショップ型生産システムにおいて、総処理時間が最小となるジョブの処理順序を決定する最適解法である。基本的な生産スケジューリングの考え方は、ジョブの待ち時間を最小にするように、各ジョブの処理順序を決定する。以下に、その手順を示す。

STEP1: 処理時間最小のジョブの選択

処理順序が決定していないジョブの中で、処理時間最小のオペレーションを持つジョブを選ぶ。

STEP2: ジョブの処理順序の決定

そのオペレーションが、ジョブの生産工程の第1工程である場合は、そのジョブを処理順序の前から割り付ける。第2工程の場合は、処理順序の後ろから割り付ける。

STEP3: 終了処理

全てのジョブの処理順序が決定していれば、生産スケジューリングを終了する。処理順序を決定していないジョブがあれば、STEP1から繰り返す。

5.5.3 生産スケジュール間の整合性管理の一例

階層的な2つの生産スケジュールモデルを作成し、生産スケジュール間の整合性の管理を行った例を示す。

Fig. 5.5(a)に、生産スケジューリングの対象となる生産システムを示す。生産スケジュールの作成者Aは、Fig. 5.5(b)に示すように、加工ショップと塗装ショップからなるフローショップ型生産システムのモデル(モデル1)を作成し、2つのロットの投入順序を決定する。生産スケジュールの作成者Bは、Fig.

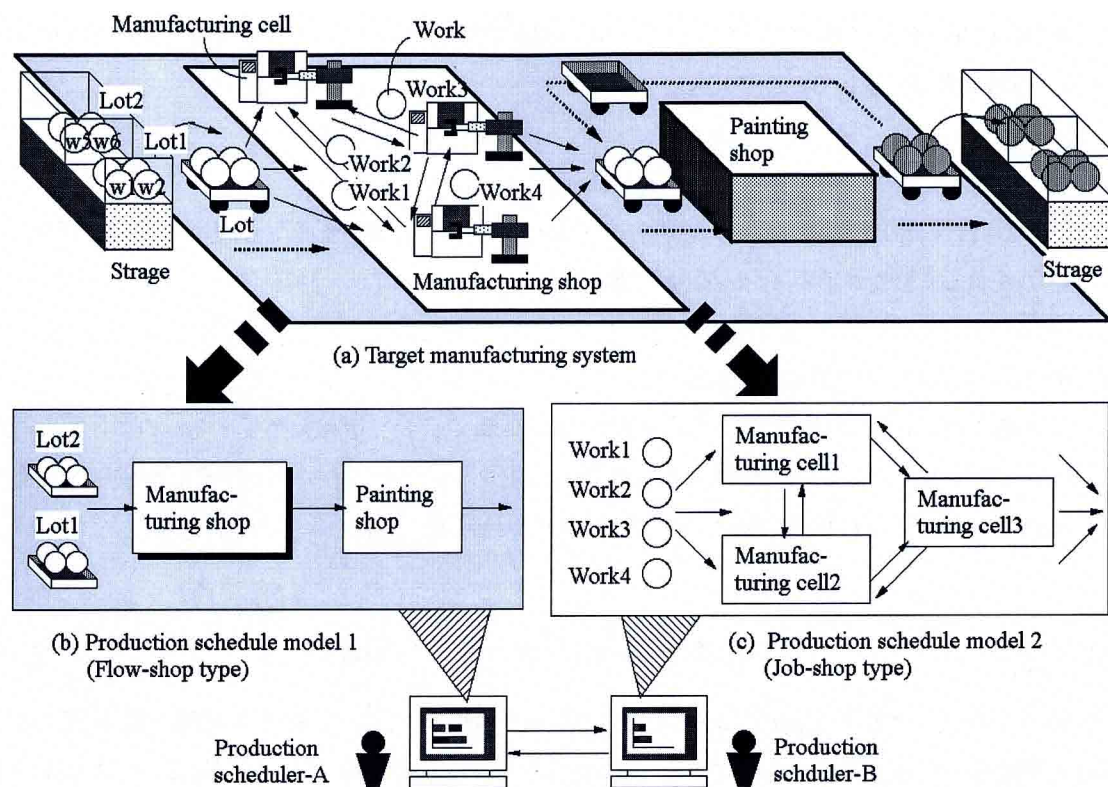


Fig. 5.5 Example of production schedule models

5.5 (c) に示すように、3つの加工セルからなるジョブショップ型生産システムのモデル(モデル2)を作成し、4つのワークの処理順序を決定する。なお、ロットとワーク、および加工ショップと加工セルの包含関係は図に示す通りである。

Fig. 5.6に、生産スケジュール間の整合性管理の過程の一例を示す。Fig. 5.6 (a-1) から (a-3) の3つのガントチャート(Gantt chart)は、Fig. 5.5のモデル1のオペレーションの変化を、Fig. 5.6 (b-1) から (b-3) は、モデル2のオペレーションの変化を示す。

まず、モデル1を用いて生産システム全体の生産スケジュール(a-1)を作成した。次に、この結果をモデル2に送り、その制約を満足するように加工ショップ内の生産スケジュール(b-1)を作成した。ここで、モデル1の生産スケジュールはジョンソン・ルールで自動的に、モデル2の生産スケジュールはマニュアルで作成している。この段階では、当然、両者の生産スケジュールは整合している。この状態において、モデル2におけるオペレーションの時間および生産スケジュールを(b-2)のように変更すると、モデル1のオペレーションの時間が自動的に変更され、生産スケジュールは(a-2)のようになる。しかし、この結果、加工完了時刻が納期を満足しなくなるため、再度、ジョンソン・ルールを適用すると、生産システム全体として(a-3)の生産スケジュールが得られる。このデータをモデル2に送ることにより、加工ショップの最終的な生産スケジュールとして(b-3)を作成した。

以上のように、本章においては、ジョンソン・ルールによる生産スケジューリングエンジンしか実装していないため、生産スケジュールをマニュアルで作成する必要があるが、オペレーションに関するデータ

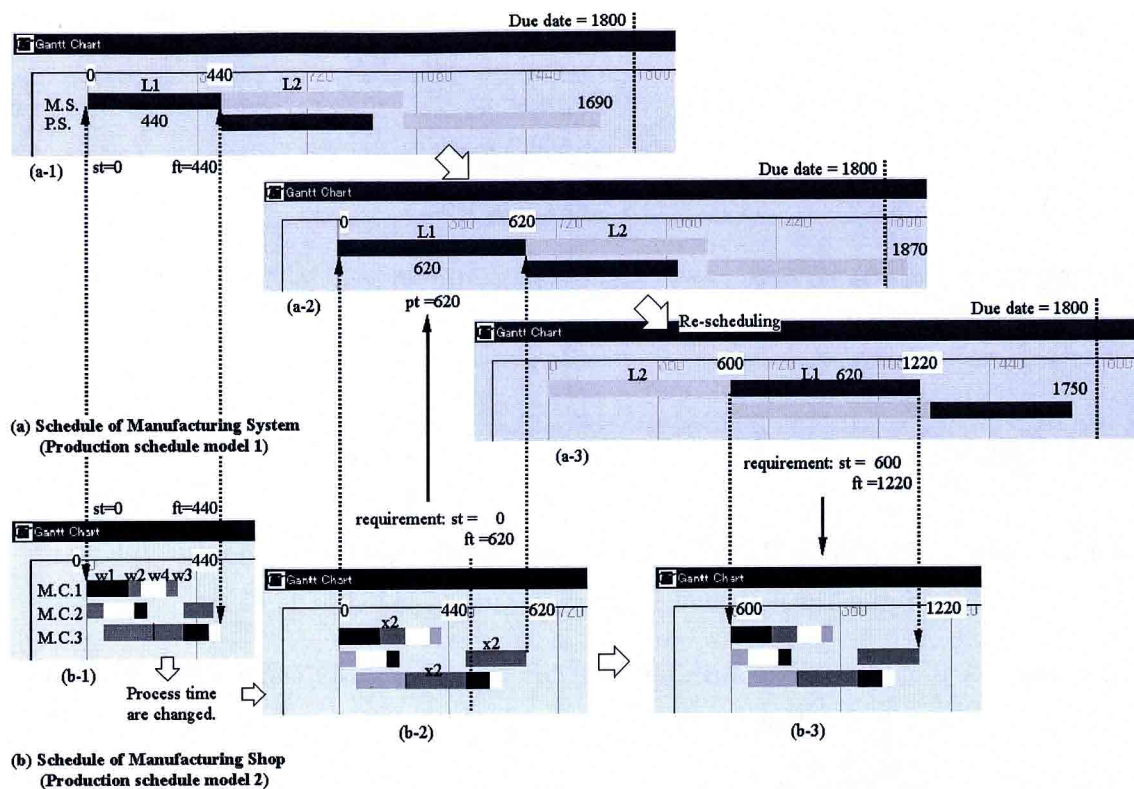


Fig. 5.6 Consistency management of schedules

をモデリングシステム間で授受することで、相互に関連する 2 つの生産スケジュール問題をコンカレントに処理することができる。

5.6 結言

本章では、第 2 章で提案した生産システムモデルの表現方法に基づき、生産スケジュールモデルの基本的な構成要素を提案するとともに、そのモデルを用いて、階層的な複数の生産スケジュールモデル間の整合性を管理する方法を提案した。以下に本章の内容をまとめる。

- (1) 第 2 章で提案した生産システムモデルの表現方法に基づき、生産スケジュールモデルの基本的要素として、ジョブ、リソース、オペレーション、およびオペレーションの順序関係という 4 つのオブジェクトと、階層的な生産スケジュールモデル間の関係を表す要素として、ジョブの包含関係、リソースの包含関係、およびオペレーションの包含関係という 3 つのオブジェクトを提案した。
- (2) (1)で示した 4 つの基本的なオブジェクトに基づいて生産スケジュールモデルを作成し、そのモデルを用いて、単一の生産スケジュールを作成する手順を提案した。
- (3) (1)で示した階層的な生産スケジュールモデル間の関係を表す 3 つのオブジェクトに基づいて、階層的な複数の生産スケジュールモデル間の整合性を管理することで、生産スケジュール間の整合

性を管理する手順を提案した。

- (4) オブジェクト指向言語の Smalltalk を用いて、生産スケジューリングのためのモデルリポジトリとモデリングシステムのプロトタイプを開発した。このプロトタイプシステムを用いて、生産スケジュールの作成、および生産スケジュール間の整合性管理を支援することができる。
- (5) 階層的な 2 つの生産スケジュールを統合的に作成し、生産スケジュールの変更に対する整合性管理の過程の一例を示した。

参考文献

- [Johnson 54] Johnson S.M., Optimal Two-and-Three-Stage Production Schedules with Setup Times Included, Nav. Res. Quart., 1, 1 (1954), 61-.
- [岩田 96] 岩田, 他 3 名, 生産領域における知識の体系化に関する研究成果報告書, IMS センター (1996), 20-23.
- [黒田 98] 黒田・遠国・榎本, ラグランジュ緩和法を用いたリアルタイム・スケジューリングの疑似最適化, 生産スケジューリング・シンポジウム'98 講演論文集 (1998), 65-70.
- [西川 91] 西川・玉置, ジョブショップ・スケジューリング問題に対する遺伝アルゴリズムの一構成法, 計測自動制御学会論文集, 27-5 (1991), 593-599.
- [錦戸 93] 錦戸・中野, TCM (Total Capacity Management) におけるスケジューリングシステム: FACTOR/SDM, SMM, 生産スケジューリングシンポジウム講演論文集, (1993), 7-12.
- [山田 93] 山田・中野, 遺伝アルゴリズムとスケジューリング問題, システム/制御/情報, 37-8 (1993), 484-489.
- [米田 97] 米田, ラグランジュ緩和法によるスケジューリング, システム/制御/情報, 41-4 (1997), 130-138.

第 6 章 階層分散型生産システムの運用におけるリスケジューリング

6.1 緒言

第 5 章では、階層分散型生産システムにおける生産スケジュールモデルの基本的な構成要素として、7 つのオブジェクトを提案するとともに、それに基づいて、階層的な複数の生産スケジュール間の整合性を管理する方法を提案した。これにより、階層分散型生産システムのプロトタイプ間の整合性を保証することができる。

しかし、生産スケジュールを整合的に作成した後に、工作機械の故障等による処理時間の増大やジョブの追加といった変更が発生する場合がある。このような変更が、生産活動の開始前に予測できる場合には、あらかじめ作成した生産スケジュールに基づき、再度、総処理時間等の計算が必要になる。これにより、あらかじめ決められた生産システム全体の納期を満たすことができない場合には、階層的な複数の生産スケジュール間で整合性を保持しながら、全体の納期を満たすように、迅速に生産スケジュールを変更(リスケジューリング)しなければならない。

そこで、本章では、第 5 章で提案したオブジェクトを用いて、階層分散型生産システムのプロトタイプ間の整合性を保持しながら、全体の納期を満たすように、コンカレントに複数の生産スケジュールを変更するリスケジューリングプロセス(Rescheduling process)を提案する。さらに、第 5 章で開発した生産スケジュールのモデリングシステムの機能を拡張して、リスケジューリングシステムのプロトタイプを開発するとともに、例を用いて、提案するリスケジューリングプロセスの有効性を検証する。

6.2 リスケジューリングに関する既存の研究

築山ら [築山 94] [森 93] は、2 階層の生産スケジュールリングシステムを提案している。このシステムでは、上位階層の生産スケジュールリングシステムが各セルへの投入開始時刻と終了時刻を決定し、下位階層の生産スケジュールリングシステムがその情報に基づき、セル内の生産スケジュールを作成する。これにより、階層間で整合性のある生産スケジュールを生成することができる。しかし、リスケジューリングプロセスは考慮していない。

西岡 [西岡 94] や Smith [Smith 95] は、オペレーションの処理時間が変化した場合でも、他のオペレーションの開始、終了時刻を修正する仕組みを提案している。諏訪ら [諏訪 98] は、定期的なリスケジューリングを実施するための最適な時間間隔を求める数理モデルを提案している。しかし、これらは、階層的なリスケジューリングは考慮していない。

また、樋野らは、再帰的伝播法【樋野 00a】を拡張し、多階層の生産スケジューリングおよびリスケジューリング問題にも適用している【樋野 00b】。しかし、この手法は、生産スケジュールの遅れ等の情報を工作機械やセル等の中で伝播しながら各構成要素が自律分散処理でリスケジューリングを行っていくため、適切な生産スケジュールを得るまでの時間が長くなるという問題がある。

6.3 リスケジューリングプロセスの提案

6.3.1 2つのリスケジューリングプロセス

ここでは、リスケジューリングの過程について述べる。リスケジューリングとは、工作機械の故障等による処理時間の増大やジョブの追加といった変更により、あらかじめ作成された生産スケジュールを変更することである。本章で提案するリスケジューリング手法では、あるオペレーションの処理時間の変化を予測し、あらかじめ決められた生産システム全体の納期、すなわち最上位階層の納期を満たすように、迅速に、将来の生産スケジュールを変更する。

リスケジューリングに対して、以下の条件を仮定する。

- (1) リスケジューリングを始める前の初期状態においては、各階層で最適な生産スケジュールが既に作成されている。
- (2) 各生産スケジュールは、異なる部門の人や計算機により、作成され、管理されている。
- (3) 処理時間の遅れは予測可能である。
- (4) 最下位階層の生産スケジュールにおける各オペレーションの処理時間を短縮することはできない。

リスケジューリングの過程を、例を用いて説明する。まず、Fig. 6.1に示すように、工場層とショップ層という2つの階層の生産スケジュールを考える。工場層の生産スケジューリングシステムでは、工場の中のショップに対する生産スケジュールを作成する。ショップ層の生産スケジューリングシステムでは、ショップの中のセルに対する生産スケジュールを作成する。Fig. 6.2(a-1)とFig. 6.2(a-2)のように、初期状態において、これらの生産スケジュールは整合している。そこで、ショップ層におけるセルのオペレーションの処理時間が増大すると予測された場合、以下の手順でリスケジューリングを行うものとする。

STEP1: 生産スケジュールの遅延の予測

リスケジューリングが必要になるのは、下位の階層において、総処理時間が増大し、あらかじめ決められた最遅終了時刻を越えてしまうと予測される場合である。Fig. 6.2(b-2)のように、セル1のオペレーションの処理時間が増大し、ショップ層の総処理時間が増大した場合、工場層の生産スケジュールが影響を受ける。つまり、オペレーションの処理時間の変化を予測した場合には、まず、それが総処理時間を変化させるかどうかを予測することが必要である。ショップ層の総処理時間に変

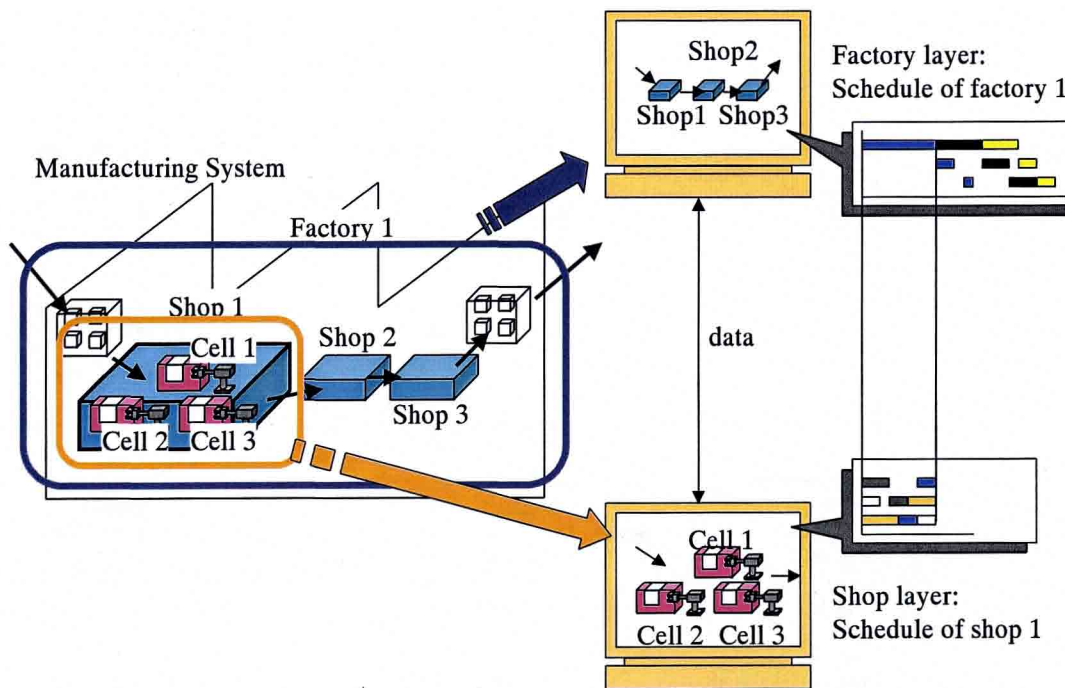


Fig. 6.1 Schedules on different layers

化がなければ、リスケジューリングは行わなくてよい。

STEP2: 総処理時間の上位階層への伝達

ショップ層の総処理時間に変化があれば、Fig. 6.2 (b-1)のように、ショップ層の総処理時間を工場層の生産スケジュールに伝達する。伝達された総処理時間の変化は、工場層において、あるオペレーションの処理時間の変化として認識される。

STEP1とSTEP2を最上位階層まで繰り返し、全体の総処理時間が増大するかどうかを調べる。Fig. 6.2の場合、工場層よりも上位の階層があれば、工場層の総処理時間を上位階層へと伝達する。最上位階層における全体の総処理時間が増大し、あらかじめ決められた最遅終了時刻を越えてしまうと予測される場合には、各階層でリスケジューリングを開始、つまり、リソースにおけるジョブの処理順序を変更し、適切な総処理時間を再計算しなければならない。

STEP3: 各階層におけるリスケジューリング

Fig. 6.2 (c-1) および (c-2) のように、ショップ層以上の全階層でリスケジューリングを実行し、初

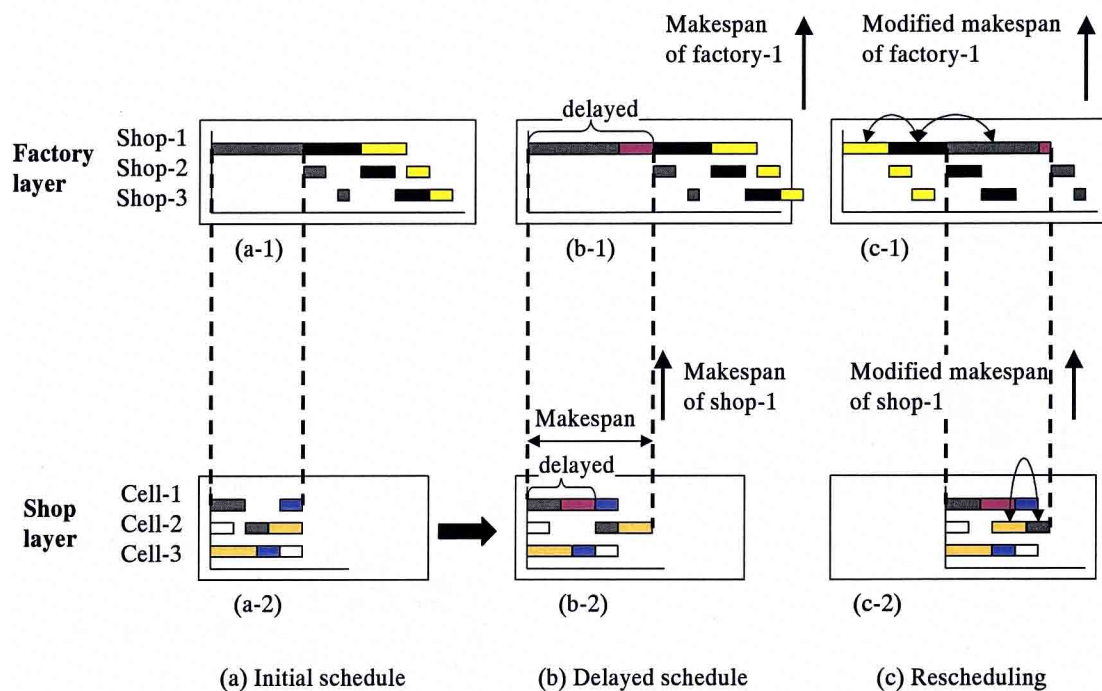


Fig. 6.2 Rescheduling process

期状態におけるジョブの処理順序を変更する。ただし、ショップ 1 およびその上位階層以外は、ショップ 1 の総処理時間の変化に対して影響を受けない。そのため、他の構成要素については、リスケジューリングを行う必要がない。

リスケジューリングの過程には、生産スケジュール間の整合性管理の観点から、次の 2 つのアプローチを考慮することができる。

(a) シーケンシャル・アプローチ

この方法では、下位階層の生産スケジュールにおいて、まずリスケジューリングを行う。Fig. 6.2 の例の場合、まず、Fig. 6.2 (c-2) のように、ショップ層においてリスケジューリングを行い、その総処理時間を短縮するように、各セルのジョブの処理順序を変更する。次に、その改善された総処理時間を工場層の生産スケジュールに伝達する。さらに、STEP1 と STEP2 を繰り返すことで、最上位階層の総処理時間を計算し、全体の納期以内に納まるかどうかを調べる。

最上位階層の総処理時間が増大する場合、工場層では、Fig. 6.2 (c-1) に示すように、ショップ層のリスケジューリング結果の総処理時間に基づいて、リスケジューリングを行う。Fig. 6.3 に示すように、このような過程を、最上位階層の生産スケジュールを更新するまで繰り返す。

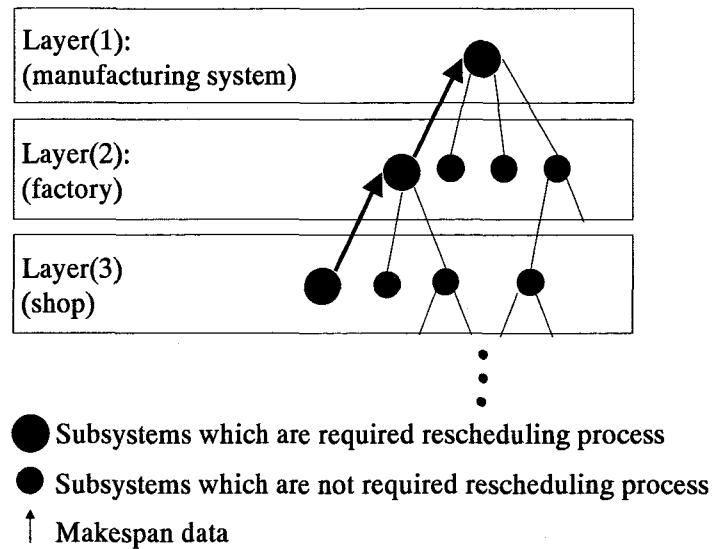


Fig. 6.3 Hierarchy of schedules

(b) コンカレント・アプローチ

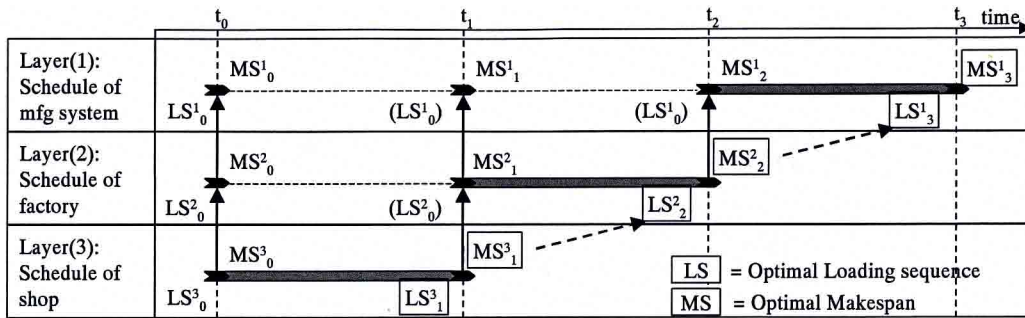
この方法では、ショップ 1 以上の上位階層の構成要素において、同時かつ並列にリスケジューリングを行う。例えば、Fig. 6.2(c-1)(c-2)のように、ショップ層と工場層が、増大したセル 1 の処理時間、およびセル 1 の処理時間の変化により増大したショップ 1 の総処理時間に基づいて、各々、リスケジューリングを検討し始める。各階層のリスケジューリングにおいて総処理時間が短縮されれば、その情報を上位階層へと伝達し、生産スケジュール間の整合性を保持するように、処理時間を変更する。上記の処理を、最上位階層における総処理時間が納期を満足するまで繰り返す。

本章では、次節に示す比較結果に基づき、リスケジューリングの方法として、コンカレント・アプローチを用いる。

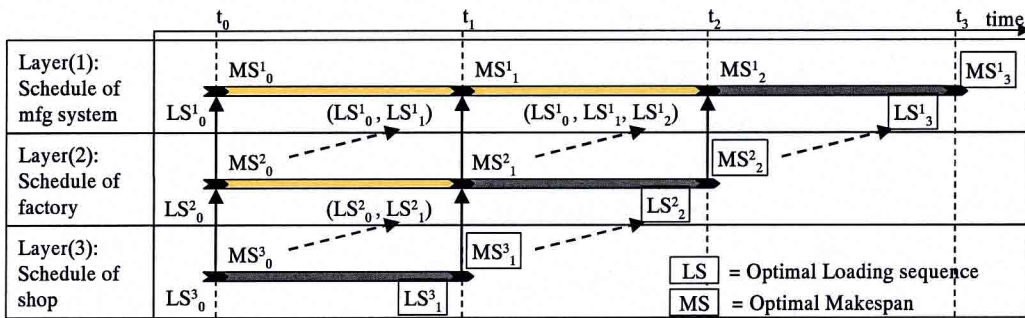
6.3.2 リスケジューリングプロセスの比較

ここでは、シーケンシャル・アプローチとコンカレント・アプローチの 2 つの方法を、リスケジューリングに要する時間の観点から比較する。つまり、リスケジューリングに要する時間が短い方が優れていると考える。比較を容易にするために、以下の条件を仮定する。

- (1) 各階層のリスケジューリングにより、ジョブの最適な処理順序と総処理時間が導き出せる。
- (2) 各階層のリスケジューリングに要する時間は等しい。



(a) Sequential approach



(b) Concurrent approach

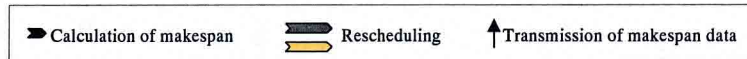


Fig. 6.4 Sequential and concurrent approaches for rescheduling

2つのアプローチによるリスケジューリングプロセスの例をFig. 6.4に示す。水平方向は時間を、垂直方向は生産スケジューリングにおける各階層を表す。時刻 t_j は、リスケジューリングの j 回目の試行の終了時刻を表す。また、 LS_j^i は i 階層の生産スケジュールにおいて、時刻 t_j の試行において決定したジョブの処理順序を、 MS_j^i はその時の総処理時間を表す。

Fig. 6.4 (a) は、シーケンシャル・アプローチによるリスケジューリングプロセスを表している。シーケンシャル・アプローチの場合、まず最初の試行で、階層 (3) におけるジョブの最適な処理順序 LS_1^3 を決定する。次に、 LS_1^3 とオペレーションの処理時間に基づいて、総処理時間 MS_1^3 を計算する。6.3.1項で述べた STEP1 と STEP2 に従い、 MS_1^3 を階層 (2) に伝達し、階層 (2) の生産スケジュールの総処理時間 MS_2^2 を計算する。このシーケンシャル・アプローチでは、 j 回目のリスケジューリングプロセスにおいて上位階層の総処理時間を次式で計算することができる。

$$MS_j^i = \begin{cases} F(MS^{i+1}_j, LS^i_0) & : \text{if } j < n-i+1 \text{ (} i=1, 2, \dots, n-1 \text{)} \\ F(MS^{i+1}_j, LS^i_j) & : \text{if } j = n-i+1 \text{ (} i=1, 2, \dots, n-1 \text{)} \end{cases} \quad (6.1)$$

上記(6.1)式は、総処理時間 MS_j^i は、 MS^{i+1}_j と LS^i_0 、または MS^{i+1}_j と LS^i_j を用いて計算することを意味

する。

最上位階層において、総処理時間 MS^1_1 が全体の納期を満たさない場合、階層(3)の総処理時間 MS^3_1 を用いて、階層(2)で、リスケジューリングを行い、ジョブの最適な処理順序 LS^2_2 と総処理時間 MS^2_2 を求めなければならない。最下位階層における、ジョブの最適な処理順序と総処理時間は、1 回目のリスケジューリングの試行により求まる。また、その 1 つ上位階層における、ジョブの最適な処理順序と総処理時間は、2 回目のリスケジューリングの試行により求まる。すなわち、一般に、 n 階層からなる生産スケジュールでは、ジョブの処理順序 LS^{i+1}_{n-i+1} と総処理時間 MS^{i+1}_{n-i+1} が最適なものとなる。

Fig. 6.4 (b) は、コンカレント・アプローチによるリスケジューリングプロセスを表している。この場合、全ての階層は、最下位階層のリスケジューリングと同時にかつ並列にリスケジューリングを検討し始める。例えば、3 階層の生産スケジュール全てにおいて、総処理時間 $MS^{i+1}_0 (i=1,2)$ を用いて、ジョブの処理順序 LS^i_1 を求める。次に、 i 階層の総処理時間 MS^i_1 は、下位階層の総処理時間 MS^{i+1}_1 と、 i 階層のジョブの処理順序 LS^i_0 もしくは LS^i_1 から計算することができる。このコンカレント・アプローチでは、 j 回のリスケジューリングを繰り返すことにより、上位階層の総処理時間を次式で計算することができる。

$$MS^i_j = \begin{cases} F (MS^{i+1}_j, (LS^i_0, LS^i_1, \dots, LS^i_j)) & \text{if } j < n-i+1 (i=1, 2, \dots, n-1) \\ F (MS^{i+1}_j, LS^i_j) & \text{if } j = n-i+1 (i=1, 2, \dots, n-1) \end{cases} \quad (6.2)$$

上記(6.2)式の第 1 式は、階層 i における総処理時間 MS^i_j を、 j 回目までのリスケジューリングプロセスにおいて得られた $j+1$ 個の処理順序 $LS^i_j (j=0, 1, 2, \dots, n-i)$ に、階層 $i+1$ の総処理時間 MS^{i+1}_j を入力することにより計算することを意味する。すなわち、階層 i において j 回目のリスケジューリング試行後には、 $j+1$ 個の処理順序が得られている。これらの処理順序に、 j 回目の試行で決定した階層 $i+1$ の総処理時間 MS^{i+1}_j を入力し、総処理時間を求め、最も小さいものを階層 i の総処理時間 MS^i_j とする。これは、 $j < n-i+1$ の場合、下位階層(階層 $i+1$)の総処理時間の最小値 MS^{i+1}_j を用いて最適な処理順序を求めることができないため、それまでのリスケジューリングプロセスで得られた処理順序に下位階層の総処理時間を入力することで、納期に間に合う生産スケジュールを決定しようとするものである。

これに対して、(6.2)式の第 2 式に示すように、 $j=n-i+1$ の場合、下位階層(階層 $i+1$)の総処理時間の最小値 MS^{i+1}_j を用いて最適な処理順序 LS^i_j を求めることができる。

例えば、Fig. 6.4(b)の階層 1 においては、2 回目のリスケジューリング試行後に、 (LS^1_0, LS^1_1, LS^1_2) の 3 個の処理順序が作成されている。これらの処理順序に対して、階層 2 で作成された総処理時間 MS^2_2 を入力し、3 個の総処理時間を求める。これらと比較し、最も小さいものが総処理時間 MS^1_2 になる。ここで、この値が全体の納期を満たすかどうかを調べ、納期を満たしていれば、全てのリスケジューリングプロセスを停止する。3 回目のリスケジューリングにおいては、階層 2 で得た最小の総処理時間 MS^2_2 を用いて最適な処理順序 LS^1_3 および総処理時間 MS^1_3 を求める。これは、シーケンシャル・アプローチの階層 1 で得られる最適な結果と同じになる。

シーケンシャル・アプローチとコンカレント・アプローチの比較結果を以下に示す。

- (1) どちらのアプローチでも、 n 回のリスケジューリングの試行後に、最上位階層の最適な総処理時間を計算することができる。
- (2) Fig. 6.4 (b) に示すように、コンカレント・アプローチでは、リスケジューリングの各試行中に、ジョブの処理順序 LS_j が複数個、検討されている。いずれかの処理順序により、最上位階層の総処理時間が納期以下に収まる場合があるため、リスケジューリング全体に要する時間を短くすることが可能である。これが、コンカレント・アプローチの優れている点である。

6.4 リスケジューリングシステムの実装

6.4.1 リスケジューリングシステムの構成

第 5 章で開発した生産スケジューリングのモデリングシステムの機能を拡張して、リスケジューリングシステムのプロトタイプを開発した。本プロトタイプシステムの開発には、オブジェクト指向言語の Smalltalk を用いた。本プロトタイプシステムは、分散環境でコンカレントに生産スケジューリングを作成できるだけでなく、生産スケジューリング間の整合性管理を自動的に行えるようにするため、以下のプログラムモジュールを準備している。

(1) 生産スケジューリングプラットフォーム(Production scheduling platform)

第 5 章では、ジョブの処理順序の矛盾を調べ、総処理時間を計算するオブジェクトを実装した。本章では、さらに、6.3 節で提案したコンカレント・アプローチによるリスケジューリングプロセスに基づき、階層的な複数の生産スケジューリング間の整合性を保持する機能を追加した。

(2) 生産スケジューリングエンジン(Production scheduling engine)

リソースにおけるジョブの適切な処理順序を決定するプログラムモジュールとして、第 5 章では、ジョンソン・ルール [Johnson 54] の解法を持つオブジェクトを実装した。本章では、さらに、遺伝的アルゴリズム(Genetic algorithm)の解法 [Holland 75] を持つオブジェクトを実装した。これにより、フローショップ型生産システムだけでなく、ジョブショップ型生産システムにおいても、適切な生産スケジューリングを求めることができる。

(3) ネットワークインタフェース(Network interface)

第 5 章では、異なる計算機上の生産スケジューリングのモデリングシステム間で、ファイルを介して情報を授受するオブジェクトを実装した。本章では、CORBA(Common Object Request Broker Architecture) [Siegel 96] [Ryan 96] に準拠して情報を授受するオブジェクトを実装した。これにより、整合性管理に必要な情報を、ネットワークを介して異なる計算機間上のリスケジューリングシステム間で直接、授受することができる。そのため、生産スケジューリング間の整合性管理を自動的に

行うことができる。

(4) ユーザインタフェース(User interface)

第 5 章で実装した生産スケジュールをグラフィカルに表示するオブジェクトを利用するとともに、リ
スケジューリングに必要な情報の入力を促すウィンドウを生成するオブジェクトを実装した。

6.4.2 遺伝的アルゴリズムによる生産スケジューリング技法

遺伝的アルゴリズムを用いた生産スケジューリングの方法について説明する。以下、これを GA と呼ぶ。
GA は、自然淘汰の手法に基づいた、改善型の実行可能解法であり、ジョブショップ型の生産システム
においても、近似最適解を求めることが可能である。まず、各階層におけるジョブの処理順序を表現す
る個体 (individual) を以下のように記述する。

$$((Jb_1(Rs_1), Jb_2(Rs_1), \dots, Jb_n(Rs_1)), (Jb_1(Rs_2), \dots, Jb_n(Rs_2)), (\dots, Jb_n(Rs_m))) \quad (6.6)$$

ここで、ジョブ i を Jb_i 、リソース j を Rs_j 、リソース j でジョブ i に施すオペレーションを $Jb_i(Rs_j)$ と表す。上記
(6.6)式において、個体のひとつのオペレーション $Jb_i(Rs_j)$ を遺伝子 (gene) と呼ぶ。遺伝子の並び
($Jb_1(Rs_j), \dots, Jb_n(Rs_j)$) は、リソース Rs_j におけるジョブ Jb_i ($i=1, 2, \dots, n$) の処理順序を表す。

GA では、複数の個体に交叉 (crossover) や突然変異 (mutation) という手段を適用し、 $Jb_i(Rs_j)$ の
並びを入れ替えることで、近似最適解となる生産スケジュールを求める。GA による生産スケジューリング
の手順 [圓川 96] を以下に示す。なお、目的関数は総処理時間とする。

STEP1: 初期設定

あらかじめ定めた個体の数、すなわち集団サイズ (Population size) を満たすまで、新しい個体
をランダムに生成し、個体集合を構成する。これを第 1 世代の個体集合とする。

STEP2: 適応度の計算

各個体の表す生産スケジュールの総処理時間を計算する。

STEP3: エリート保存

最短の総処理時間を持つ個体を次世代の個体集合に残す。

STEP4: 交叉

残りの個体に対し、交叉を行う個体をランダムに決定する。交叉とは、2 つの個体間で、いくつか
の遺伝子 $Jb_i(Rs_j)$ を選択し、交換する方法である。これにより得られた個体の表す生産スケジュー
ルの実行の可能性を調べ、矛盾のある個体は破棄する。

STEP5: 突然変異

STEP4 で矛盾のない個体に対して、確率的に突然変異を行う。突然変異とは、ある個体のいくつかの遺伝子 $Jb_i(Rs_j)$ を選択し、その並びを入れ替える方法である。その方法のひとつとして、ここでは逆位(reverse)を適用する。逆位とは、ある個体のいくつかの遺伝子 $Jb_i(Rs_j)$ を選択し、それを逆向きに並び替える方法である。個体の表す生産スケジュールの実行の可能性を調べ、矛盾のない個体のみを次世代の個体集合に残す。

STEP6: 個体数と集団サイズとの比較

次世代の個体の数が、集団サイズよりも小さければ、STEP4 から繰り返す。

STEP7: 世代数の比較

現在の世代があらかじめ定めた最終世代よりも小さければ、STEP2 から繰り返す。

STEP8: 終了処理

これまでに得られた個体で、総処理時間が最短の生産スケジュールを解として決定し、生産スケジューリングを終了する。

6.5 ケーススタディ

6.5.1 コンカレント・アプローチを用いたリスケジューリングの試行

Fig. 6.5のような3階層の生産スケジュールモデルを作成し、コンカレントにリスケジューリングを行った例を示す。

Fig. 6.5 (a) に示すように、生産システムは2つの工場で構成されている。工場には、ロットの集まりが順次投入される。これは2工程のフローショップ型生産システムである。これを階層(1)とする。工場1は、Fig. 6.5 (b) に示すように、3つのショップから構成されており、工場1に投入されたロットの集まりから、ロットが一つずつ順次投入される。これは、3工程のフローショップ型生産システムである。これを階層(2)とする。また、加工ショップ1は、Fig. 6.5 (c) に示すように、3つのセルで構成されており、ショップに投入されたロットの中のワークが個々に各セルへと投入される。これは、3工程のジョブショップ型生産システムである。これを階層(3)とする。

階層(1)の生産システムの初期生産スケジュールはジョンソン・ルールのプログラムモジュールを、階層(2)の工場1および階層(3)の加工ショップ1の初期生産スケジュールはGAのプログラムモジュールを用いて自動的に作成した。生産スケジュールの情報を互いに授受することで、初期状態では、3つの生産スケジュールは整合している。また、これらの生産スケジュールは、総処理時間の観点からは最適解もしくは近似最適解である。

各階層におけるジョブの生産工程および処理時間をTable 6.1, Table 6.2およびTable 6.3に示す。

また、各階層の生産スケジューリングにより得られたジョブの処理順序をTable 6.4, Table 6.5およびTable 6.6に示す。なお、表中の括弧内の数値は、各生産工程の処理時間を表す。

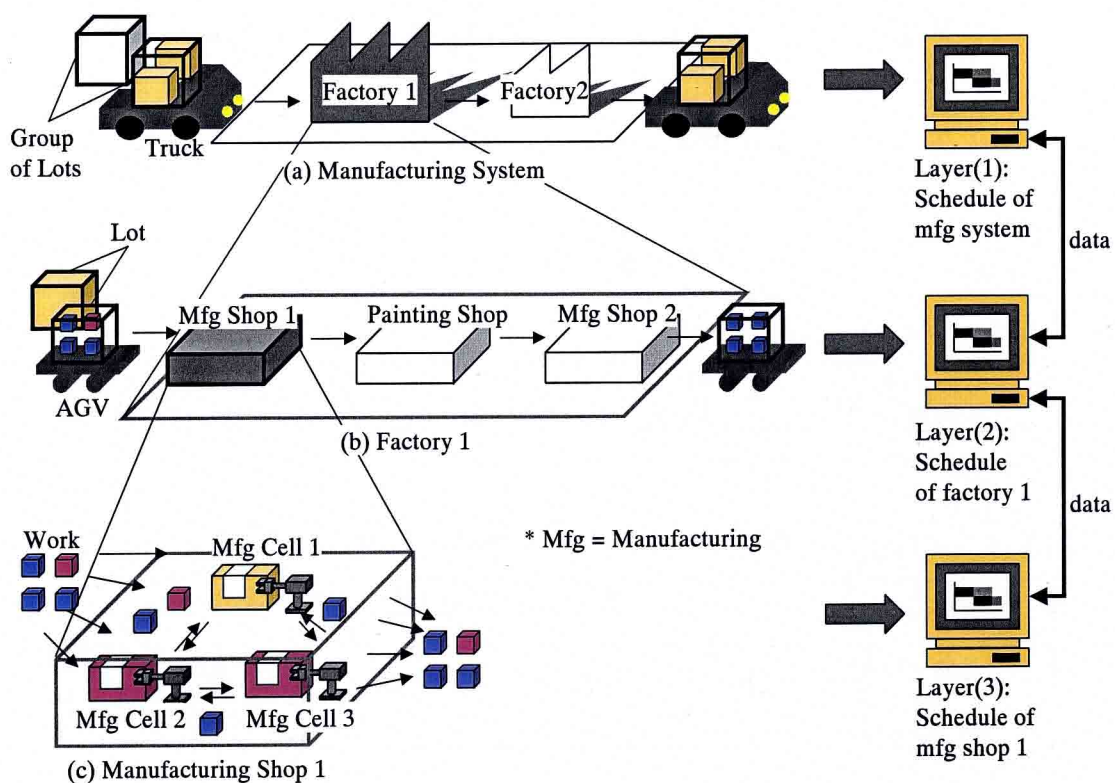


Fig. 6.5 Target manufacturing system for case study

Table 6.1 Production process for layer (1)

Job	Production process	
	1	2
Group of lot 1	Factory 1	Factory 2
Group of lot 2	Factory 1	Factory 2

Table 6.2 Production process for layer (2)

Job	Production process		
	1	2	3
Lot 1	Shop 1	Shop 2	Shop 3
Lot 2	Shop 1	Shop 2	Shop 3

Table 6.3 Production process for layer (3)

Job	Production process		
	1	2	3
Work 1	Cell 1	Cell 2	Cell 3
Work 2	Cell 1	Cell 3	Cell 2
Work 3	Cell 2	Cell 3	Cell 1
Work 4	Cell 2	Cell 1	Cell 3

Table 6.4 Sequence of jobs for layer (1)

Resource	Sequence of jobs (Processing time)	
	1	2
Factory 1	Group of lot 1 (2000)	Group of lot 2 (2150)
Factory 2	Group of lot 1 (2000)	Group of lot 2 (2150)

Table 6.5 Sequence of jobs for layer (2)

Resource	Sequence of jobs (Processing time)	
	1	2
Shop 1	Lot 1 (440)	Lot 2 (510)
Shop 2	Lot 1 (510)	Lot 2 (540)
Shop 3	Lot 1 (540)	Lot 2 (510)

Table 6.6 Sequence of jobs for layer (3)

Resource	Sequence of jobs (Processing time)			
	1	2	3	4
Cell 1	Work 2 (100)	Work 1 (50)	Work 4 (80)	Work 3 (40)
Cell 2	Work 3 (60)	Work 4 (100)	Work 1 (100)	Work 2 (100)
Cell 3	Work 3 (160)	Work 2 (100)	Work 1 (80)	Work 4 (40)

Fig. 6.6に、リスケジューリングプロセスの一例を示す。Fig. 6.6 (a-1) と (b-1) のガントチャートは生産システムの生産スケジュールの変化を、Fig. 6.6 (a-2) と (b-2) は工場1の生産スケジュールの変化を、Fig. 6.6 (a-3) と (b-3) は加工ショップ1の生産スケジュールの変化を表す。

まず、階層(3)の加工ショップ1において、ワーク3を加工する加工セル2の処理時間を60から160に、加工セル3の処理時間を160から260に増大すると、階層(3)のリスケジューリングシステムが再計算を行い、(a-3)のように加工ショップ1の生産スケジュールが変化する。それにより増大した総処理時間 $MS_0^3(640)$ が伝達され、階層(2)の工場1の生産スケジュールが(a-2)のように自動的に変更される。同様にして、 $MS_0^2(2200)$ が伝達され、階層(1)の生産システムの生産スケジュールが(a-1)のように変更される。

その結果、最上位階層の生産システムの総処理時間が6500になり、あらかじめ与えられた納期

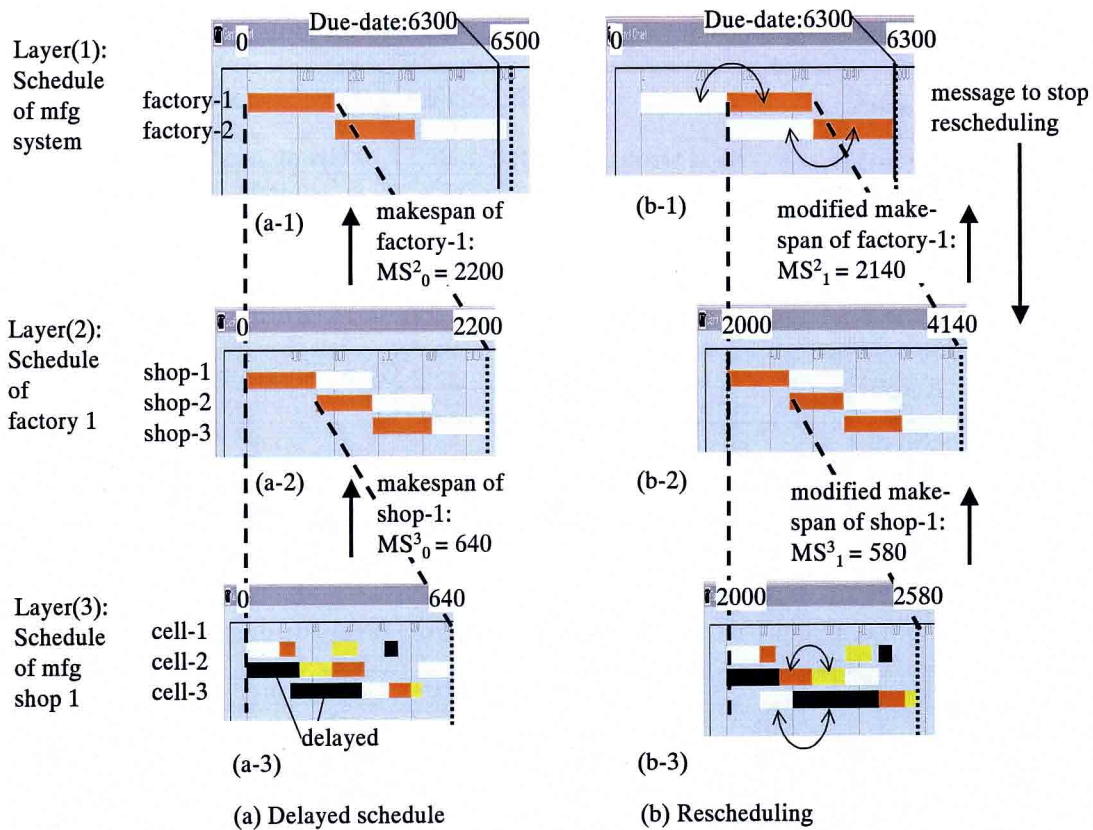


Fig. 6.6 Rescheduling process of target manufacturing system

6300 を満足しなくなる。そのため、生産システム、工場 1、および加工ショップ 1 の生産スケジュールに対して、(b-1)、(b-2)および(b-3)のようにコンカレントに、リスケジューリングが行われる。この場合、階層(1)と階層(3)の生産スケジュールにおいて、ジョブの処理順序が変更された。(b-3)に示す加工ショップ 1 のリスケジューリング終了後、工場 1 の生産スケジュールに伝達された総処理時間 $MS^3_1(580)$ により、総処理時間 $MS^2_1(2140)$ が(b-2)に示すように得られる。これにより、階層(1)の生産システムの総処理時間がさらに短縮され、(b-1)の生産スケジュールが得られた。その結果、最上位階層の生産システムの総処理時間が 6300 になり、納期を満足するため、全構成要素のリスケジューリングプロセスが停止される。すなわち、この場合、1 回の並列的なリスケジューリングにより納期を満たすことができた。

リスケジューリング終了後のジョブの処理順序および処理時間を Table 6.7, Table 6.8 および Table 6.9 示す。表中の括弧内の数値は、各生産工程の処理時間を表す。また、# 印の付いたジョブはリスケジューリングにより処理順序を変更したジョブを表し、* 印の付いた数値は変化した処理時間を表す。

Table 6.7 Sequence of jobs for layer (1) after rescheduling process

Resource	Sequence of jobs (Processing time)	
	1	2
Factory 1	# Group of lot 2 (2150)	# Group of lot 1 (2140*)
Factory 2	# Group of lot 2 (2150)	# Group of lot 1 (2000)

Table 6.8 Sequence of jobs for layer (2) after rescheduling process

Resource	Sequence of jobs (Processing time)	
	1	2
Shop 1	Lot 1 (580*)	Lot 2 (510)
Shop 2	Lot 1 (510)	Lot 2 (540)
Shop 3	Lot 1 (540)	Lot 2 (510)

Table 6.9 Sequence of jobs for layer (3) after rescheduling process

Resource	Sequence of jobs (Processing time)			
	1	2	3	4
Cell 1	Work 2 (100)	Work 1 (50)	Work 4 (80)	Work 3 (40)
Cell 2	Work 3 (160*)	# Work 1 (100)	# Work 4 (100)	Work 2 (100)
Cell 3	# Work 2 (100)	# Work 3 (260*)	Work 1 (80)	Work 4 (40)

6.5.2 数値実験による2つのリスケジューリングプロセスの比較

Fig. 6.5に示す同じ生産システムに対して、100例のリスケジューリング問題を作成し、シーケンシャル・アプローチとコンカレント・アプローチの数値実験を行った。各実験では、処理時間が増大するオペレーションを1~3個でランダムに選択し、その処理時間の増加率を100~500%の乱数で与えた。実験結果のグラフをFig. 6.7に示す。この実験結果に基づいて、2つのアプローチを計算回数および計算時間の観点から比較すると、以下の結果が得られた。

まず、最適な生産スケジュールが求まるまで、すなわち3回のリスケジューリングを行っても納期に間に合う解が求まらない場合は、両アプローチとも100例中53例であり、いずれも同一の問題であった。

解が求まった例について、リスケジューリングを開始してから終了するまでの繰り返しの回数は、シーケンシャル・アプローチの場合、平均2.68回、コンカレント・アプローチでは平均0.89回であった。また、2回以下の繰り返し回数でリスケジューリングを終了するのは、シーケンシャル・アプローチの場合、47例中5例、コンカレント・アプローチでは47例全てであった。

リスケジューリングを開始してから終了するまでの計算時間は、シーケンシャル・アプローチの場合、平均79.0秒、コンカレント・アプローチでは平均19.4秒であった。納期に間に合う解が求まった47例のうち、その計算時間がコンカレント・アプローチの方が短くなったのは42例、両アプローチで同一なのは5例であった。

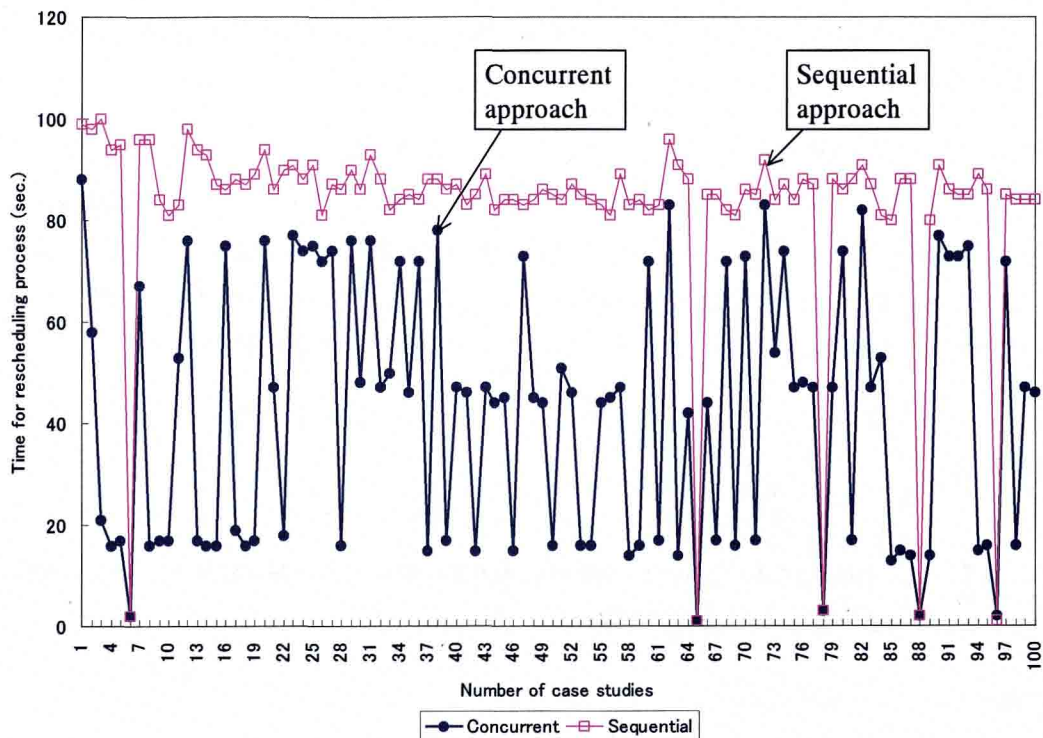


Fig. 6.7 Results of case studies

以上より、コンカレント・アプローチの方が、シーケンシャル・アプローチよりも短い時間か、もしくは同一の時間でリスケジューリングを終了することができると思われる。すなわち、コンカレント・アプローチの有効性を検証することができた。

6.6 結言

本章では、第5章で提案した生産スケジュールモデルのオブジェクトを用いて、階層分散型生産システムの生産スケジュール間の整合性を保持しながら、最上位階層の納期を迅速に満たすように、階層的な複数の生産スケジューリングシステムで、コンカレントにリスケジューリングを行う方法を提案した。以下に本章の内容をまとめる。

- (1) 階層分散型生産システムの生産スケジュール間の整合性を保持しながら、最上位階層の納期を満たすように、将来の生産スケジュールを変更する方法として、シーケンシャル・アプローチとコンカレント・アプローチの2つのリスケジューリングプロセスを示した。
- (2) 2つのリスケジューリングプロセスを、リスケジューリングに要する時間の観点から比較した結果、コンカレント・アプローチの方が優れていることを示した。

- (3) 第 5 章で開発した生産スケジュールのモデリングシステムの機能を拡張して、リスケジューリングシステムのプロトタイプを開発した。本プロトタイプシステムは、分散環境でコンカレントに生産スケジュールを作成できるだけでなく、階層的な複数の生産スケジュール間の整合性管理を自動的に行うことができる。
- (4) このプロトタイプシステムを、3 階層のリスケジューリング問題に適用し、コンカレント・アプローチによるリスケジューリングプロセスの一例を示した。さらに、100 例のリスケジューリング問題に対して、シーケンシャル・アプローチとコンカレント・アプローチの数値実験を行った。その結果、コンカレント・アプローチの方が、シーケンシャル・アプローチよりも短い時間か、もしくは同一の時間でリスケジューリングを終了できることを示し、コンカレント・アプローチの有効性を検証した。

参考文献

- [Holland 75] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).
- [Johnson 75] 第 5 章と同じ。
- [Ryan 96] 第 1 章と同じ。
- [Siegel 96] 第 1 章と同じ。
- [Smith 95] Smith, S. F., Reactive scheduling systems, *Intelligent scheduling systems*, Kluwer academic (1995), 155-192.
- [圓川 96] 圓川・伊藤, 生産マネジメントの手法, 朝倉書店 (1996), 119-127.
- [諏訪 98] 諏訪・浅田・三道, 最適定期リスケジューリング政策に関する数理モデル, 生産スケジューリングシンポジウム'98 講演論文集 (1998), 83-88.
- [築山 94] 築山・森・福田, 大規模生産システムの階層分散型スケジューリング, 計測と制御, 33-7 (1994), 581-584.
- [西岡 94] 西岡, 状況対応型スケジューリング問題へのアプローチ, 計測と制御, 33-7 (1994), 571-574.
- [樋野 00a] 樋野・森脇, 再帰伝播法による分散型ジョブショップスケジューリング, 日本機械学会論文集, 66-647, C (2000), 335-340.
- [樋野 00b] 樋野・泉原・森脇, 再帰伝播法による分散型ジョブショップスケジューリング (第 2 報, 階層構造を有するシステムのスケジューリング), 日本機械学会論文集, 66-651, C

(2000), 267-274.

[森 93]

森・築山・福田, 階層型意思決定システムによる FMS のオンラインスケジューリング法, 電気学会論文誌, 113-9, D (1993), 1054-1060.

第 7 章 ホロニック生産システムの運用 におけるリアルタイムスケジュー リング

7.1 緒言

第 5 章および第 6 章で対象とした階層分散型生産システムは、階層的な生産制御構造を形成する FMS(Flexible Manufacturing System)や FMC(Flexible Manufacturing Cell)といった自動化生産システムの生産制御構造に適応している。しかし、近年、生産システムの構成要素である生産設備は、計算機の導入による情報処理能力の向上、およびネットワーク化による情報通信能力の向上により、生産設備が自身の行動に関する意思決定を行う自律的制御が可能となっている。そこで、新たな生産システムとして、種々の自律分散型生産システムが提案されている [Iwata 94] [Tharumarajah 96] [Warnecke 93] [大倉 96] [沖野 95] [松本 99]。自律分散型生産システムのひとつとして、ホロニック生産システム(Holonc Manufacturing System) [森脇 92] [Sugimura 97] [Suyoto 99]がある。以下、これを HMS と呼ぶ。本章では、研究の領域を HMS における生産スケジューリングに絞って考察する。

第 5 章および第 6 章では、階層分散型生産システムにおいて、生産スケジュール間の整合性を管理する方法、および複数の生産スケジューリングシステム間で同時かつ並列にリスケジューリングを行う方法を提案した。これにより、生産開始前に予測した生産現場の状況の変化に対し、階層的な複数の生産スケジュール間の整合性を保持しながら、迅速にリスケジューリングを行うことができることを示した。しかし、生産開始後に生産現場に起こった変化には対応できないという問題がある。

自律分散型生産システムのスケジューリング手法として、リアルタイムスケジューリング(Realtime scheduling)手法が提案されている [鳩野 94] [馬 97]。リアルタイムスケジューリング手法は、作業の終了といったイベントが発生した時点で、次の作業の選択といった意思決定を動的に行う生産スケジューリング手法である。そのため、機械の故障など、生産開始後に起こった生産現場の状況の変化に、柔軟かつ頑健に対応することが可能であると考えられている。

そこで、本章では、まず、HMS の概念を示した後、HMS におけるリアルタイムスケジューリング手法を提案する。次に、第 5 章で提案した生産スケジュールモデルを拡張して、HMS のモデル化を行う。そのモデルに基づき、HMS のシミュレーションシステムのプロトタイプを開発するとともに、簡単な例を用いて、HMS のリアルタイムスケジューリングの柔軟性や頑健性の特長について考察する。

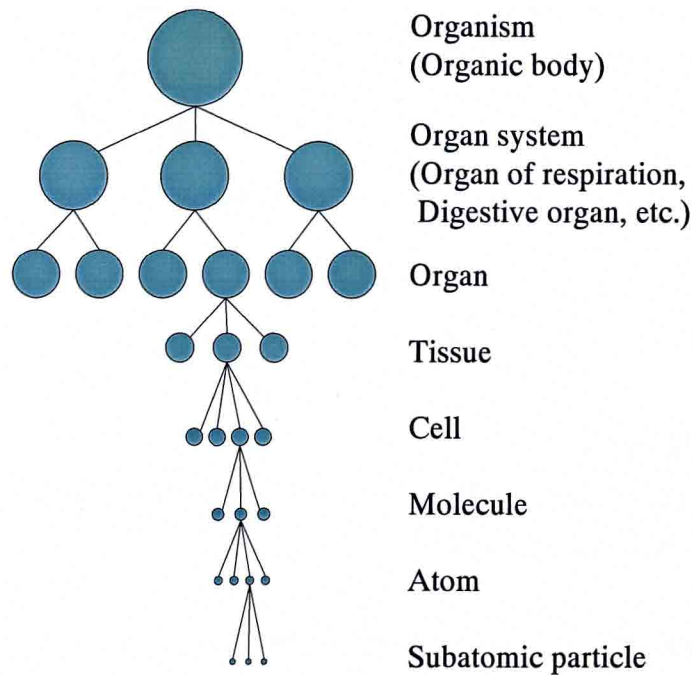


Fig. 7.1 Holarchy

7.2 HMS の概要

7.2.1 HMS の定義

Koestler [Koestler 78] は、システムとは、自律性を持った要素が全体-部分関係の階層構造を形成していると考えた。また、その階層構造において、各要素は、上位に対し部分として従属的であるが、同時に準自律的な全体としても機能していると考えた。そこで、階層構造の中間に位置する、全体とも部分とも呼べる要素を表現するために、ギリシャ語で全体をあらわすホロス(Holos)と、部分を意味する接尾辞オン(on)を組み合わせるホロン(Holon)という言葉の提案した。さらに、ホロンが構成する階層構造を表現するために、ホラーキ(Holarchy)という言葉の提案した。Fig. 7.1に、Koestler の提案したホラーキの例を示す。この図は、有機体のホラーキを表しており、節がホロンを、分岐線がホロン間の伝達と制御の経路を表している。

HMS は、Koestler の提唱したホロンの概念に基づく自律分散型生産システムである。HMS コンソーシアムでは、HMS に関する概念を以下のように定義している [Wyns 96]。

(1) ホロン(Holon)

HMS は、複数のホロンから構成される。ホロンとは、自律性と協調性を持つ生産システムの構成要素である。ホロンは、生産対象製品の加工や組立などの変形、搬送、および貯蔵を行う物理的機

能だけでなく、データの処理、伝達および蓄積を行う情報処理機能を持つ。また、ひとつのホロンは、他のいくつかのホロンから構成される。

(2) 自律性

自身の執るべき戦略や計画の実行を制御する能力。

(3) 協調性

要素間で互いに容認できる計画を作成し、それを実行する過程。

(4) ホラーキ (Holarchy)

ある目的を達成するために協調する複数のホロンのシステム。これにより、ホロンが協調するための基本的な規則が与えられ、自律性が制限される。

人工知能の分野では、エージェント(Agent)という概念がある。Russell ら [Russell 95] は、エージェントとは、ある環境をセンサ(Sensors)で知覚し、ある環境にエフェクタ(Effectors)を通して動作するものであると定義している。この定義に従うと、ホロンとは、自律性と協調性を持つエージェントであると言える。

7.2.2 HMS の基本構造

Russell らの定義では、エージェントは、アーキテクチャ(Architecture) とエージェントプログラム(Agent program)から構成される。アーキテクチャとは、センサからの知覚をエージェントプログラムが利用できるように処理し、エージェントプログラムの実行により得られた動作命令をエフェクタに伝達して動かす装置である。エージェントプログラムとは、エージェントの知覚から動作へのマッピングを実現する関数を設計することである。同様に、HMS はFig. 7.2に示すように、以下の 2 つの機能部分から構成される。

(1) 物理的処理部(Physical processing part)

情報処理部から伝達された命令に基づき、素材を生産対象製品へと変化させる製造プロセスを実行する部分。ここで認識された製造プロセスの情報は、情報処理部に伝達される。

(2) 情報処理部(Information processing part)

物理処理部から伝達された製造プロセスの情報に基づき、計算、解析および意思決定を行う部分。ここで生成した命令は、物理処理部に伝達される。

また、製造プロセスの観点から、HMS の構成要素は以下の 2 種類に分けられる。

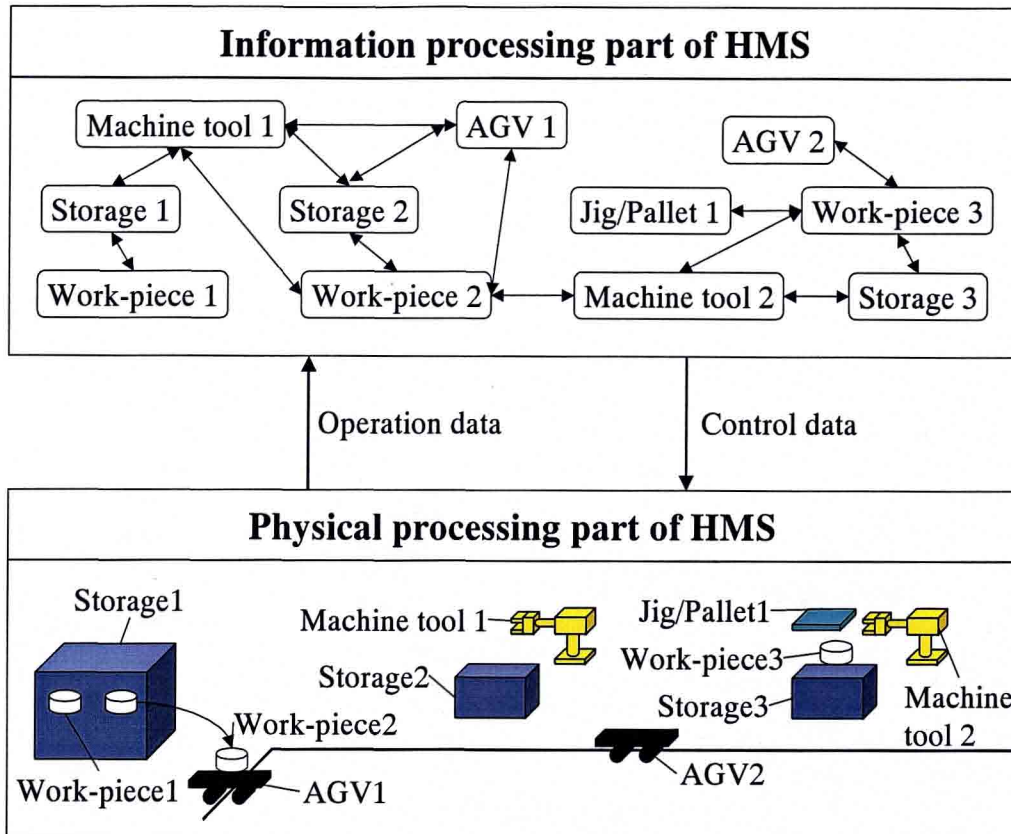


Fig. 7.2 Basic structure of holonic manufacturing system

(a) 生産設備(Manufacturing equipment)

生産対象製品の加工，組立，搬送，貯蔵，および検査を行う要素．ここでは，工作機械，組立ステーション，工具およびジグといった物理的要素だけでなく，生産スケジューリングや生産設備のNC 制御において意思決定を行う情報処理要素も含まれる．ここでは，生産設備ホロン(Manufacturing equipment holon)と呼ぶ．

(b) ワーク(Work-piece)

生産設備により素材から生産対象製品へと変換される要素．ここでは，素材，部品，製品といった物理的要素だけでなく，工程設計や生産スケジューリングの際に意思決定を行う情報処理要素も含まれる．ここでは，ワークホロン(Work-piece holon)と呼ぶ．

HMS における意思決定間の協調は，ホラーキと呼ばれる階層構造に基づき行われる．Fig. 7.3に，HMS におけるホラーキの例を示す．この生産システムは加工ショップと組立ショップで構成されており，各ショップ内には，工作機械，組立ステーション，部品，半製品などのホロンが存在する．例えば，ショップ

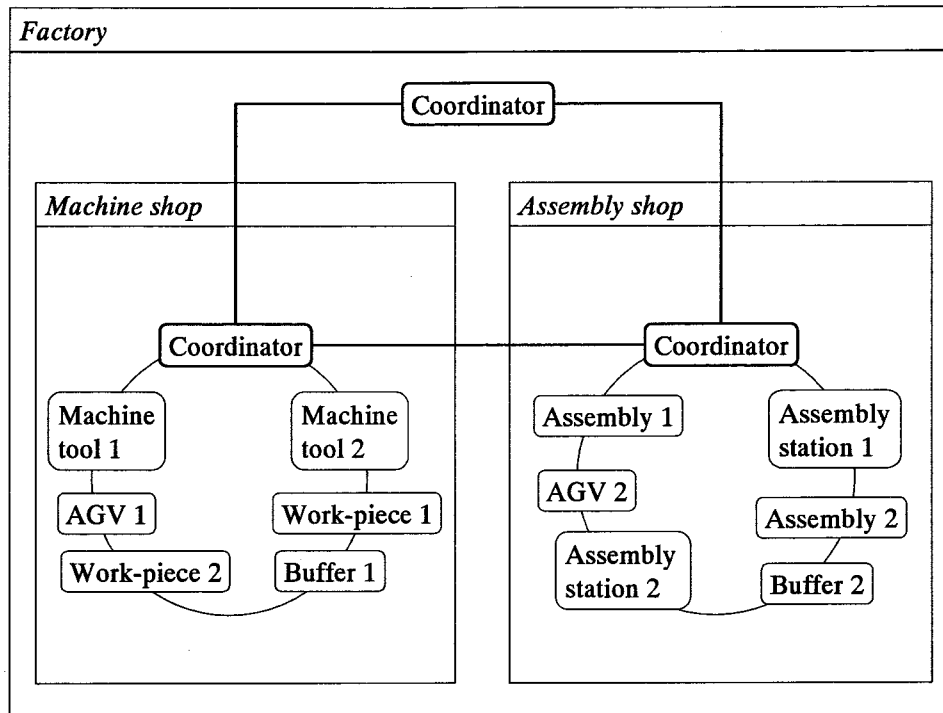


Fig. 7.3 Hierarchy of HMS

プラ内の生産スケジューリングといった下位レベルの意思決定プロセスにおいては、各ホロンが自律的に意思決定を行うとともに、ショップ内のコーディネータ(Coordinator)がホロンの意思決定間の協調を行う。また、加工ショップや組立ショップもホロンであり、工場内の生産スケジューリングといった上位レベルの意思決定プロセスにおいて意思決定を行う。さらに、工場内のコーディネータにより、ショップホロン間の協調が行われる。

7.3 HMS における生産スケジューリング

7.3.1 リアルタイムスケジューリング

本章では、HMS における生産スケジューリングの手法として、リアルタイムスケジューリングを適用する。リアルタイムスケジューリングとは、生産設備での作業の終了や、新たなワークの投入といったイベントが生産システムに発生した時点で、生産設備での次の作業やワークの投入先などの意思決定を動的に行う生産スケジューリング手法である。リアルタイムスケジューリングでは、生産設備やワークの短期間の生産スケジュールしか決定しないため、機械の故障や特急ジョブの追加といった生産開始後に起こった生産現場の状況の変化に、柔軟かつ頑健に対応することが可能である。

HMS におけるリアルタイムスケジューリングでは、生産設備ホロンとワークホロンが、自ら環境を認識

し、互いに情報を授受して、自律的に意思決定を行うことで、生産スケジュールが決定される。各ホロンの意思決定には、ヒューリスティックルール(Heuristic rule)を用いる。生産スケジュールを決定する手順を以下に示す。

STEP1: 状態の認識

以下のようなイベントが発生した場合、ホロンは自身の状態を認識する。

- (1) ワークにおける作業の終了
- (2) 生産システムへのワークの投入
- (3) 生産設備の故障の発生や故障からの復帰
- (4) ワークの優先度の変更

STEP2: 次の作業の決定

この時点で作業を行っていないホロンは、自身の意思決定基準に基づき、次に実行する作業を選択する。その情報をコーディネータへ伝達する。コーディネータは、伝達された情報に基づき、ホロンの意思決定間における競合の存在を調べる。競合が存在する場合は、コーディネータの指示により、ホロンは意思決定を変更し、競合を解消する。競合が解消した後に、選択した作業がホロンに割り付けられ、次に実行する作業が決定する。

以上の手順を、イベントの発生にあわせて繰り返すことで、生産スケジュールを決定するとともに、生産プロセスを進行させる。

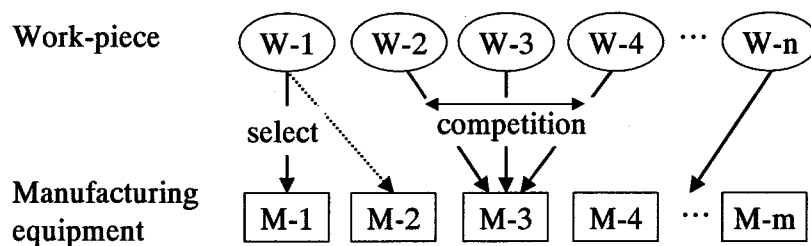
7.3.2 生産設備の代替を考慮したリアルタイムスケジューリングプロセスの提案

第5章では、生産スケジューリングに必要な情報として、以下の4つを示した。

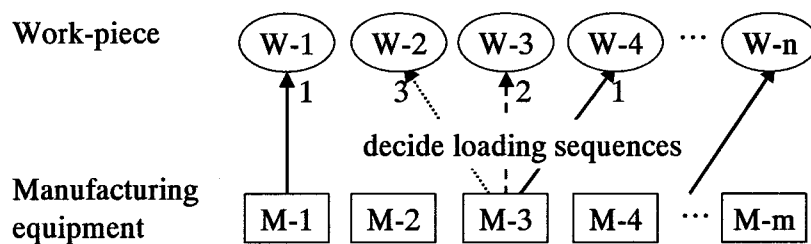
- (1) ジョブにおけるリソースの処理順序
- (2) オペレーションの処理時間
- (3) リソースにおけるジョブの処理順序
- (4) オペレーションの開始時刻および終了時刻

生産スケジューリングでは、生産工程設計により決定され、生産スケジューリングへの入力情報として与えられた(1)および(2)の情報に基づき、(3)および(4)の情報を決定する。第5章および第6章においては、(1)の各生産工程を実行できるリソース、すなわち生産設備はひとつと仮定したが、本章では、生産設備は複数の代替案を含む場合を考慮する。

生産設備の代替を考慮した場合、生産スケジューリングにおいて決定しなければならない情報は以下の3つである。



(a) Selection of manufacturing equipment by work-piece



(b) Decision of loading sequences of work-piece by selected manufacturing equipment

Fig. 7.4 Real-time scheduling process of HMS

- (a) 各工程を実行する生産設備の選択
- (b) 生産設備におけるワークの処理順序
- (c) 作業の開始時刻および終了時刻

7.3.1項の STEP2 において、各ホロンは以下の手順で生産スケジュールを決定する。

STEP2-1: ワークホロンによる生産設備ホロンの選択

Fig. 7.4(a) に示すように、STEP1 のイベントが発生した時点で作業を行っていない全てのワークホロンは、生産設備の代替を考慮した上で、自身の意思決定基準に基づき、次に実行する生産設備ホロンを選択する。

STEP2-2: 生産設備ホロンによるワークホロンの処理順序の決定

Fig. 7.4 (b) に示すように、STEP2-1 において、複数のワークホロンから選択された生産設備ホロンは、自身の意思決定基準に基づき、それらのワークホロンの処理順序を決定する。選択したワ

ークホロンがひとつのみの場合は、それを次の作業として決定する。

STEP2-3: 生産設備ホロンによる作業の開始, 終了時刻の決定

生産設備ホロンは, 作業の開始および終了時刻を決定する。

この手順では, 次の作業として, 複数のワークホロンが, 同時に, 同一の生産設備ホロンを選択するという競合を, 選択された生産設備ホロンが解消する。すなわち, ここでは, 生産設備ホロンがコーディネータの役割を行っている。

7.4 HMS のシミュレーションのためのモデルリポジトリ

本章では, HMS のシミュレーションシステムのプロトタイプを開発し, それを用いて, HMS におけるリアルタイムスケジューリングの柔軟性や頑健性を検証する。そこで, まず, 第 5 章で提案した生産スケジュールモデルを拡張して, HMS のシミュレーションのためのモデルを提案する。以下, これを HMS モデルと呼ぶ。

7.4.1 HMS モデルの構成要素

HMS モデルの構成要素を以下のオブジェクトで表す。また, OMT(Object Modeling Technique) 手法 [Rumbaugh 91] のオブジェクト図を用いて, オブジェクト間の構造をFig. 7.5に示す。

(1) ジョブ(Job)

ワークホロンの物理的処理部を表すオブジェクト。以下, これを Job と呼ぶ。部品やロットなどは, Job クラスのサブクラスとして表される。

(2) リソース(Resource)

生産設備ホロンの物理的処理部を表すオブジェクト。以下, これを Resource と呼ぶ。工作機械やAGV(Automated Guided Vehicle), セル, ショップなどは, Resource クラスのサブクラスとして表される。

(3) オペレーション(Operation)

生産設備ホロンにおける作業を表すオブジェクト。以下, これを Operation と呼ぶ。フライス加工やドリル加工などは, Operation クラスのサブクラスとして表される。

(4) プロセス(Process)

ワークホロンにおける工程を表すオブジェクト。以下, これを Process と呼ぶ。荒加工や穴加工な

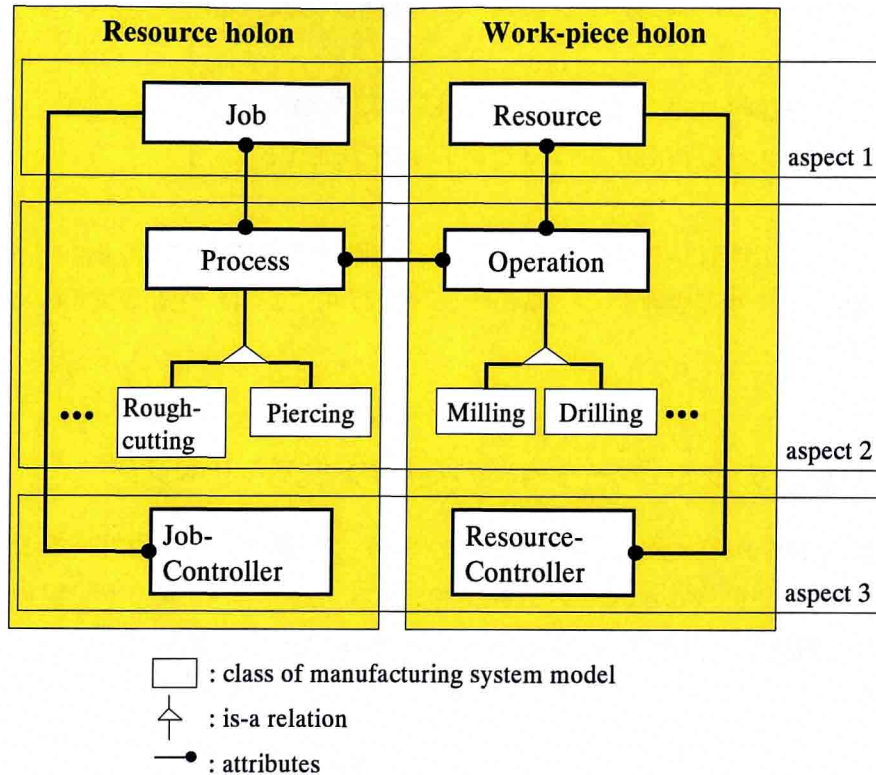


Fig. 7.5 Model repository of HMS

どは, **Process** クラスのサブクラスとして表される.

(5) ジョブコントローラ (Job controller)

ワークホロンの情報処理部を表すオブジェクト. 以下, これを **Job-controller** と呼ぶ. ヒューリスティックルールなどの意思決定基準を持つ.

(6) リソースコントローラ (Resource controller)

生産設備ホロンの情報処理部を表すオブジェクト. 以下, これを **Resource-controller** と呼ぶ. ヒューリスティックルールなどの意思決定基準を持つ.

ワークホロンにおけるひとつの工程が多数の代替工程を持つこと, および生産設備ホロンがフレキシブルな生産機能を持つことから, **Job** とそれを加工できる **Resource** との関係を, **Process** および **Operation** の2つのオブジェクトを介して関係付けている. これにより, 生産設備の代替の関係を表すことができる. また, ワークホロンおよび生産設備ホロンによる意思決定は, **Job-controller** および **Resource-controller** が, 自身の持つ意思決定基準に基づき行う. これにより, ホロンの自律的な意思決定の行為を表すことができる.

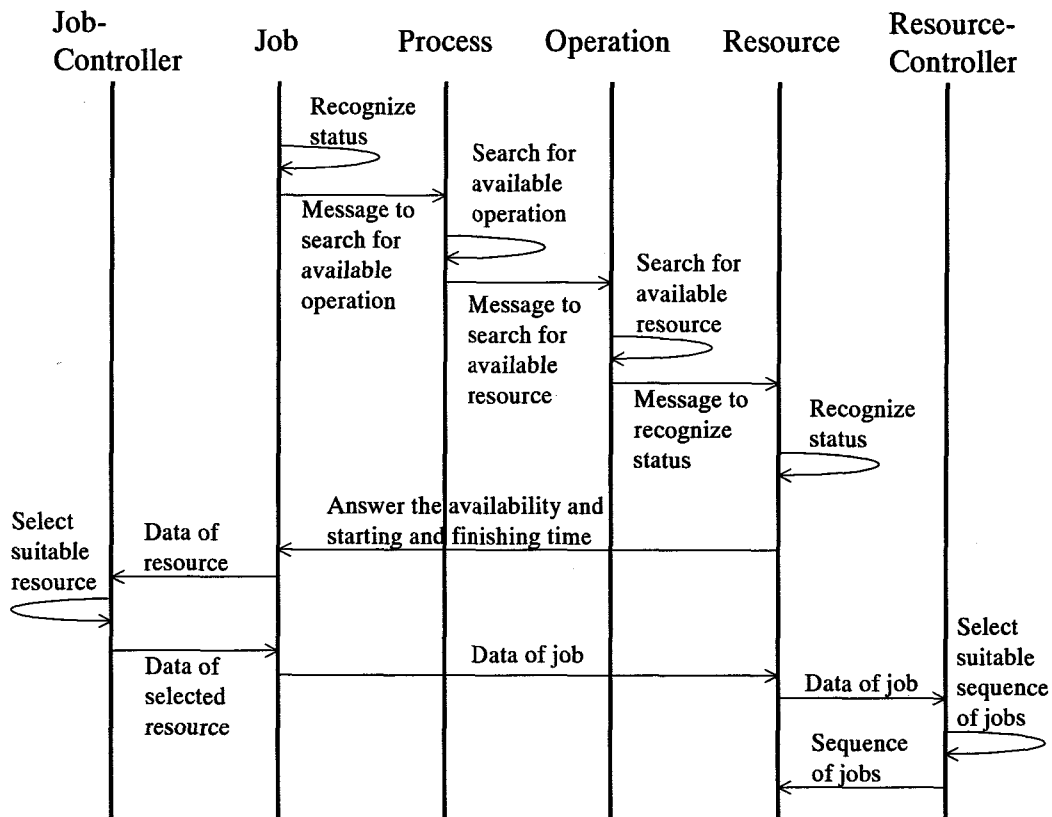


Fig. 7.6 Real-time scheduling process of HMS models

7.4.2 HMS モデルによるリアルタイムスケジューリングプロセス

HMS モデルにより、生産スケジュールを作成する過程を以下に示す。また、Fig. 7.6に示すように、OMT手法のイベントトレース図を用いて、生産スケジューリングプロセスを表す。図の縦軸が時間を、矢印がオブジェクト間で授受される情報を表す。

STEP1: Job によるワークホロンの状態の認識

Job は、ワークホロンの作業の終了や加工の優先度の変更といった状態の変化を認識し、次の工程を表す Process に、次工程が実行可能な全ての Operation を検索させる命令を送る。

STEP2: Process による Operation の選択

Process は、次工程が実行可能な全ての Operation を検索する。さらに、選択した Operation に、次工程の作業が実行可能な全ての Resource を検索させる命令を送る。

STEP3: Operation による Resource の選択

Operation は、次工程の作業が実行可能な全ての Resource を検索する。さらに、選択した Resource に、生産設備ホロンの状態を認識させる命令を送る。

STEP4: Resource による生産設備ホロンの状態の認識

Resource は、生産設備ホロンの状態を認識して、作業の実行可能性、および作業の開始、終了の予定時刻を調べ、その情報を Job に伝達する。

STEP5: Job-controller による Resource の選択

Job は、Resource からの情報を Job-controller に伝達する。Job-controller は、自身の持つ意思決定基準に基づき、次工程を実行する Resource を選択し、Resource に Job の情報を伝達する。

STEP6: Resource-controller による Job の処理順序の決定

Resource は Job-controller から受け取った Job の情報を Resource-controller に伝達する。Resource-controller は、自身の持つ意思決定基準に基づき、Job の処理順序を決定し、その情報を Resource に伝達する。

以上の手順を、ワークホロンのイベントの発生にあわせて繰り返すことで、生産スケジュールを決定するとともに、生産プロセスを進行する。

7.5 シミュレーションシステムの実装

7.5.1 シミュレーションシステムの基本構成

7.4節で提案した HMS モデルに、シミュレーションを実行するための機能を追加し、HMS のシミュレーションシステムのプロトタイプを開発した。本システムの開発には、オブジェクト指向言語 Smalltalk を使用した。シミュレーションシステムの構成要素を以下に示す。また、要素間の関係を Fig. 7.7 に示す。

(1) Holonic manufacturing system model

7.4節で提案した HMS モデルの構成要素を表すプログラムモジュールを実装した。

(2) Event and time manager

シミュレーション上の時間を管理し、シミュレーションを進行させる機能を持つプログラムモジュールを実装した。

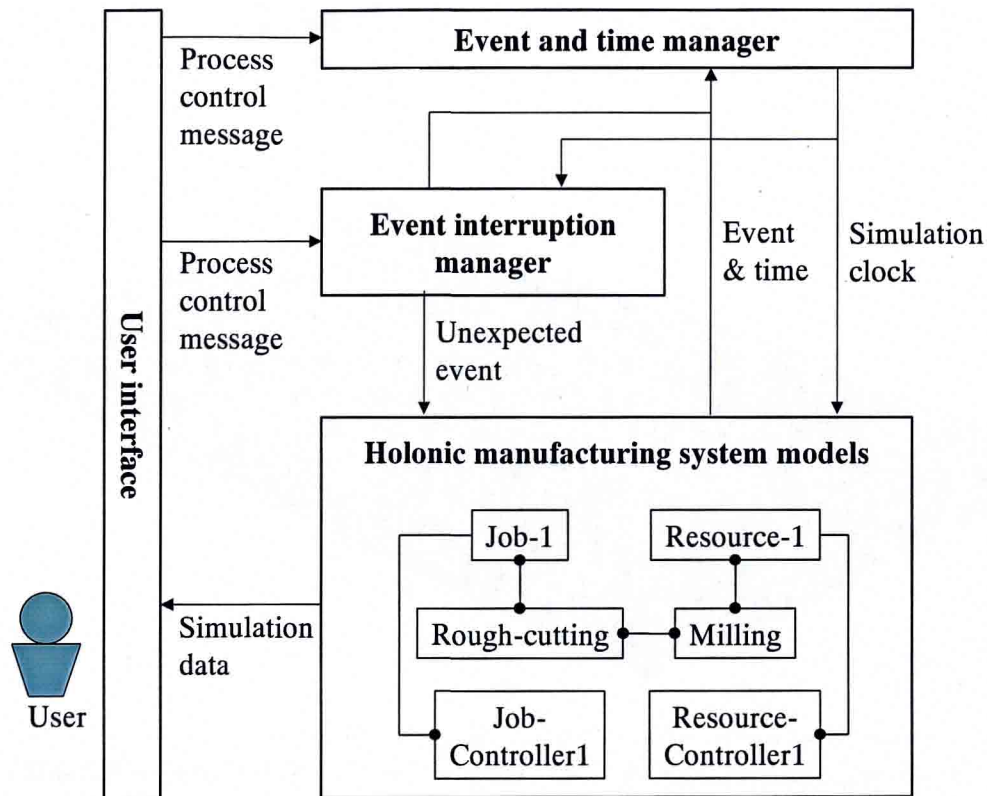


Fig. 7.7 Basic structure of object oriented simulation system of HMS

(3) Event interruption manager

シミュレーションの開始, 終了や, 機械の故障といった, ユーザが発生させた特別なイベントを実行する機能を持つプログラムモジュールを実装した.

(4) User interface

ユーザが特別なイベントの入力を行うためのインタフェースや, シミュレーションの結果を表示するためのインタフェース等を生成するプログラムモジュールを実装した. 本章では, 第5章において開発したガントチャート(Gantt chart)を表示するオブジェクトを利用するだけでなく, シミュレーションにおけるHMSの状態の変化を3次元の動画で表示するオブジェクトを実装した. 3次元の動画の表示例をFig. 7.8に示す. これにより, ユーザが, シミュレーションにおけるホロンの動きを確認することが容易になる.

7.5.2 シミュレーションにおける時刻の進行方法

シミュレーションにおける時刻の進め方として, 本章では, 集中時刻管理法のひとつである, タイムホ

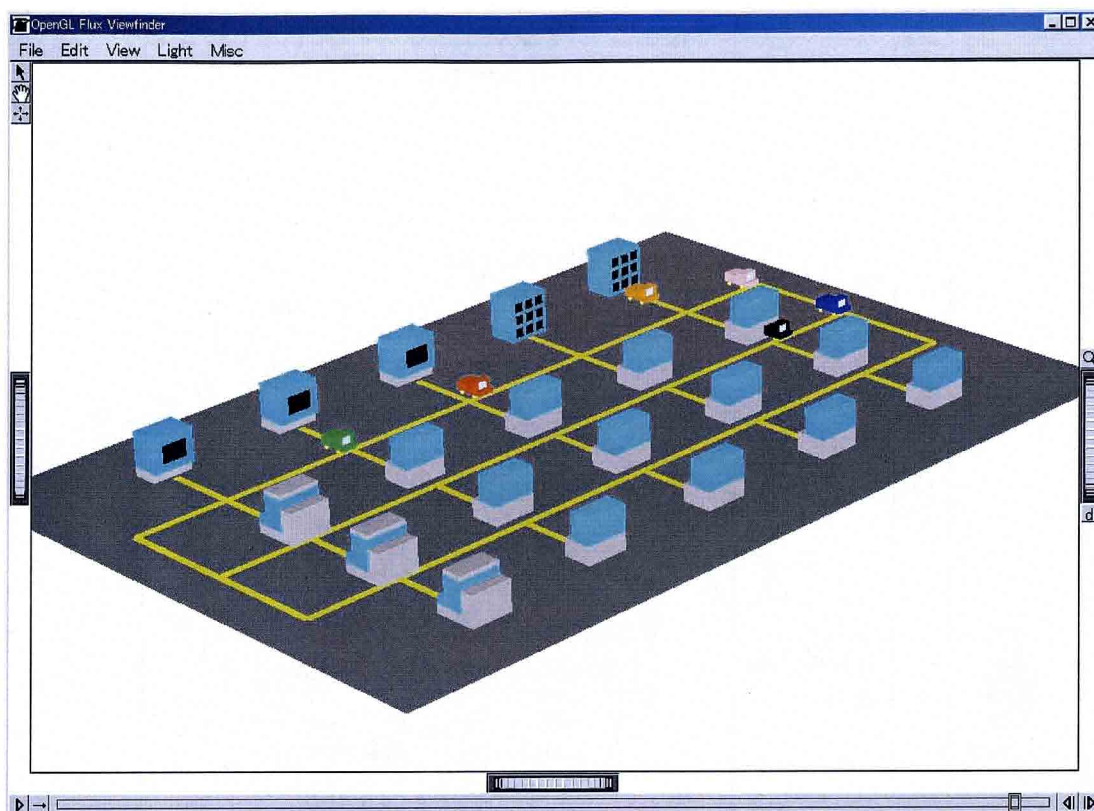


Fig. 7.8 3-dimensional animation interface

ール法 [加藤 94] を用いた。タイムホイール法は、集中的に時刻を管理するスーパーバイザとなるプログラムモジュールが、1 イベント処理毎に、全てのモジュールにおける最早生起イベントの生起時刻を求め、シミュレーションの大域時刻を進行させる手法である。

以下に、7.5.1項で定義したシミュレーションシステムの構成要素を用いて、シミュレーションの時刻を進める手順を示す。

STEP1: 生起イベントの発生時刻の伝達

Event and time manager の依頼により、Holonic manufacturing system model および Event interruption manager の全てのモジュールは、次の生起イベントの生起時刻を Event and time manager に伝達する。

STEP2: 最早生起イベントの発生時刻の計算

Event and time manager は、伝達された生起時刻の中で、最も早い時刻をシミュレーションにおける次の時刻とし、Holonic manufacturing system model および Event interruption manager の全てのモジュールに、その時刻を伝達する。

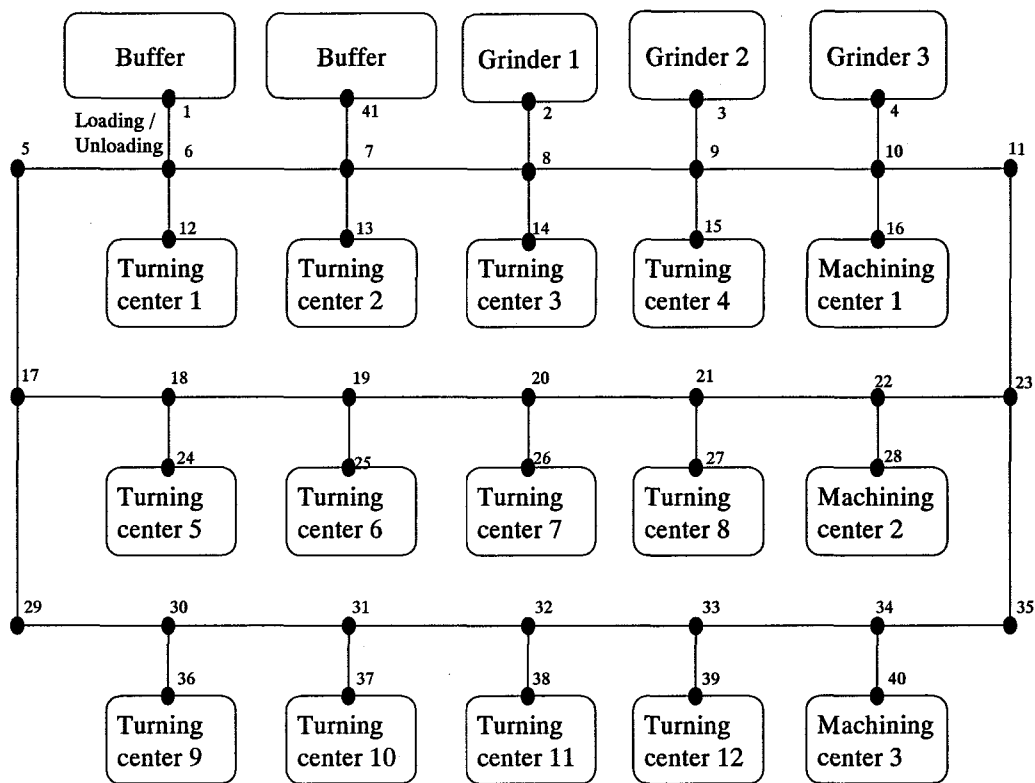


Fig. 7.9 Holonic manufacturing system for simulation

STEP3: シミュレーションの時刻の更新

時刻を受け取った各モジュールは、その時刻までシミュレーションを進行する。この時刻において生起するイベントがあれば、それを実行する。すなわち、Job や Resource における意思決定を行う。

7.6 ケーススタディ

7.6.1 シミュレーションの対象とする生産システム

本章では、Fig. 7.9に示すような機械加工システムにおけるリアルタイムスケジューリングのシミュレーションを行う。機械加工システムの構成を以下に示す。

- (1) ターニングセンタ (Turning Center): 12 台
- (2) マシニングセンタ (Machining Center): 3 台
- (3) グラインダ (Grinder): 3 台

(4) AGV(Automated Guided Vehicle) : 6 台

(5) バッファ(Buffer) : 2 台

(6) 部品 : 30 個

部品は、軸対称部品を主対象とし、一部、角物部品を含む。バッファには、素材、加工途中部品および完成品を格納する。ターニングセンタ、マシニングセンタ、グラインダおよび AGV は生産設備ホロンであり、これらは、HMS モデルの Resource のサブクラスとしてモデル化する。また、部品はワークホロンであり、HMS モデルの Job を用いてモデル化する。

7.6.2 適用するディスパッチングルール

生産スケジューリングの評価指標は、生産システム全体の総処理時間の最小化とする。ワークホロンおよび生産設備ホロンは、加工作業終了などのイベントが発生した時点で、自身の意思決定基準に基づき、次に実行する行為を決定する。そこで、本ケーススタディでは、各ホロンの意思決定基準として、以下のディスパッチングルールを準備した。

(1) ワークホロンによる生産設備ホロンの選択のためのディスパッチングルール

Job-controller のオブジェクトに、以下のディスパッチングルールに基づいて意思決定を行う機能を実装した。

(a) SPT (Shortest Processing Time)

加工や搬送などの処理時間が最小の生産設備ホロンを選択する。

(b) STRT (Shortest Total Running Time)

加工や搬送など、実際に稼動した時間の総和が最小の生産設備ホロンを選択する。

(2) 生産設備ホロンによるワークホロンの処理順序決定のためのディスパッチングルール

Resource-controller のオブジェクトに、以下のディスパッチングルールに基づいて意思決定を行う機能を実装した。

(a) SPT/TWKR (Shortest Processing Time / Total WorK Remaining)

加工や搬送などの現作業の処理時間を、現作業を含む残り作業の全処理時間で割った値が最小のワークホロンを選択する。

(b) LRM

現作業を除く残り作業の全処理時間が最大のワークホロンを選択する。

SPT/TWKRとLRMはどちらも、現作業の処理時間が小さく、残り作業の全処理時間の大きなワークホロンを選択する特性があるが、SPT/TWKRは現作業の処理時間をより重視し、LRMは残り処理時間をより重視する特徴がある。しかし、その性能の優劣は、対象となる生産システムの特性により異なる [大場 99]。なお、SPT/TWKRとLRMにおいて、ワークホロンの生産設備ホロンへの割付に代替が考慮できる場合には、同一工程を表す全ての処理の中で最大の処理時間を、その工程における残り作業の処理時間とみなす。

7.6.3 リアルタイムスケジューリングの試行

7.6.2項のディスパッチングルールを用いて、7.6.1項の機械加工システムにおけるリアルタイムスケジューリングのシミュレーションを行った。まず、生産開始後に、生産現場の状況が変化しない場合についてシミュレーションを行った。次に、生産開始後、故障により生産設備が、ある一定時間使用できない状態になる場合についてシミュレーションを行った。各シミュレーションの結果について考察する。

(1) 状況が変化しない場合

ワークホロンによる生産設備ホロンの選択のためのディスパッチングルールとして、SPT または STRT を適用し、生産設備ホロンによるワークホロンの処理順序決定のためのディスパッチングルールとして、SPT/TWKR または LRM を適用した。各試行を以下のケース 1、ケース 2、ケース 3 およびケース 4 とし、ディスパッチングルールの組をTable 7.1に整理する。各ケースにおいて全作業が終了するまでの時刻、すなわち総処理時間をTable 7.2に示す。

Table 7.1 Experimental condition for simulation

	Rule of Work-piece Holon	Rule of manufacturing equipment holon
Case 1	SPT	SPT/TWKR
Case 2	SPT	LRM
Case 3	STRT	SPT/TWKR
Case 4	STRT	LRM

Table 7.2 Total processing time at case studies

	SPT/TWKR	LRM
SPT	15703 (case1)	16778 (case2)
STRT	16770 (case3)	18970 (case4)

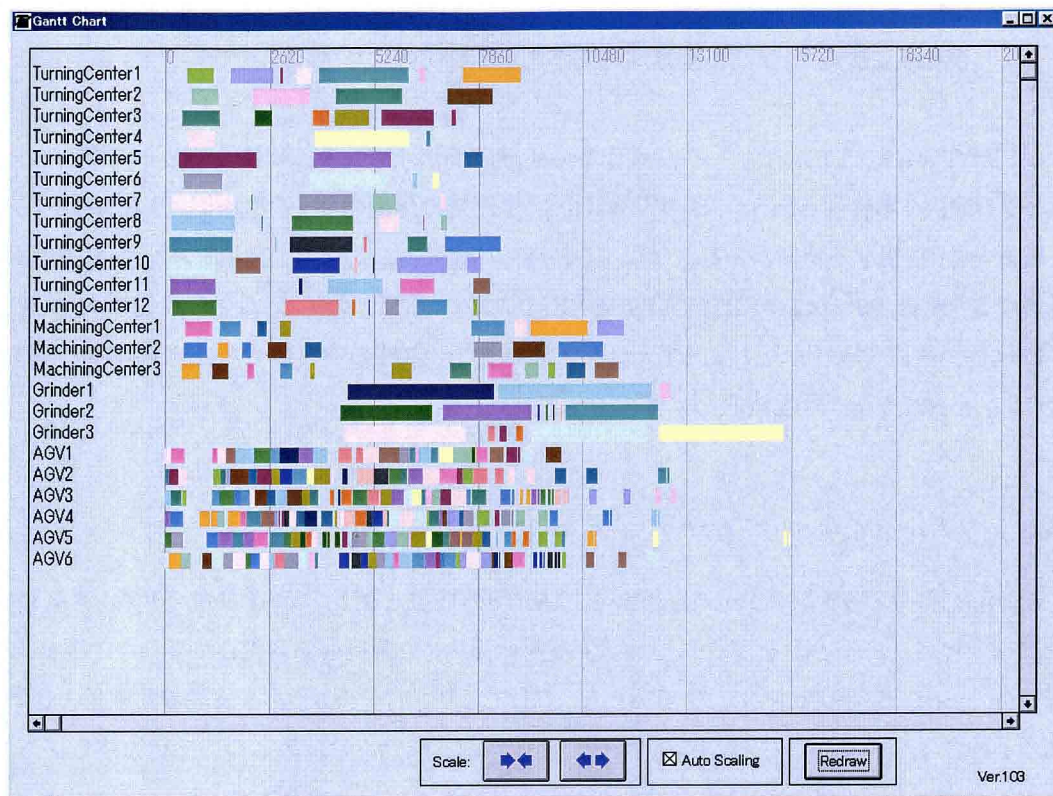


Fig. 7.10 Simulation result of real-time scheduling of case 1

この結果より、ワークホロンのディスパッチングルールは、STRT よりも SPT を用いる方が、総処理時間が短くなるのがわかる。すなわち、ワークの処理時間が最小の生産設備を選択するルールを用いた方が、総処理時間は短くなることを表しており、妥当な結果であると言える。また、この生産システムの場合、生産設備ホロンのディスパッチングルールには、LRM よりも SPT/TWKR を用いた方が総処理時間は短くなるのがわかった。

ケース 1 におけるリアルタイムスケジューリングの結果をFig. 7.10のガントチャートで表す。横軸は時間の経過を、縦軸は各生産設備を表し、図中の横棒が各生産設備で実行された加工プロセスの開始、終了時刻を表す。

(2) 生産設備が故障する場合

(1)のケーススタディにおいて、最も総処理時間の短かったケース 1 の例に対し、生産開始後、機械の故障を発生させ、生産設備がある一定時間使用できない状態になるようにして、シミュレーションを行った。ケース 1 の場合と同様、ワークホロンのディスパッチングルールには SPT を、生産設備ホロンのディスパッチングルールには SPT/TWKR を用いた。故障が起こる生産設備は、ケース 1 において比較的稼働率の高い生産設備 1 台を選択した。ここでは、以下の 3 つのケースについて、シミュレーションを行った。各試行をケース 5、ケース 6、およびケース 7 とし、故障の条件をTable

7.3に整理する。なお、生産設備が故障により使用できない時間を5000とした。これは、ケース1での総処理時間の約1/3に相当する。

各ケースにおけるリアルタイムスケジューリングの結果として、総処理時間をTable 7.4に示し、ガントチャートをFig. 7.11, Fig. 7.12およびFig. 7.13に表す。

Table 7.3 Experimental condition of simulation for failure

	Manufacturing equipment holon having failure	Period during which holon is unable to work because of failure
Case 5	TurningCenter1	1221 ~ 6221
Case 6	Grinder3	7567 ~ 12567
Case 7	AGV6	2021 ~ 7021

Table 7.4 Total processing time at case studies for failure

	Manufacturing equipment holon having failure	Total processing time
Case 1	(Nothing)	15703
Case 5	TurningCenter1	16901
Case 6	Grinder3	17426
Case 7	AGV6	16019

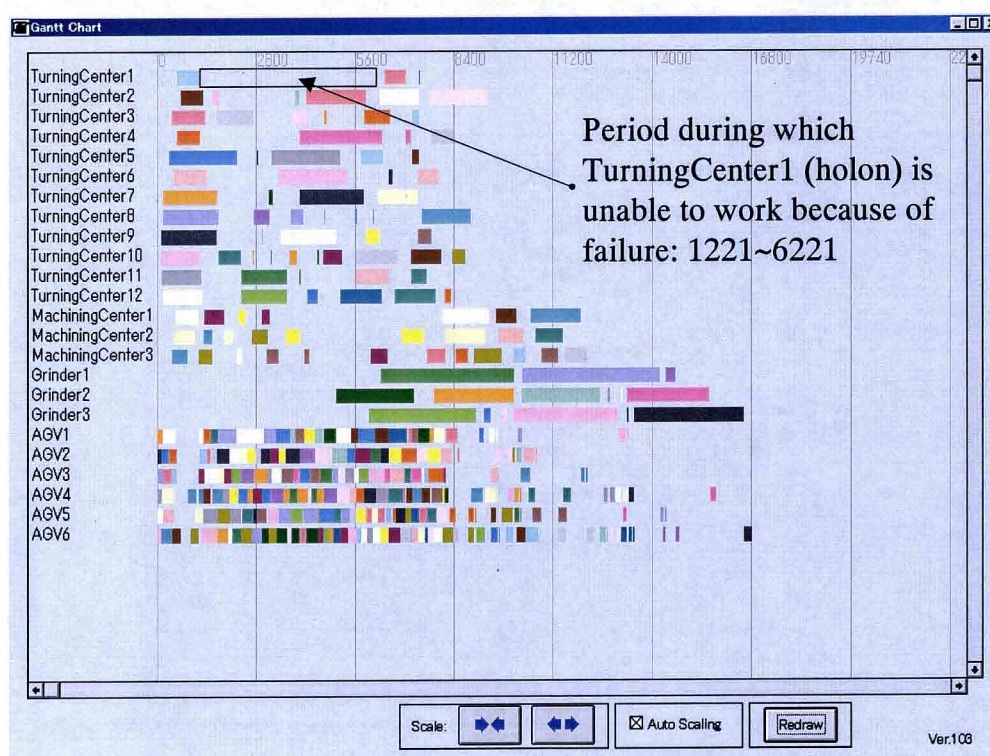


Fig. 7.11 Simulation result of real-time scheduling of case 5

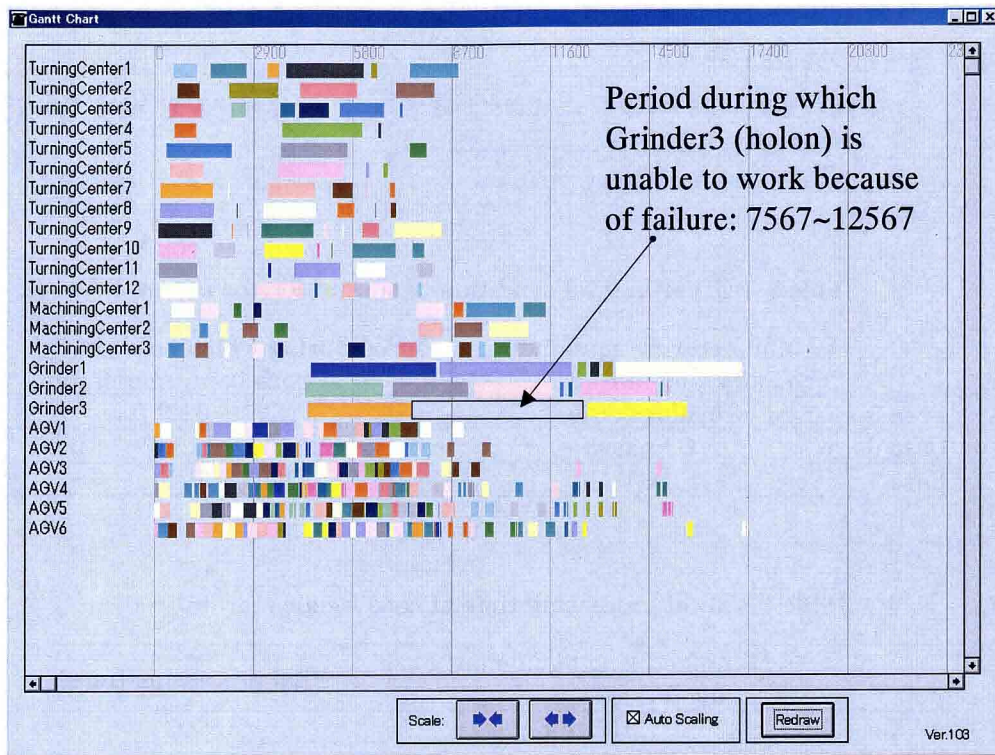


Fig. 7.12 Simulation result of real-time scheduling of case 6

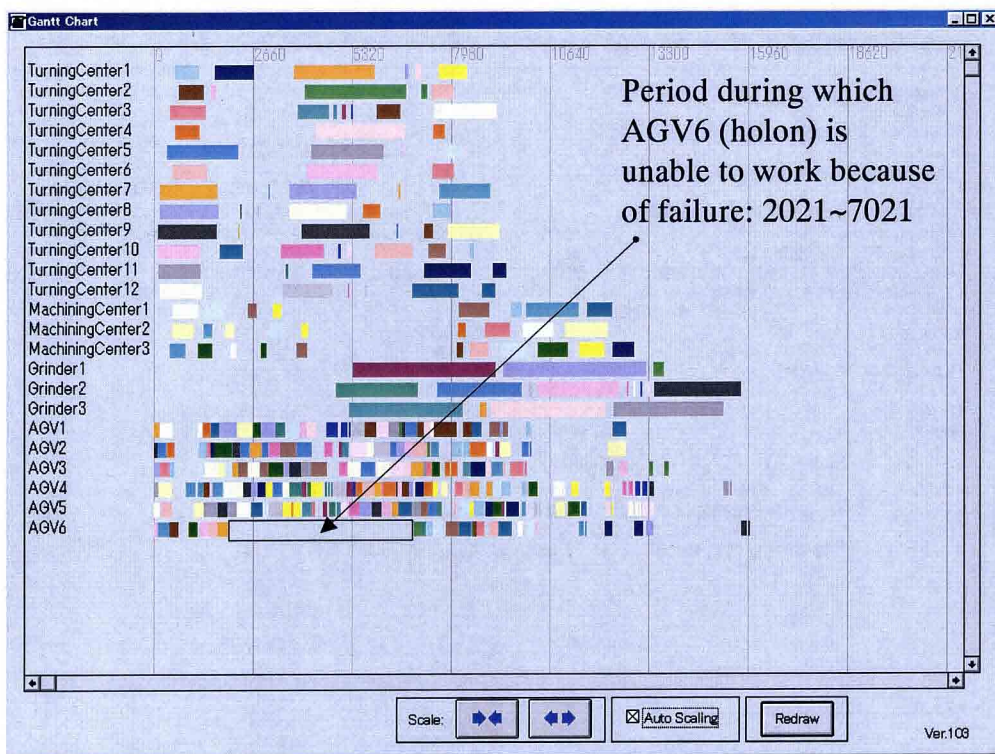


Fig. 7.13 Simulation result of real-time scheduling of case 7

以上のシミュレーションの結果により、生産設備に予期せぬ故障が起きた場合でも、各ホロンの持つプログラムを変更することなく、リアルタイムに生産スケジュールが生成できることを示した。また、ケース5、ケース6、およびケース7のいずれにおいても、総処理時間の延びは、故障の起こらないケース1に比べてそれ程大きくはない。これは、ワークホロンが、故障の起こった生産設備ホロンの故障回復を待つのではなく、ワークホロンの自律的な意思決定により、代替となる生産設備ホロンが適宜、選択されているからであると考えられる。このように、HMS のリアルタイムスケジューリングは、状況の変化に対して、柔軟性や頑健性が高いという特長を示すことができた。

7.7 結言

本章では、HMS におけるリアルタイムスケジューリングの柔軟性や頑健性を検証するために、第5章で提案した生産スケジュールモデルを拡張して、HMS のモデル化を行うとともに、HMS のシミュレーションシステムのプロトタイプを開発し、それを簡単な例に適用した。以下に本章の内容をまとめる。

- (1) HMS の定義と基本構造を示した。HMS の構成要素は、機能的な観点から、物理的処理部と情報処理部に分け、さらに、製造プロセスの観点から、生産設備ホロンとワークホロンに分けることを示した。
- (2) HMS のリアルタイムスケジューリング手法として、各ホロンが自身の状態を認識し、次に行う作業を自律的に決定することで、リアルタイムに生産スケジュールを決定する手順を提案した。
- (3) HMS のシミュレーションを行うために、HMS のモデル化を行った。まず、ワークとそれを加工できる生産設備の代替の関係を表すために、第5章で提案した生産スケジュールモデルの基本的要素を拡張し、ジョブ、リソース、オペレーション、プロセスの4つのオブジェクトを提案した。さらに、各ホロンの自律的な意思決定によるリアルタイムスケジューリングプロセスを表すために、ジョブコントローラとリソースコントローラの2つのオブジェクトを提案した。
- (4) (3)で示した6つのオブジェクト間での情報授受により、リアルタイムスケジューリングを行う手順、すなわちリアルタイムスケジューリングプロセスを提案した。
- (5) (3)で示した6つの基本的なオブジェクトに、シミュレーションを実行するためのプログラムモジュールを追加して、HMS のシミュレーションシステムのプロトタイプを開発した。本システムの開発には、オブジェクト指向言語の Smalltalk を使用した。
- (6) 開発したシミュレーションシステムを簡単な例に適用した。ここでは、生産現場の状況が変化しない場合に総処理時間が最も短くなった試行の条件で、生産設備に故障を発生させてシミュレーションを行い、HMS におけるリアルタイムスケジューリングの柔軟性や頑健性の特長を示した。

参考文献

- [Iwata 94] Iwata, K., Random manufacturing system: a new concept of manufacturing system for production to order, *Annals of the CIRP*, 43-1 (1994), 379-384.
- [Koestler 78] Koestler, A., Janus, Hutchinson & Co. (1978); 田中・吉岡 訳, ホロン革命, 工作舎 (1983).
- [Rumbaugh 91] Rumbaugh, J., et al., Object oriented modeling and design, Prentice hall (1991); 羽生田 訳, オブジェクト指向方法論 OMT モデル化と設計, トップラン (1992).
- [Russell 95] Russell, S. and Norvig, P., *Artificial interigence – a modern approach*, Prentice hall (1995); 古川 監訳, エージェントアプローチ 人工知能, 共立出版 (1997).
- [Sugimura 97] Sugimura, N., Moriwaki, T., Hozumi, K. and Shinohara, Y., Modeling of holonic manufacturing system and its application to real-time scheduling, *Manufacturing systems*, 25-4 (1997), 345-352.
- [Suyoto 99] Suyoto, R., Sugimura, N. and Kokubun, A., A study on process planning system for holonic manufacturing systems – selection of machining sequences based on GA and DP –, *Proc. of the 32th CIRP international seminar on manufacturing systems* (1999), 571-578.
- [Tharumarajah 96] Tharumarajah, A., Wells, A. J., and Nemes, L., Comparison of the bionic, fractal and holonic manufacturing system concepts, *International journal of computer integrated manufacturing*, 9-3 (1996), 217-226.
- [Warnecke 93] Warnecke, H. J., *The fractal company*, Springer-verlag (1993).
- [Wyns 96] Wyns, J., et al., Workstation architecture in holonic manufacturing systems, *Proc. of the 28th Int. seminar on manufacturing systems* (1996), 220-231.
- [大倉 96] 大倉・上田, 生物指向型生産システムにおける簡単なモデルを使った生産スケジューリング問題の一解法, 日本機械学会論文集, 62-595, C (1996), 429-435.
- [大場 99] 大場・江口・岡崎, 生産特性適合型複合ディスパッチングルールの提案, 1999年度精密工学会秋季大会学術講演会講演論文集 (1999), 2.

- [沖野 95] 沖野, 生物型システムの構想—擬生物化メタファーによるシステム構築論, システム制御情報学会論文誌, 8-8 (1995), 381-390.
- [加藤 94] 加藤 ほか 47 名, 生産システムの計画・評価手法に関する研究成果報告書, IMS センター (1994).
- [鳩野 94] 鳩野・田村, 分散型リアルタイム生産スケジューリング, 計測と制御, 33-7 (1994), 575-580.
- [馬 97] 馬・大場・李, 自律分散型リアルタイムスケジューリングのためのカラーペトリネットによるモデル化手法, 日本機械学会論文集, 63-616, C (1997), 324-331.
- [松本 99] 松本・加藤・長舟・若松・白瀬・荒井, アクティブデータベースを用いた自律分散型生産システムの開発, 日本機械学会論文集, 65-630, C (1999), 405-411.
- [森脇 92] 森脇 ほか 3 名, 自律分散型生産システムの設計と運用に関する研究(基本構成と生産スケジューリングへの適用), 日本機械学会論文集, 58-549, C (1992), 340-345.

第 8 章 結論と展望

8.1 結論

近年, オブジェクト指向を用いた分散的な生産システムのライフサイクルサポートに対する注目が高まっている. 本研究は, 分散環境において, 生産システムの設計から運用までの過程における複数のアプリケーションをコンカレントに実行するために, オブジェクト指向技術を用いた生産システムの体系的なモデル化のアーキテクチャを提案するとともに, 生産システムの設計および運用の各フェーズにおいて, 生産システムモデルとその利用手法を提案し, コンカレントな分散処理における有効性を示した. 本研究で得られた成果を各章毎にまとめる.

(1) 第 2 章のまとめ

第 2 章では, 生産システムの設計から運用までの過程における多数のアプリケーションモデルを生成するとともに, これらのモデル間の整合性を管理する手法を提案した. 以下に第 2 章の内容をまとめる.

- (a) 生産システムモデルの構成要素を分類するための基準として, 視点, 抽象性, 範囲の 3 つの軸を提案した. また, これらの軸に基づいて, 生産システムモデルの原型を格納するモデルリポジトリとその構成を提案した.
- (b) オブジェクト指向の考え方に基づいて, 生産システムモデルの原型を表現するとともに, 上記 3 つの軸に基づく原型間の関係を表現する手法を提案した.
- (c) モデルの原型間の関係に基づき, アプリケーションの異なるモデルの間で, 整合性を保証すべきモデルの構成要素を検出する方法を提案するとともに, オブジェクト指向言語の Smalltalk を用いてプロトタイプシステムを開発し, ケーススタディを行った.

(2) 第 3 章のまとめ

第 3 章では, 生産システムの設計プロセス(MSE プロセス)の支援システムを開発するための準備として, 実際の MSE プロセスを記述し, 分析して, MSE プロセスの問題点を明らかにした. 以下に第 3 章の内容をまとめる.

- (a) 第 2 章で提案した生産システムモデルを拡張して, MSE プロセスのモデル化手法を提案し, MSE プロセスを記述するためのツールを提案した.

- (b) 自動車の回転機器部品を製造する加工および組立工場の設計を具体的に実行すると同時に、その MSE プロセスを記述し、分析した。ここでは、生産システムの設計における総計約 180 個のエンジニアリングアクティビティ(EA)および EA 間の情報授受の関係を記述し、その内容を分析した。
- (c) (b)の分析の結果として、生産システムの設計期間の短縮化には、複数の設計者間での設計情報の授受、設計変更による出力情報の値の変更、および事例の蓄積と利用における問題の解決が重要であることを明らかにした。

(3) 第 4 章のまとめ

第 4 章では、第 3 章で得られた MSE プロセスの問題点から、MSE プロセス支援のための要求される機能を整理し、オブジェクト指向言語の Smalltalk を用いて、MSE プロセス支援ツールのプロトタイプを開発した。以下に第 4 章の内容をまとめる。

- (a) MSE プロセスを記述する機能と、設計変更を支援する機能を持つ MSE プロセス支援ツールのプロトタイプを開発した。このツールでは、複数の設計者が、異なる計算機上からネットワークを介して、コンカレントに MSE プロセスを記述、参照できるだけでなく、設計変更の際には、既に設計が終了した出力情報の値の変更が MSE プロセス全般にわたって及ぼす影響を求めることができる。
- (b) 開発した MSE プロセス支援ツールを、簡単な生産システムの設計例に適用し、その有効性を検証した。ここでは、設計変更の及ぼす影響が小さい情報から設計変更を検討することで、設計変更による設計期間の長期化を抑制できることを示した。

(4) 第 5 章のまとめ

第 5 章では、第 2 章で提案した生産システムモデルの表現方法に基づき、生産スケジュールモデルの基本的な構成要素を提案するとともに、そのモデルを用いて、階層的な複数の生産スケジュール間の整合性を管理する方法を提案した。以下に第 5 章の内容をまとめる。

- (a) 第 2 章で提案した生産システムモデルの表現方法に基づき、生産スケジュールモデルの基本的要素を表すジョブ、リソース、オペレーション、およびオペレーションの順序関係という 4 つのオブジェクトと、階層的な生産スケジュールモデル間の関係を表すジョブの包含関係、リソースの包含関係、およびオペレーションの包含関係という 3 つのオブジェクトを提案した。
- (b) (a)の 7 つのオブジェクトを用いて生産スケジュールモデルを作成し、そのモデルを用いて、生産スケジュールを作成する手順、および階層的な複数の生産スケジュール間の整合性を管理する手順を提案した。
- (c) オブジェクト指向言語の Smalltalk を用いて、生産スケジューリングのためのモデルリポジトリと

モデリングシステムのプロトタイプを開発した。これを用いて、階層的な2つの生産スケジュールを整合的に作成し、生産スケジュールの変更に対する整合性管理の過程の一例を示した。

(5) 第6章のまとめ

第6章では、第5章で提案した生産スケジュールモデルの構成要素を用いて、階層分散型生産システムにおける複数の生産スケジューリングシステムで、コンカレントにリスケジューリングを行う方法を提案した。以下に第6章の内容をまとめる。

- (a) 階層的な複数の生産スケジュール間の整合性を保持しながら、短い時間で、最上位階層の納期を満たすように、全ての階層の生産スケジューリングシステムで、同時かつ並列にリスケジューリングを行うコンカレント・アプローチを提案した。
- (b) 第5章で開発した生産スケジュールのモデリングシステムの機能を拡張して、リスケジューリングシステムのプロトタイプを開発し、3階層のリスケジューリング問題に適用した。100例の数値実験を行った結果、リスケジューリングを順次行うシーケンシャル・アプローチよりも、同時かつ並列に行うコンカレント・アプローチの方が短い時間か、もしくは同一の時間でリスケジューリングが終了できることを示し、コンカレント・アプローチの有効性を検証した。

(6) 第7章のまとめ

第7章では、自律分散型生産システムのひとつであるホロニック生産システム(HMS)について、そのリアルタイムスケジューリングの柔軟性や頑健性を検証するために、第5章で提案した生産スケジュールモデルを拡張して、HMSのモデル化を行うとともに、HMSのシミュレーションシステムのプロトタイプを開発し、それを簡単な例に適用した。以下に第7章の内容をまとめる。

- (a) HMSの定義と基本構造を示し、ホロンによるリアルタイムスケジューリングの手順を提案した。
- (b) HMSのモデル化を行った。まず、生産設備の代替の関係を表すために、第5章で提案した生産スケジュールモデルを拡張し、ジョブ、リソース、オペレーション、プロセスの4つのオブジェクトを提案した。さらに、各ホロンの自律的な意思決定によるリアルタイムスケジューリングプロセスを表すために、ジョブコントローラとリソースコントローラの2つのオブジェクトを提案した。
- (c) (b)の6つのオブジェクトを用いたリアルタイムスケジューリングプロセスを提案した。
- (d) オブジェクト指向言語の Smalltalk を用いて、HMSのシミュレーションシステムのプロトタイプを開発した。このシミュレーションシステムを用いて、生産設備に故障が発生しない場合と、故障が発生する場合についてのシミュレーションを行い、HMSにおけるリアルタイムスケジューリングの柔軟性や頑健性の特長を示した。

生産システムの設計および運用における各アプリケーションは、共通のモデルリポジトリに基づいて実

行される。そのため、生産システムの設計フェーズにおいて MSE プロセス支援ツールで作成した生産システムモデルは、生産システムの運用フェーズにおける生産スケジュールのモデリングシステム、リスケジューリングシステム、および HMS のシミュレーションシステムで利用することができる。

以上のように、本論文では、オブジェクト指向技術を用いて開発した MSE プロセス支援ツール、生産スケジュールのモデリングシステム、リスケジューリングシステム、および HMS のシミュレーションシステムを用いて、生産システムの設計から運用までの過程における複数のアプリケーションを分散環境でコンカレントに実行する方法を提案し、簡単な例を用いて、コンカレントな分散処理の効果を示すことができた。

分散的な CIM や Virtual enterprise など、コンピュータを用いて製造企業内外の諸活動を統合化するためには、標準化機関等により、頑健な標準的モデルリポジトリが提案されるとともに、標準的モデルリポジトリに基づき製造企業個別の生産システムモデルを構築できる論理的かつ系統的な方法の確立が必要である。また、このようなモデルベースの製造企業の統合化を推進するためには、各アプリケーションのコンカレントな分散処理による時間の短縮など、構築された生産システムモデルの効果的な利用の方法を示すことも必要であると考え、本研究は、そのための基礎的な研究のひとつであると考え。

8.2 展望

本研究の成果を、実用的な生産システムの設計および運用支援ツールとして開発していく上での今後の課題と展望をまとめると、以下のようになる。

(1) 生産システムのモデル化における研究課題

- (a) モデルリポジトリの構築法の確立
- (b) モデルリポジトリの標準化

(2) 生産システムの設計フェーズにおける研究課題

- (a) MSE プロセスの記述機能の拡張
- (b) MSE プロセスにおける設計者の支援機能の拡張
- (c) MSE プロセスの事例の蓄積と再利用

(3) 生産システムの運用フェーズにおける研究課題

- (a) 生産スケジューリングおよびリスケジューリングの機能の拡張

- (b) 統合的生産管理支援システムへの拡張
 - (c) 実世界における HMS の構築および運用方法の検討
- (4) 製造企業のライフサイクル全般における研究課題
- (a) 生産システムライフサイクル全般の支援
 - (b) 製品ライフサイクル(Product lifecycle)との統合化
 - (c) サプライチェーンマネジメント(Supply chain management)との統合化

謝辞

大阪大学大学院 工学研究科 生産科学専攻の荒井栄司教授には、本論文の主査を引き受けていただき、大変お忙しい中、本論文をまとめるに際しての御指導、御助言をいただいた。また、機械システム工学専攻の三好隆志教授、先端科学技術共同研究センターの佐藤了平教授、生産科学専攻の白瀬敬一助教授には本論文の副査を引き受けていただき、本論文の問題点について詳細に御指摘、御指導をいただいた。ここに厚く御礼申し上げる次第である。

本論文は、大阪大学大学院 工学研究科 電子制御機械工学専攻 計算機援用生産工学講座に在学中の4年、さらに、大阪府立大学 工学部 機械システム工学科に助手として任官して以来6年余りの研究成果をまとめたものである。その間、先生方、諸先輩方をはじめとする多くの方々に御指導、御協力をいただいた。

本研究は、1991年当時、大阪大学大学院 工学研究科に在籍しておられた岩田一明教授(現 大阪大学名誉教授)の御指導のもと、知的 CIM のモデリングに関する研究としてスタートした。岩田教授には、大変お忙しい中、多大な御指導をいただいた。特に、研究活動において最も大切な事をお教えいただいた。心より感謝し、厚く御礼申し上げる。

1995年に大阪府立大学の助手として任官して以来、大阪府立大学大学院 工学研究科 機械系専攻の杉村延広教授には、本研究の遂行にあたり、多大な御指導、御支援をいただいた。特に、研究の進め方や論文のまとめ方について、懇切、丁寧にご指導をいただいた。また、本論文の副査を引き受けていただき、本論文をまとめるに際しての多大な御指導、御助言をいただいた。心より深甚なる謝意を表す。

大阪大学大学院の計算機援用生産工学講座(現 生産システム学領域研究室)に在籍中は、研究室の皆さんとの多くの議論を通じ、研究の素地を作らせていただいた。特に、小野里雅彦助教授には、研究における論理的なものの考え方について非常に多くのことをお教えいただいた。また、高田祥三助教授(現 早稲田大学 理工学部 教授)、平井慎一助手(現 立命館大学 理工学部 助教授)、寺本孝司助手、産業技術短期大学 機械工学科 小池稔講師をはじめ、研究室の皆さんに様々なアドバイスをいただいた。ここに深く感謝の意を表す。

大阪府立大学に任官してからは、学内外の多くの方々と議論をさせていただく機会を得た。大阪府立大学大学院 工学研究科の杉本正勝助教授には、生産分野以外の観点から多くの御助言をいただいただけでなく、時に励ましの御言葉をいただいた。また、IMS 国際共同研究プロジェクト MISSIONの研究活動において、生産システムの設計プロセスを記述、分析する際には、(株)デンソーの小島史夫氏、光行恵司氏、杉戸克彦氏、小野敏英氏、(株)牧野フライス製作所の砂金総一郎氏、マキノ J(株)の伴政明氏、清水建設(株)の由良佳之氏をはじめ、プロジェクトメンバーの方々に、多大なる御協力、御支援をいただいた。ここに厚く御礼申し上げる。

なお、本研究の遂行には、大阪府立大学大学院 機械生産工学研究室の学生諸君に多くの御協力

をいただいた。特に、永田智一君(現 日立製作所), 岩村幸治君(現 東京都立航空工業高等専門学校 助手), 竹内賢治君(現 ユニ・チャーム), 阪口龍彦君, 古賀樹文君(現 富士通), 井上和顕君(現 三機工業)には, 各プロトタイプシステムの開発において建設的な御協力をいただいた。ここに感謝の意を表す。

最後に, 研究が行き詰まり思い悩んだ時, それを忘れさせてくれる家族の笑顔にいつも救われた。この場を借りて, 妻と子供達に心から感謝の意を表したい。どうもありがとう。

2001年9月 谷水義隆

本論文の基礎となる発表論文

[1] 査読付き論文

No.	論文題目	著者名	発表雑誌	本論文との対応
1	生産システムの多目的モデリングに関する研究 (第1報, 複数のモデル間の整合性管理の支援)	谷水・杉村・岩田・小野里	日本機械学会論文集, 63-615, C (1997), 4065-4070.	第2章
2	生産システムの設計プロセス支援に関する研究 (第1報 設計プロセスの記述と支援ツールの開発)	谷水・永田・杉村・光行・由良・徳永	日本機械学会論文集, [投稿中].	第3章及び第4章
3	Consistency management among various models for production planning	Tanimizu, Y., Kurosawa, M., Sugimura, N.	<i>Internatinal journal of the Japan society for precision engineering</i> , 32-3 (1998), 219-220.	第5章
4	A study on object-oriented model repository for production planning of manufacturing systems	Tanimizu, Y., Kurosawa, M., Sugimura, N.	<i>Proc. of 1998 Japan-USA symposium on flexible automation</i> , vol.1 (1998), 365-372.	第5章
5	A study on object-oriented model repository for manufacturing system modeling	Tanimizu, Y., Sugimura, N., Iwata, K.	<i>Globalization of manufacturing in the digital communications era of the 21st century</i> , Kluwer academic publishers (1998), 337-350.	第2章及び第5章
6	生産システムの多目的モデリングに関する研究(第2報, 複数の生産計画モデル間の整合性管理)	谷水・杉村・岩田	日本機械学会論文集, 65-632, C (1999), 1725-1730.	第5章
7	A study on rescheduling of hierarchical and decentralized manufacturing systems	Tanimizu, Y., Iwamura, K., Sugimura, N.	<i>Proc. of 2000 Japan-USA Symposium on Flexible Automation</i> , CD-ROM13028 (2000), 1-8.	第6章
8	生産システムの多目的モデリングに関する研究(第3報, 階層分散型生産システムのコンカレント・リスケジューリング)	谷水・岩村・杉村・岩田	日本機械学会論文集, 67-654, C (2001), 283-288.	第6章

9	A study on object oriented modeling of holonic manufacturing system	Sugimura, N., Tanimizu, Y., Yoshioka, T.	<i>Manufacturing systems</i> , 27-3 (1999), 253-258.	第7章
---	---	--	--	-----

[2] 国際会議発表論文 (Extended Abstract 査読付き)

No.	論文題目	著者名	発表雑誌	本論文との対応
1	A study on model architecture of computer integrated manufacturing system	Tanimizu, Y., Sugimura, N., Iwata, K., Onosato, M.	<i>Proc. of APMS'96, IFIP (1996)</i> , 343-348.	第2章
2	A study on description method for manufacturing system engineering process	Tanimizu, Y., Nagata, T., Mitsuyuki, K., Sugito, K., Ono, T., Yura, Y., Tokunaga, N., Sugimura, N.	<i>Proc. of AMSMA'2000, Guangdong people's publishing house (2000)</i> , 749-752.	第3章
3	Description and analysis of manufacturing system engineering process	Tanimizu, Y., Sugimura, N., Yura, Y., Mitsuyuki, K., Sugito, K., Ono, T., Tokunaga, N.	<i>Proc. of 2000 Japan-USA symposium on flexible automation, CD-ROM13096 (2000)</i> , 1-4.	第4章
4	Object-oriented simulation model for holonic manufacturing systems	Sugimura, N., Tanimizu, Y.	<i>Proc. of the 14th international conference on production research (1997)</i> , 766-769.	第7章
5	Object-oriented simulation of real-time scheduling in holonic manufacturing systems	Sugimura, N., Tanimizu, Y., Ae, S.	<i>Proc. of APMS'99 (1999)</i> , 466-473.	第7章