

Department of Social Systems and Management
Discussion Paper Series

No. 1184

要求スループットを実現するためのビジネスプロセス構成：原理と
SCM 情報システムへの応用

by

佐藤 亮 , 河合 亜矢子

September 2007

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

題目： 要求スループットを実現するためのビジネスプロセス構成：原理と SCM 情報システムへの応用

概要： 本論文ではビジネスプロセスの動特性設計として、プロセスのスループットとサイクル在庫の関係を明らかにする。ビジネスプロセスはアクティビティ・インタラクション・ダイアグラムを使って離散事象システムとしてモデル化される。そのモデルに基づいて、決定論的状況におけるビジネスプロセスのサイクルタイムと在庫とスループットをリトルの公式によって計算する。分析の特徴は、クリティカル・サーキットとトークンの概念を用いることである。この計算原理によって適正在庫の意味が定まりその設計が可能になる。また、IT 導入などの改善によってボトルネック部分が変化した場合にも、適正在庫を再設計できる。本理論の応用として、サプライチェーンにおいて、セットメーカーがサプライチェーン全体のスループットとなるようにするために、前工程にあたるサプライヤー各社のビジネスプロセスを構成する方法を示す。サプライヤの情報システムも含めて平衡的動作を設計することによって構成される。

キーワード： トークン業務取引システム、アクティビティ・インタラクション・ダイアグラム、決定論的リトルの公式、サイクル在庫、ボトルネック・サーキット

Title: Organizing a business process that realizes required throughput: the principle and an application to information systems for SCM

Abstract: This paper clarifies logical relation between the throughput and cycle stocks of a business process that is a prominent part of dynamic property of a business process. A business process is modeled as a hierarchical discrete-event system depicted by Activity Interaction Diagram. By applying the Little's formula, the cycle time, inventory, and throughput of a business process in deterministic situation is calculated. Main characteristics of this analysis is to use the concepts of critical circuit and token in a business process. This theory allows us to define proper inventory for required dynamics and then to design it. If some improvement in efficiency had been attained by utilizing IT and then if new bottleneck process had been formed, proper inventory can be re-designed. Furthermore, a function of information systems in a supply chain is explained, as an application of this principle. It is shown how the process and information systems of suppliers should be organized properly, by considering stationary behavior of the whole process of the supply chain.

Keywords: token business transaction process, activity interaction diagram, deterministic little's formula, cycle stock, bottleneck circuit

1 はじめに

ビジネスプロセスがスムーズに動いて行くためには、プロセス内の適切なポイントに適切な在庫を持つことの必要性はよくいわれるところである。「すべての部品の在庫を一律に 5 日分にせよ。」とか「在庫をゼロにせよ。」というような在庫削減・原価削減の号令は、プロセス内で在庫切れと手待ちが頻出することとなって実行不能であるといわれる。

伝統的在庫管理論ではランダム変動とみなすことで在庫の時間変化をモデル化し、納入リードタイム期間中の使用量を分布でとらえることによって安全在庫とサイクル在庫の和として統計的に扱ってきた[13]。生産やそれに関連するプロセスの構造やサプライチェーンのネットワーク構造や情報システムがランダム性のふるまいの中に埋没してモデル化されるきらいがあった。プロセス内に保持される在庫とビジネスプロセスの時間的特性の関連について工学的原理が不明であった。

ビジネスプロセスのふるまいの不確実さの原因をこうしたランダム性に求めるアプローチと異なり、現場で設定されている標準作業時間を用いてプロセスの時間的なふるまいをある程度正確に計画し管理しようとするのが MRP (materials requirements planning)などの計画情報システムを用いるアプローチである[7, 16, 17]。部品表や作業手順をつかってビジネスプロセスの構造をとらえ、必要な数量を必要な時期に提供する制御方法を指向した。さらに能力所要量計画も加えて MRP II または ERP (enterprise resource planning package)、および APS (advanced planning and scheduling system)となっている[12]。こうした計画システムを用いる場合でも、原料から製品に至るプロセスの中で長い調達時間が必要なものは見込みによって資材や部品の形で用意しておかざるを得ない。そうしないと顧客にとって妥当なリードタイムや、生産者が目標とする生産速度(スループット)を確保できないからである[15]。制約理論[1]やそれに基づく生産計画システムの説明[14]では、「ボトルネックに注目して円滑に動かす」という気分は理解できるが、ボトルネックとなる作業を見つける方法や、改善する方法、スループット実現のために必要な在庫量の決定方法を分析し設計や改善する原理が不明確であった。

ビジネスプロセスは作業やワークセンターや部署や工場や企業といった大小

の離散事象システムの結合として構成されている[1,10,11]との立場にたつて、本論文ではビジネスプロセスの動特性設計として、スループットとプロセスの中のサイクル在庫の関係を明らかにし、その原理を使って適正在庫の設定方法を示す。つまり、決定論的状况における Little の公式 [3, 5]を用いて、ビジネスプロセスのサイクルタイムと在庫とスループットを計算する方法を示す。これによって、適正在庫の意味が定まりその設計が可能になる。また、作業改善や行程改善によってボトルネック部分が変化した場合にも、再度適正在庫を設計できるようになる。さらに、本理論の応用として、サプライチェーン・マネジメントにおける情報システムの機能を明らかにする。平衡的な周期的動作に注目することでビジネスプロセスの過渡的適応力に対しても示唆を与える。

2 ビジネスプロセス

2.1 アクティビティ・インタラクション・ダイアグラム

ビジネスプロセスの多くの機能とプロセス内で大量に使われる部材やデータなどのオブジェクトが多様に結合してできあがるビジネスプロセスは、全体としてひとつの離散事象システムとなる。離散事象システムは DEVS というモデルによって表現できる[8,9,10]。ビジネスプロセスは多くの離散事象システムが相互に関連し、全体として機能する。したがって、ビジネスプロセスを合成する立場からいえば、目標とする全体特性を持つようにするには、部品となる離散事象システムを用意し、ビジネス環境下で目標を達成するようにどのように組み合わせればよいか問題となる。

離散事象システムはいくつかの種類モデルが提案されている[2]が、本論文では DEVS 方法論(Discrete event system specification :単に DEVS と呼ばれる)を用いている[18,19]。一般に動的な時間変化する対象を記述するには、微分方程式や差分方程式のように因果性を正確に表現できなければならない。文献[6]によれば動的変化のモデル記述が因果的であることは、状態遷移関数が存在することと等価である。文献[9]に示されているように DEVS から状態遷移関数を構成することが常に可能であり、さらに、DEVS は離散事象システムのモデル化において普遍性を持っている。つまり、離散事象システムの特徴を持つシステムは必ず DEVS によって時間変化を表現できるのである。

ビジネスプロセスを業務活動とオブジェクトの結合によって離散事象システムとして構成する際には、プロセスの構成と、その構成による時間的ふるまいである動的性質の分析方法を提示する必要がある。プロセスを図的に構成する方法は activity interaction diagram (AID) として定義できる。

定義 1 アクティビティ・インタラクション・ダイアグラム (AID) [8,10]

活動図形と待ち行列図形とそれらをつなぐ矢印から成る図 (グラフ) であって、どのように矢印をたどっても、活動と待ち行列が交互にあらわれる構成となっているものをアクティビティ・インタラクション・ダイアグラムという。

AID の例は以下で多く示される。活動はそれを実行するための実行資源を持つ。たとえば生産の場合は、機械や組立ラインや作業者である。

AID は、システムの中にオブジェクト (またはエンティティ) の待ち行列が存在するような、オブジェクト間の変換や相互作用を離散事象システムとしてモデル化するひとつの方法である。すなわち、AID はオブジェクト / エンティティの各クラスが作る待ち行列と活動の関係 (相互作用) をグラフで表示する。

2.2 トークン業務取引システム

ビジネスプロセスではひとつの業務が離散事象システムとしてモデル化できるが、さらに一連の業務を行う部署であるとか、サプライヤやセットメーカーなどの企業もひとつの離散事象システムとなる。こうした大小の離散事象システムを含むビジネス実行プロセスのモデル作成や分析の議論を可能にするために、離散事象システムが結合して構成される結合離散事象システムの構成方法が必要であり、これをトークン業務取引システムとして定める。この際、構成方法の図的表現として AID を使うことができる。

定義 2 トークン業務取引システム

業務取引システムであって次の 3 つの条件を満たすとき、トークン業務取引システムとよぶ。

- (1) 待ち行列変数への入力と出力の矢印はそれぞれ一つだけである。

(2) トークン業務取引システムにおいては、各待ち行列変数が保持するオブジェクトをトークンと呼ぶ。トークンが移動する単位は1個である。つまり、活動開始によって入力からとられるトークンは一度に1個ずつであり、また、活動終了の場合もトークンは一度に1個ずつ出力される。ただし、同時刻に複数回の活動開始や終了が起こり、結果としてその時刻において複数個のトークンが移動することは起こる。

(3) 入力も出力もどちらも持たないような孤立した活動や待ち行列変数はない。つまり、AID は全体として必ず他とつなぐ矢印がある（連結性）。

トークンとはしるしとか偽コインという意味であり、トークン業務取引システムにおいては部品やデータに対応する。一般の業務取引システム[8]の概念ファイルを、単なる待ち行列にしたものがトークン業務取引システムである。トークン業務取引システムの静的表現はAIDである。後にトークン業務取引システムの周期性を分析する。

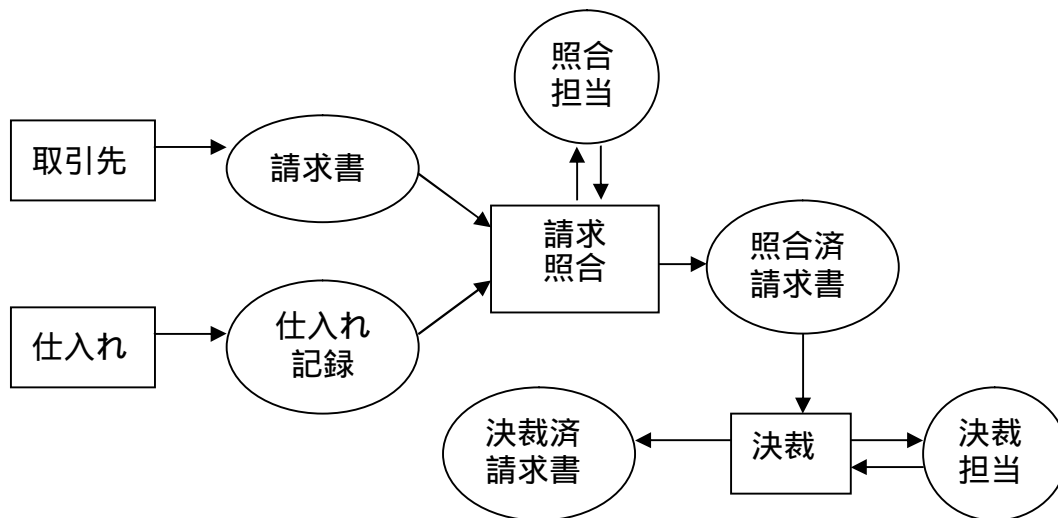


図 1. 買掛金決済プロセス

図 1 の買掛金決済プロセスの AID を考える。この例では待ち行列変数の集合 = {請求書, 仕入れ記録, 照合担当, 照合済請求書, 決裁担当者, 決裁済請求書} である。活動には 2 つの種類がある。入力となる待ち行列変数を持つ活動は内部活動と呼ぶ。内部トランザクションと呼ぶこともある。入力を持たない活動は

外部活動、もしくは外部トランザクションと呼ぶ。AID では外部活動の実行資源はしばしば省略される。図 1 の外部活動である「取引先」の出力は「請求書」であり、内部活動の「請求書照合」の出力は「照合担当、照合済請求書」の 2 つである。

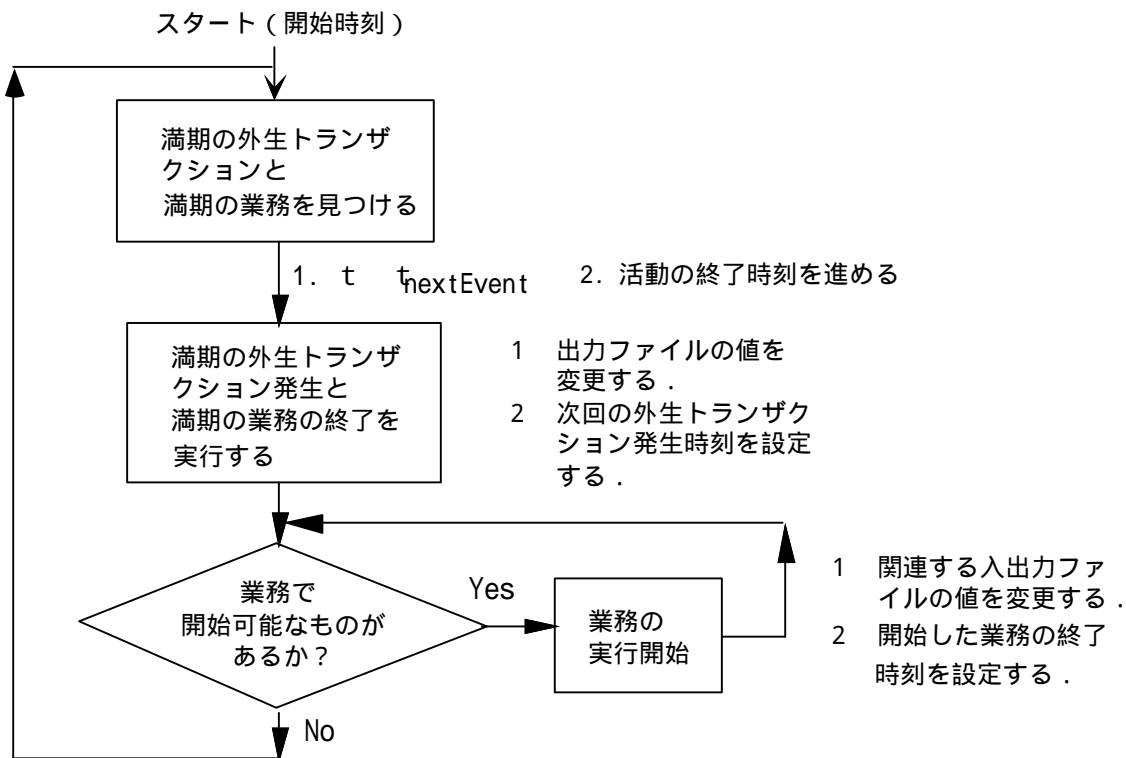


図 2 トークン業務取引システムの状態遷移を示すフローチャート

次にトークン業務取引システムの動的構造を定義する。システムの内部状態が時間とともに変化して行く状態遷移の仕組みが動的構造であり、図 2 のフローチャートで表現される。集合論的定式化は文献[8]を参照されたい。トークン業務取引システムは外部入力を外部活動としてモデル化しており、全体としては外部入力を持たない離散事象システムとなる。

図 1 の AID で表現された買掛金決済プロセスの状態遷移表は表 1 となる。各活動は DEVS であり、満期時間は表の t_a の欄にあるとおり、取引先による請求書発行、仕入れ、請求書照合、決裁について、それぞれ 15, 7, 10, 29 である。図 1 の結合変数を、たとえば請求書を請求書 Q (Q は待ち行列(queue)の略)として表記している。表 1 で、たとえば、取引先の欄の“(1,120)”とは現在 1 個の請求事項の処理中であり、それを開始したのが時点 120 であることを示す。照合担当者

は2名いる。照合担当1の欄の"(1,105)"は現在請求書と仕入れ記録の照合事項1個の処理中で、開始時点が105であることを示す。また、 "---" とは、照合活動がその時点で何も行っていないことを示す。

時刻	取引先	請求書Q	仕入係	仕入記録Q	照合担当1	照合担当2	照合済請求Q	決裁担当	決裁済請求Q
120	(1,120)	0	(1,115)	100	(1,105)	(1,120)	5	(1,116)	4
123	(1,120)	0	(1,115)	100	---	(1,120)	6	(1,116)	4
127	(1,120)	0	(1,127)	101	---	(1,120)	6	(1,116)	4
135	(1,135)	0	(1,127)	100	(1,135)	(1,120)	6	(1,116)	4
138	(1,135)	0	(1,127)	100	(1,135)	---	7	(1,116)	4
139	(1,135)	0	(1,139)	101	(1,135)	---	7	(1,116)	4
145	(1,135)	0	(1,139)	101	(1,135)	---	6	(1,145)	5

表1. 買掛金決裁プロセスの状態遷移表の一部

取引先	請求書Q	仕入係	仕入記録Q	照合担当1	照合担当2	照合済請求Q	決裁担当	決裁済請求Q
(1,120)	0	(1,115)	100	(1,105)	(1,120)	5	(1,116)	4

図3. 買掛金決裁プロセスの状態

買掛金決裁のビジネスプロセスのトークン業務取引システムは、ある時点で図3のような具体的な状態になっている。この情報は、一般的には、図4のように表現される。ここで、各々の業務活動の状態は、たとえば取引先を表す離散事象システムについては、取引先が $g_{取}$ という状態を開始した時刻が $t_{取}$ であることを表現している。

取引先	請求書Q	仕入係	仕入記録Q	照合担当1	照合担当2	照合済請求Q	決裁担当	決裁済請求Q
$(g_{取}, t_{取})$	2	$(g_{仕入}, t_{仕入})$	105	$(s_{照1}, t_{照合1})$	---	7	$(s_{決済}, t_{決済})$	89

図4. 買掛金決裁プロセスの状態

買掛金決済における請求書などのデータは、顧客の連絡先や口座などの他のデータと連動している。また、仕入れ先に支払う金額は一括して手形によって支払われたり、あるいは逆に一件の仕入れに対して何回かに分割して支払ったりもする。こうしたデータを保持する仕組みは単純な待ち行列ではないので、一般の業務取引システム[8]として帳票や EDI（電子データ交換）などのビジネス要求のデータを格納する情報システムを使って維持管理して行く必要がある。トークン業務取引システムの状態遷移の数学的定義は一般の業務取引システムと同じだが、上記のように単に請求書を 1 件ずつカウントして待ち行列ととらえることで単純化している。単純化されているので、状態遷移の仕組みの鍵となるポイントは、図 4 のような状態情報を更新して状態遷移表ができ上がって行く仕組みである。

買掛金決済プロセスは図 4 の状態情報を使って任意の時点で、

$$(t_{\text{取}} + ta_A(g_{\text{取}})), \quad (t_{\text{仕入}} + ta_B(g_{\text{仕入}})), \quad (t_{\text{照合}} + ta_C(g_{\text{照合}})), \quad (t_{\text{決済}} + ta_D(g_{\text{決済}}))$$

の 4 つからどれかが早く満期になる。これら 4 つの値の最小値を $t_k + ta_k(s_k)$ とし、さらに、 $t' = \max\{t_{\text{取}}, t_{\text{仕入}}, t_{\text{照合}}, t_{\text{決済}}\}$ とするとき、

$$ta(s) = t_k + ta_k(s_k) - t_k$$

によって、 s の次にイベントが起こるまでの時間を得る。つまり、ある状態から次の状態へと遷移するのは、活動の中で最も早く満期になる時点である。

以上のようにして、AID の構造によって業務活動や組織が待ち行列を介して結合しているビジネスプロセスのネットワークが、全体として機能する仕組みがモデル化される。

3. ビジネスプロセスの平衡状態と周期性

3.1 トークン業務取引システムの周期性

ビジネスプロセスの動的な特性を考察する際にリトルの公式を適用するためには、システムが不動点に到達していたり、周期をもって繰り返し動作をしているなどの平衡状態にあることが十分条件である。つまり周期的動作をしていればリトルの公式を適用できる。

さらに、活性という性質を考える。

定義 3 活性

トークン業務取引システムがある状態 s について活性であるとは、初期状態を s としたときの状態遷移表が永久に続くことである。また、ある状態について活性なとき、トークン業務取引システムが活性であるという。

活性であっても永久に開始しない活動が含まれていることがある。その場合はそれを取り除いたトークン業務取引システムを考える。取り除く場合にその活動の入出力アローも同時に取り除く。

トークン業務取引システムは静的構造としての AID を持つが、AID のつながり方に注目する。すべての内部活動を集めた集合を A とし、外部活動の集合を E と表す。また、結合変数の集合を Q とする。AID におけるパス(path)とは矢印の向きに沿った A または E または Q の要素の列である。パスにおいて初めの活動と最後の活動が同じときサーキットと呼ぶ。この定義によれば孤立した活動や孤立した待ち行列変数もサーキットとなるが、本論文では孤立活動や孤立待ち行列変数を含まないもののみを考える。また、元の AID の部分集合を部分 AID とか部分業務取引システムと呼ぶ。

定義 4 AID の強連結性

トークン業務取引システムの静的構造(AID)をとる。任意の活動記号 $a \in A$ と待ち行列変数 $q \in Q$ に対して、矢印の向きにしたがって到達できるような a から q へのパスと q から a へのパスがそれぞれあるとき、その静的構造は強連結という。また、強連結な静的構造を持つトークン業務取引システムを強連結であるという。

強連結なトークン業務取引システムでは外部活動を含まない。

次に、強連結な部分 AID のつながり合った AID を考える。

定義 5 強連結な部分 AID

任意の静的構造(AID)をとる。

AID の部分グラフ(部分構造) G が元の AID のひとつの極大強連結部分であ

るとは、次の3つの条件を満たすことである。

(1) G の任意の待ち行列変数と活動は、互いに矢印を順方向にたどって、つながっていること。

(2) 任意の結合変数に対して、ある活動があって、結合変数から活動へのパスと、活動から結合変数へのパスとを適宜用いて構成できる場合には、それらのパスがすべて部分グラフ G に含まれること、および、それらのパスに現れた活動と待ち行列変数も G に含まれること。

(3) 任意の活動 a に対して、ある待ち行列変数 $q \in Q$ があって、 q から a へのパスと a から q へのパスとが構成できる場合には、それらのパスがすべて G に含まれること。

上の定義の意図は簡単であり、図5に示すようにお互いにパスがある活動と待ち行列変数を、パスを構成する矢印ごと取り出したものである。定義により、もしある活動が(または待ち行列変数が) G に含まれないということは、活動と G 中の待ち行列変数との間には、パスが存在しないか、一方向のパスだけが存在する。

一般にひとつの AID において強連結部分は複数あり、複数の強連結部分は一意に定まる。

図5で例を述べると「顧客・注文1・組立1・作業員1・組立1・部品1」はひとつのパスである。「顧客・注文1・組立1・部品1」は別のパスである。「作業員1・組立1・作業員1」はサーキットである。また、3つの強連結部分を持ち、「作業員1・組立1・作業員1」、「作業員2・組立2・作業員2」、「作業員F・組立F・作業員F」はそれぞれ極大強連結部分である。

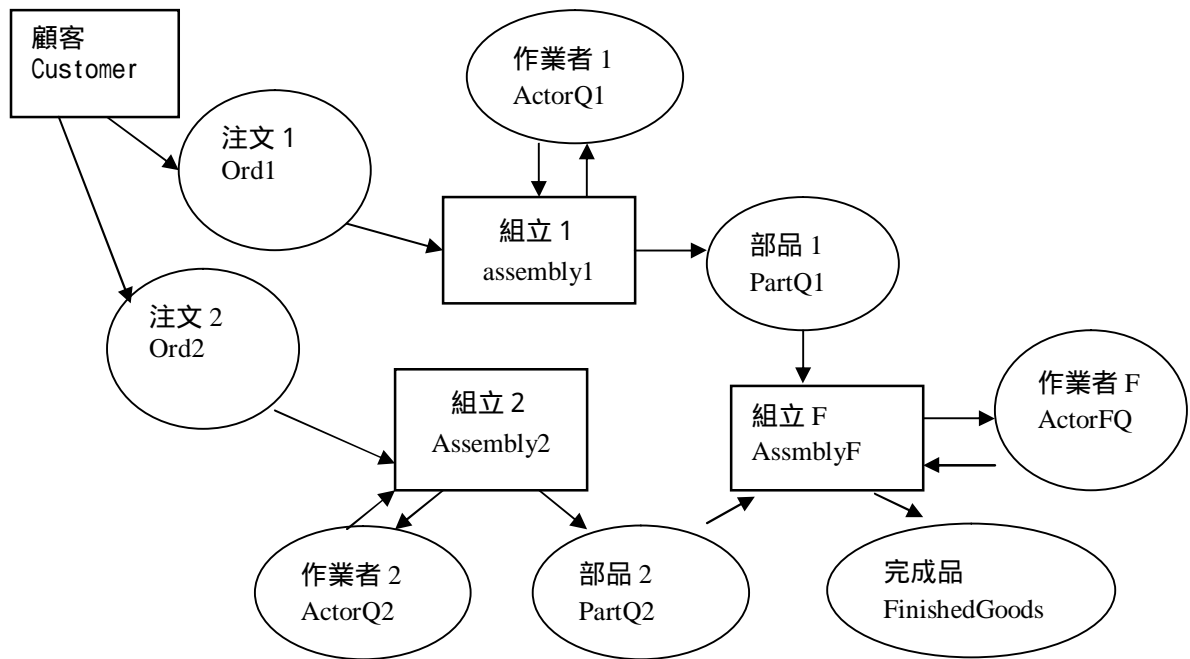


図 5 シンプル業務取引システムのサーキットと強連結部分

命題 1

シンプル業務取引システムの静的構造は、有限個の極大強連結部分にユニークに分かれる。

命題 2

シンプル業務取引システムのどの強連結部分にも含まれない待ち行列変数が保持するトークン数が、任意の時間において一定の個数以上にならないとする。このとき、シンプル業務取引システムが時間変化する過程で持ちうるトークンの総数は上限を持つ。

上の命題のような「シンプル業務取引システムのどの強連結部分にも含まれない待ち行列変数が保持するトークン数が、任意の時間において一定の個数以上にならない」とき、その業務取引システムが有界結合であるという。以下に示すとおり、有界結合性は周期の存在を意味する。

命題 3

活性で有界結合なシンプル業務取引システムの状態表において、活動の残り時間の値は、有限通りの場合しかない。

命題 4

活性で有界結合なシンプル業務取引システムは周期を持つ。

3.2 周期的振る舞いとサイクルタイム計算

前節で述べた命題の例を示す。図 5 に示された 2 つの部品を組み合わせて製品にするためのシンプル業務取引システムの周期的な動きを示す。図において各活動の加工時間（所要時間）は次のように設定されている。各作業者の数は 1 人とする。

組立 1： 3 分 / 個

組立 2： 57 分 / 個

組立 F： 33 分 / 個

顧客からの注文の間隔の大小によって、シンプル業務取引システムが有界結合になることもあればならないこともある。有界結合であれば周期的状態において成立するリトルの公式によってプロセス全体のサイクルタイムが計算される。

(1) 顧客からの注文間隔が 65 のときの平衡的動作

初期値として、顧客からの注文のうちで初期に未処理の注文 1 の個数が 2、注文 2 の初期未処理個数が 5、部品 1 と部品 2 の初期在庫がそれぞれ 2 個と 0 個、最終製品の初期在庫が 1 個とする。図 5 のプロセスでは全体が強連結ではない。この初期値から状態遷移表によって動的変化を調べる。すると、有界結合であって周期状態となっている。時刻 2,405 から 2,470 の 1 周期の様子を示す。

time	Cust	Ord1	Ord2	A1Q	Asm1	Pt1Q	A2Q	asm2	Pt2Q	AcFQ	AsmF	FingsQ
2405	(1,65)	1	1	1	-asm1-	0	1	-asm2-	1	1	-asmF-	41
2405	(1,65)	0	1	0	(1,3)	0	1	-asm2-	1	1	-asmF-	41
2405	(1,65)	0	0	0	(1,3)	0	0	(1,57)	1	1	-asmF-	41
2408	(1,62)	0	0	1	-asm1-	1	0	(1,54)	1	1	-asmF-	41
2408	(1,62)	0	0	1	-asm1-	0	0	(1,54)	0	0	(1,33)	41
2441	(1,29)	0	0	1	-asm1-	0	0	(1,21)	0	1	-asmF-	42
2462	(1,8)	0	0	1	-asm1-	0	1	-asm2-	1	1	-asmF-	42
2470	(1,65)	1	1	1	-asm1-	0	1	-asm2-	1	1	-asmF-	42

表 2 注文間隔 65 分のときの 1 周期の状態遷移

プロセス全体を律しているのは、AID の構造から直感的には流れが合流している組立 F である。その組立 F の起動を決定しているのは、組立 1 の活動である。その様子は、上の状態遷移表で矢印とアミかけで示した部分である。

リードタイム計算 1 : 組立 1 + 組立 F のパスについて。

TH は周期から決定される。TH = 1/65 [個 / 分]

このパスの WIP を状態遷移表から組立 1 + 組立 F のパスについて見て取れる。

$$\text{Asm1}(3) + \text{AsmF}(33) = 36$$

または、2441 - 2405 = 36 分である。

そのパスに沿った 1 周期間の平均在庫は、(Ord1, Asm1) と (Pt1Q, AsmF) について 1 周期分を観察することにより、WIF = (3+33)/65 である。

よってサイクルタイムは CT = WIF/TH = 36 [分]

この例では、状態遷移表が得られている時にリードタイムを調べるには、AID の各活動の可能な連鎖において、開始から終了までがいずれもブロックされない (待たされない) ような連鎖を見つけてそれらの活動の満期時間の和をとればよい。定常的な待ち行列 (WIP) がないので、満期時間の和がすなわちリードタイムとなる。

(2) 顧客からの注文間隔が 57 分のときの平衡的動作

注文間隔が 8 分違うだけで、業務取引システムの構造と共鳴するようなことが起こり、様子が大きく変わる。

time	Cust	Ord1	Ord2	A1Q	Asm1	Pt1Q	A2Q	asm2	Pt2Q	AcFQ	AsmF	FingsQ
26334	(1,57)	1	5	1	-asm1-	4	1	-asm2-	1	1	-asmF-	462
26334	(1,57)	0	5	0	(1,3)	4	1	-asm2-	1	1	-asmF-	462
26334	(1,57)	0	4	0	(1,3)	4	0	(1,57)	1	1	-asmF-	462
26334	(1,57)	0	4	0	(1,3)	3	0	(1,57)	0	0	(1,33)	462
26337	(1,54)	0	4	1	-asm1-	4	0	(1,54)	0	0	(1,30)	462
26367	(1,24)	0	4	1	-asm1-	4	0	(1,24)	0	1	-asmF-	463
26391	(1,0)	0	4	1	-asm1-	4	1	-asm2-	1	1	-asmF-	463
26391	(1,57)	1	5	1	-asm1-	4	1	-asm2-	1	1	-asmF-	463
26391	(1,57)	0	5	0	(1,3)	4	1	-asm2-	1	1	-asmF-	463
26391	(1,57)	0	4	0	(1,3)	4	0	(1,57)	1	1	-asmF-	463
26391	(1,57)	0	4	0	(1,3)	3	0	(1,57)	0	0	(1,33)	463
26394	(1,54)	0	4	1	-asm1-	4	0	(1,54)	0	0	(1,30)	463
26424	(1,24)	0	4	1	-asm1-	4	0	(1,24)	0	1	-asmF-	464
26448	(1,0)	0	4	1	-asm1-	4	1	-asm2-	1	1	-asmF-	464
26448	(1,57)	1	5	1	-asm1-	4	1	-asm2-	1	1	-asmF-	464

表3 注文間隔 57 分のときの 2 周期間の状態遷移

1 周期が 57 分の周期状態において（平衡状態において）矢印とアミかけで示した部分がプロセス全体を律している。つまり、オブジェクトを処理して行く活動の連鎖があって、その連鎖においてはどの活動も開始をブロックされて待たされるということがない。そのようなパスは組立 F と組立 2 である。

リードタイム計算：組立 2 + 組立 F のパスについて。

TH は周期から決定される。TH = 1/57 [個 / 分]

平均在庫の計算：状態遷移表から組立 2 + 組立 F のパスについて見て取る。

Asm2(57)+AsmF(33) = 90[個分] (1 周期間)

および OrdQ2 の在庫の 4 個が 1 周期に渡って存在するから、4*57[個分]。

合計を 1 周期の長さで平均すると

$$W2F = (90+4*57) / 57$$

よってリトルの公式で計算されるサイクルタイムは CT = W2F/TH = 90+4*57 = 318[分]

実際に状態遷移表からトレースして、新たに入った注文が完成するまでの時間は、ord2 の待ち行列で先にいる 4 個がなくなるまでの時間（4*57 分）と、当該注文が Asm2 と AsmF で加工される時間（57+33=90 分）の和である。

4*57+90=318 分である。

4 サイクル在庫によるボトルネックの連続運転原理

4.1 WIP の定義

WIP (work-in-process:仕掛け在庫) は在庫量の時間平均値である[5]。入庫が一定期間ごとに行われる一方で、一定のスピードで出荷がなされて在庫が減っていく場合について計算方法を確認しておく。実際は品物は離散的に減っていくのであるが、連続量によって離散的变化を近似すると理想的には次の図のようなノコギリ状の図形とみなせる。時刻 t における在庫量を $w(t)$ と表すと、

$$\begin{aligned} WIP &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T w(t) dt \\ &= \frac{1}{N} \int_0^N \left(-\frac{Q}{N}t + Q\right) dt = \frac{Q}{N} \left[-\frac{t^2}{2N} + t\right]_0^N = \frac{Q}{2} \end{aligned}$$

である。ノコギリ図形の中のひとつの三角形の面積からも計算される。図 6 のような入出荷の在庫はサイクル在庫と呼ばれる[3,13]。なお、伝統的在庫管理論ではサイクル在庫に安全在庫を加えて確率的変動に備える。

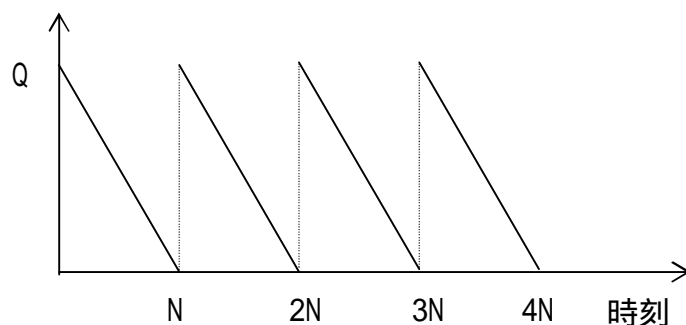


図 6 サイクル在庫

4.2 ビジネスプロセスに対する決定論的リトルの公式

強連結で活性なトークン業務取引システムについての時間的な特性の計算方法を考える。

命題 5

強連結で活性なトークン業務取引システムでは、任意のサーキットの中のト

トークン数は活動の開始と終了によらずに一定数を保つ。

命題 6

強連結で活性なトークン業務取引システムは、かならず周期的なふるまいをする。また、すべての活動は1周期の間に、同じ回数の開始(と終了)を行う。

定義 6 シンプル周期動作

1周期の間に、すべての活動がただ1回の開始と終了をする強連結で活性なトークン業務取引システムは、シンプル周期動作をするという。

定義 7 クリティカルサーキット

最大のサイクル平均をもつサーキットをクリティカルサーキットという。ここで、サイクル平均とは業務取引システムの各サーキットの1トークンあたりの活動時間のことであり、以下で定義される。

サーキット C_i 上の活動時間の和を $\sum_{a_j \in C_i} a_j$ 、トークン総数を k_i で表す。このとき

$$\text{サーキット } C_i \text{ のサイクル平均} = \frac{\sum_{a_j \in C_i} a_j}{k_i}$$

定義 8 クリティカル活動サーキット

活動とその実行資源からなるサーキットがクリティカルになっているトークン業務取引システムを、クリティカル活動サーキットを持つという。

一般に、「たくさんの」オブジェクトを与えると、プロセスはクリティカル活動サーキットを持つようにできる。

図7のような強連結な AID を考える。活動 a,b,c の3つがひとつのサーキット C を形づくる一方で、各々の活動が活動実行資源を n_a, n_b, n_c 個ずつ持っている。

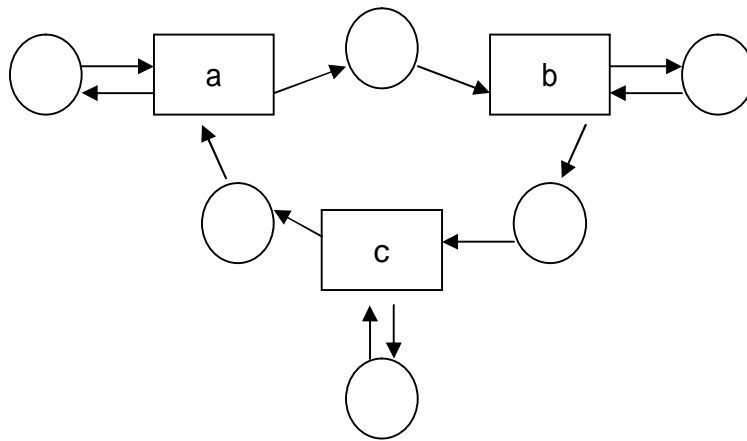


図7 簡単な強連結 AID

各活動の保持時間をやはり a, b, c と表すことにする。サーキット C 上のトークン数を n とする。4つの基本サーキットのサイクル平均値の大小によって、クリティカル活動サーキットになるかどうかが決まる。C が最大サイクル平均値を持つためには、つまり、C がクリティカルサーキットであるためには、次の3つの不等式が成立する必要がある。

$$\frac{(a+b+c)}{n} > \frac{a}{n_a}, \frac{(a+b+c)}{n} > \frac{b}{n_b}, \frac{(a+b+c)}{n} > \frac{c}{n_c}$$

もしサーキット C が生産工程であって、生産速度を上げて運転しているときは、 n が適当に大きな値となっているので $n \gg 1$ が成り立つ。たとえば $n=5$ とする。一方で、生産プロセス a が3つの工程のうちで最も長い生産時間を必要する高価な機械であるため $n_a = 1$ 台である場合は、

$$\frac{(a+b+c)}{5} < \frac{3a}{5} < \frac{a}{1}$$

となって、C はクリティカルにサーキットにならず、機械 a がクリティカルになる。このような意味において、クリティカル活動サーキットになっている場合は少なくない。

命題7

強連結で活性なトークン業務取引システムをとる。クリティカル活動サーキットを持ち、シンプル周期動作をすると仮定する。周期を L とする。このとき、ク

リティカルな活動を含む 任意のサーキット C について、 C 内のトークン数を w とすると、このサーキットの任意の活動から出発したトークンが再びその活動に戻ってくるまでのサイクルタイム CT は、

$$CT = wL$$

である。つまり、リトルの公式が成立する。

系 8

強連結で活性なトークン業務取引システムがクリティカル活動サーキットを持ち、シンプル周期動作をすると仮定する。周期 L とする。このとき、任意のサーキット C について、 C 内のトークン数を w とすると、サイクルタイム CT は、

$$CT = wL$$

である。

命題 9

強連結で活性なトークン業務取引システムがクリティカル活動サーキットを持ち、シンプル周期動作をすると仮定する。周期 L とする。このとき、クリティカル活動のサイクル平均値を \bar{h}_C とすると (ネットワークの最大サイクル平均値に他ならないが)、

$$\bar{h}_C = L$$

である。

一般にビジネスプロセスの動特性を評価する尺度として、リトルの公式に表現される 3 つの特性であるスループット (TH)、サイクルタイム (CT)、仕掛り在庫 (WIP) と、それらに加えて稼働率がある [3]。シンプル周期動作の強連結で活性なトークン業務取引システムでは、稼働率が次のように計算される。ただし、 $A(C)$ はサーキット C に含まれる活動の集合を表し、 $|A(C)|, |A|$ はそれぞれ集合 $A(C), A$ の要素数を表す。

$$\text{ひとつの活動 } i \in A \text{ の稼働率 } \frac{h_i}{L}$$

$$\text{サーキット } C \text{ の平均稼働率} \quad \frac{\left(\sum_{i \in A(C)} \frac{h_i}{L} \right)}{|A(C)|}$$

$$\text{プロセス全体の平均稼働率} \quad \frac{\left(\sum_{i \in A} \frac{h_i}{L} \right)}{|A|}$$

このように、シンプル周期動作のトークン業務取引システムの稼働率に関係するのは上の式から周期 L だけである。周期は TH の逆数である。周期動作をしているとき、稼働率には CT も WIP も関係しない。

4.3 SCM の計画情報システムの計画リリース機能

サプライチェーン全体の動特性を設計するために、生産計画管理情報システム (production planning and control system: PPC system) の機能を、物質やオブジェクトを変換していく流れであるロジスティクスプロセスへのフィードバックの流れを追加して閉経路を形成することによる平衡運転達成装置として位置付ける。

まず、図 8 のような単純なプロセスとサーキットを考える。

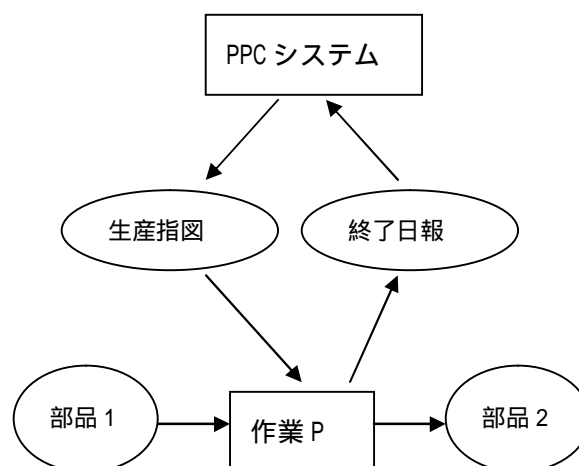


図 8 作業と生産計画管理情報システム

作業 P は部品 1 を加工して部品 2 を製造する。PPC（生産計画管理システム）は、MRP のように販売計画値を在庫の情報から一定期間後に生産指図を発行する。部品 1 の結合変数には十分な量の部品があって作業 P が平衡的運転状態にあるとき、以下のように指図の発行サイクルを調整することにより、WIP と CT を変更することができる。

平衡運転状態では作業 P への部品 1 の入力速度と部品 2 の出力速度は同一である。これはスループットと呼ばれる。TH[個/日]で表す。PPC-生産指図-作業 P-終了日報-PPC で構成されるサーキットを考える。一定個数ロットの製造を繰り返すものとして、1 ロットを 1 つのトークンで抽象化して考えると、このサーキットではトークンが回ることになる。トークンは生産指図であったり、作業中の部品であったり、終了報告であったりする。このサーキットのサイクルタイムとは、トークンが一周して戻るまでの時間であった。周期的に動作している場合、サーキット上のどこから出発したとしてもサイクルタイム[日]は一定である。1 周期の長さが L[日]、サーキット上のトークン総数が w[個]とし、1 周期の間には作業 P がただ 1 回だけ開始と終了を行うと仮定するとき、作業 P の加工速度が TH[個/日]だから次が自明に成り立つ。

$$CT = wL = w / (TH) [日]$$

命題 7 はこのことを一般的状況について明らかにしているのでより複雑な状況も分析できる。例として図 9 を考える。

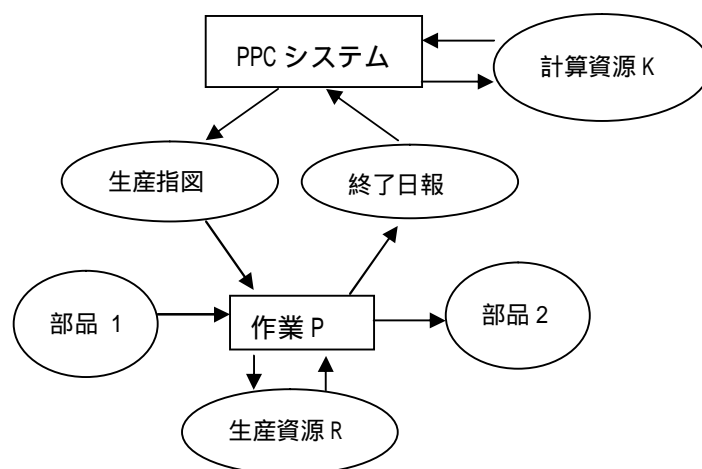


図 9 2 つのサーキット

図9においては、図8のサーキット以外に2つのサーキットを含んでいる。作業Pとその作業を実行する生産資源R（機械や作業員）が作るサーキットと、PPCとKが作るサーキットである。全体はクリティカル活動サーキットを持ち、周期Lでシンプル周期動作をすると仮定する。図9のサーキットをC1、PPCとKのサーキットをC2、PとRのサーキットをC3とし、それぞれのサーキット上のトークン数を w_1, w_2, w_3 とするとき、サイクルタイムは各々

$$CT_i = w_i L$$

となる。

図9は、部品1と2を除くと全体が強連結になっている。部品1が作業Pの開始に影響を及ぼさないように十分ある状況では、部品1と2を除いて動作を分析できる。その場合、C1, C2, C3それぞれのサイクル平均値[日/個]は

$$(h_{PPC} + h_p) / w_1, h_{PPC} / w_2, h_p / w_3$$

である。ただし、 h_p は活動Pにおいて1台の生産資源（機械）を使って部品1を1個加工して部品2にするまでの加工時間である。活動Pを実行する生産資源が複数ある時には、並行的に実行できる場合を考える。たとえば、もし5台の機械がある場合には、1台だけの場合の5倍速く加工をすることができる。生産指示がたくさん発行されるとき、最大（最速）で $h_p / 5$ [日]ごとに部品2を1個生産できる。 h_{PPC} は生産日報を受け取ってから計算の後に次の生産指示を出すまでの計画期間であり、たとえば $h_{PPC} = 1$ 週間である。PPCは計算資源Kを無数に持っているものとし、活動Pに生産資源が複数ある場合と同様に、生産日報を受け取る度に、 h_{PPC} 後には対応する生産指図を発行する。

与えられた TH [個/日]を達成するには、

$$\frac{(h_{PPC} + h_p)}{w_1} \leq \frac{1}{TH}, \frac{h_{PPC}}{w_2} \leq \frac{1}{TH}, \frac{h_p}{w_3} \leq \frac{1}{TH}$$

が成立していなければならない。そうでないと、処理速度が部品1の入力速度（=部品2の出力速度）に追い付かずに、部品1の個数が時間が経つに従って無限に増えて行く。言い換えれば、図9のプロセスの処理速度を上げるには、必要なだけのWIPを置けばよい。このことを利用して次のような在庫配置の調整によるプロセス全体のスループット制御を行える。

図10のようなシンプルな直列のサプライチェーンにそれぞれPPCをもつプロセスを考える。図で四角形は企業活動を表す離散事象システムであり、たとえば「組織2(17)」とあるのは、組織2では1個を生産するのに17日を要することを表す。また「PPC3(2)」はPPC3という計画情報システムは作業終了のトークンを受け取ると2日後に生産指図を出すことを表す。各組織は活動実行資源を表すトークンをそれぞれ何個か保持しているが、図では省略されている。サプライヤの実行資源は1つとする。個々の四角で表された企業や活動が離散事象システムで、図のように結合変数で結合されているとき、サプライチェーン全体はひとつの離散事象システムとなる[8]。

組織3がセットメカで最終製品を作っているとき、セットメカからの注文を受けて組織2が製造と納品し、組織2の注文を受けて組織1が製造と納品を行う。サプライヤは組織1に原材料を納品する。

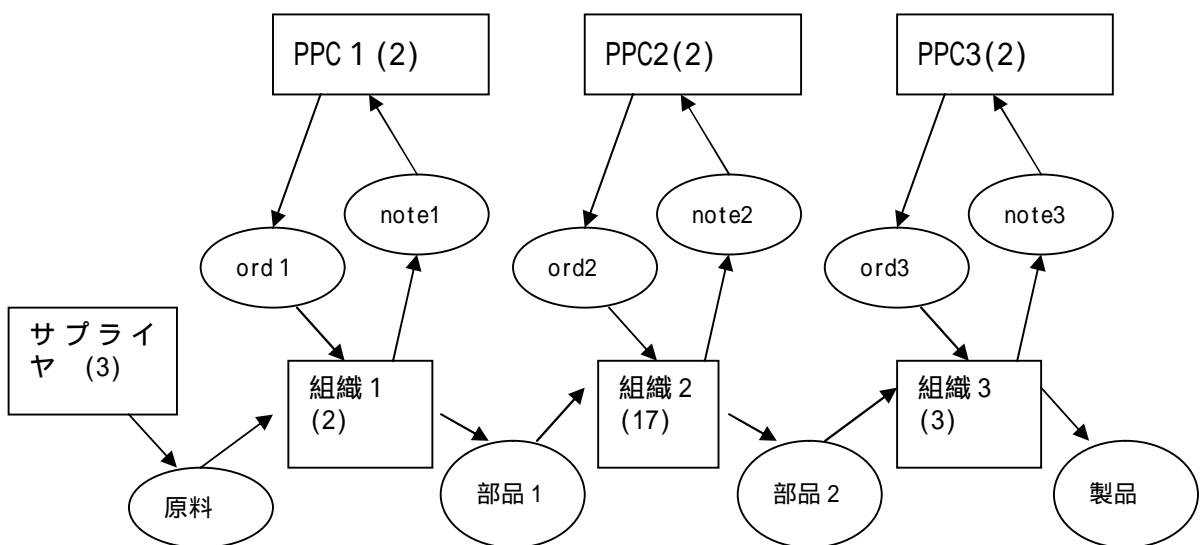


図10 4つの組織の直列サプライチェーン

チェーン全体は確定的に動作しているとき、中間部品の仕掛り在庫は必須だろうか、あるいは確定的だから不要だろうか。これは次の命題として述べることができる。

命題 10

図 10 のサプライチェーンプロセスが企業 3 をクリティカル活動とするクリティカル活動サーキットを持ち、シンプル周期動作をすると仮定する。周期 L とする。図 10 において、組織 1 と PPC1 でつくるサーキットを $C1$ 、組織 2 と PPC2 とでつくるサーキットを $C2$ とする。企業に十分な生産資源があるとき、組織 3 をスタブさせないような、つまり、組織 3 を 100%稼働させるような $C1$ 、 $C2$ それぞれの最小の仕掛かり在庫が存在し、

$$w_i = CT_i \cdot TH \quad (i=1,2)$$

によって定まる。ただし、 TH は組織 3 のスループットであり、 $TH = 1/L$ である。

上の命題から次のようなことが分かる。第 1 に、セットメーカの作業が滞らないようにするには、つまり、材料不足や生産指示不足が起こらないようにするために、基本的に、いくつかのトークン（材料であったり情報であったりする）が一定量存在する必要がある。つまり、在庫ゼロによるセットメーカの手待ちゼロは原理的に不可能である。これは、決定論的な場合にでも、原理として不可能であることが分かる。第 2 に、トークンは一定量が必要であるが、それ以上は「セットメーカの手待ちゼロ」運転のためには無駄である。第 3 に、最小量として必要なトークン量は、在庫日数で表現すると、サイクルタイムの分が必要である。つまり、定常に運転できているときには、わが社のサイクル在庫日数がわが社のサイクルタイムと一致する。安全在庫は、それに追加して設定することになる（図 11）。第 4 に、計画サイクルの設定がセットメーカのサイクルタイムより長すぎると、原理として、絶対にどう頑張っても定常運転を達成できない。

図 10 に当てはめると、 $C1$ には、PPC1 と企業 1 の間に最少で 2 つのトークンが必要であり（実現としては、初期生産指図を 2 通流すること）、 $C2$ には、生産資源が最少で 6 つ（ $17/6 < 3$ より）と PPC2 と企業 2 の間の生産指図として 7 つが必要である（ $(17+2)/7 < 3$ ）。また、状態遷移表は表 4 のようになる。周期が 3、クリティカル活動サーキットを持ち、シンプル周期動作であるのが分かる。なお、表 4 においては組織 2 は 6 個の生産資源を持っていて、状態遷移表で Act21 から Act26 までで表現されている。それら 6 個は無差別である。また、表で "1(5)"

のように記されているのは、1個のトークンを処理中であって、終了までの時間が5単位時間であることを示す。このような表記法でも状態遷移が表現できることは命題3の証明中で示してある。

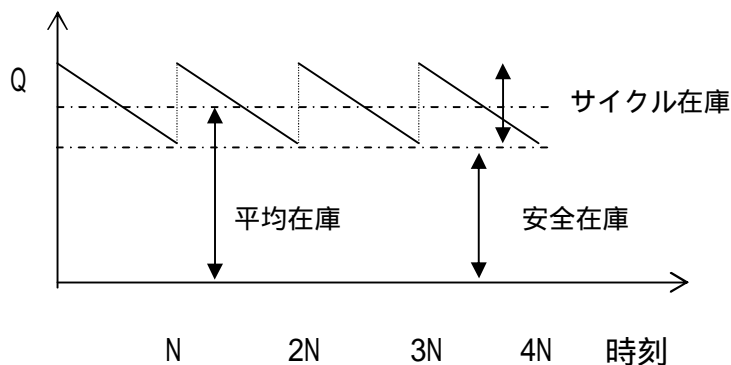


図 12 平均在庫レベル

time	Sply	mat	PPC1	ord1	Act1	note1	prt1	PPC2	ord2	note2	Act21	Act22	Act23	Act24	Act25	Act26	Prt2	PPC3	ord3	Aact3	note3	prt3
35	1(1)	0	1(2)	1	----	0	0	1(1)	0	0	1(17)	1(2)	1(5)	1(8)	1(11)	1(14)	0	1(2)	0	1(3)	0	8
36	1(3)	0	1(1)	0	1(2)	0	0	----	1	0	1(16)	1(1)	1(4)	1(7)	1(10)	1(13)	0	1(1)	0	1(2)	0	8
37	1(2)	0	----	1	1(1)	0	0	1(2)	1	0	1(15)	----	1(3)	1(6)	1(9)	1(12)	1	----	1	1(1)	0	8
38	1(1)	0	1(2)	1	----	0	0	1(1)	0	0	1(14)	1(17)	1(2)	1(5)	1(8)	1(11)	0	1(2)	0	1(3)	0	9

表 4 状態遷移表 (組織 2 が 6 つの生産資源を持つ場合で、周期は 3)

5 結論

本論文ではビジネスプロセス全体に要求されるスループット（稼働速度）を実現するようなビジネスプロセス内の適正在庫配置原理を考察した。さらにこれを適用して、サプライチェーンで典型的なケースである、セットメーカーからの納入要求を満たすためのサイクル在庫の意味と設定方法を明確にした。

セットメーカーはサプライチェーン全体のスループット（産出速度）を定め、前工程にあたるサプライヤー各社はそのスループットを達成するようなビジネスプロセスを構成しなければならない。その際、チェーンの部分についても全体についても、動的特性はプロセス内に存在するサーキットについてのリトル

の公式として表現される。リトルの公式で表現される関係は、物質が到着し加工を受け出荷されるという流れにおいて、物質が消えたり真空中からあらわれたりしない限り、つまり、物資が保存される限りは必ず成立し、何ものもそれから逃れられない。そのため、目標スループットを達成するにはセットメーカーがクリティカルサーキットになるようにチェーン内のすべてのサーキットのサイクル平均値がセットメーカーのそれよりも小さな値となる必要がある。そのためには、サーキット内に存在するトークンが多いことが必要である。ここでトークンとは、部品や材料などの物質以外に、労働者や機械やセル生産の屋台といった生産資源も、伝票や計画や日報などのデータも含まれる。業務遂行に使われるものがビジネスプロセスにとってのトークンである。トークンはオブジェクトと言い換えることも可能である。それらのトークンが存在する WIP としてリトルの公式に表現される。結果として、従来のリードタイム期間の安全在庫概念に代わる概念枠組みが提示された（命題 10）。また、命題 7 と系 8 に表現されているとおり、ビジネスプロセスのスループットを設計する際に、ボトルネックとして認識すべき対象は単独の業務ではなくボトルネック・サーキットである。特定の業務だけを単独で問題視してやみくもに人員や作業ラインを増やしても効果は保証されないので、業務や組織のつながりでできたビジネスのネットワークを見渡して達成したい（目標とする）スループットを持つボトルネック・サーキットを定め、それ以外のサーキットの WIP を最適に設定することができるのである。このようにネットワークのダイナミックな特性を設計するという事は、在庫変動をランダム変動とみなして対応しようとする伝統的な在庫管理論ではまったく説明されない方法である。

計画情報システムがサーキットに含まれる場合、本文中でトークンとして扱ったように、計画作成や更新による出荷指図や製造指図のリリースのタイミングも考慮に入れる必要がある。この意味において、計画情報システムを含めたビジネスプロセスの構造と運転方式を設定する必要があるということである。

本論文で考察したビジネスプロセスはシンプル周期動作し、かつ、クリティカル活動サーキットを持つという限定したものである。本文中で例示したように重要な例でその条件を見たすものも多いのであるが、将来のビジネスプロセス工学の理論的な発展としては、より一般的なビジネスプロセスの分析のため

の原理が開発されなければならない。また、需要は一定の増加傾向を示したり、あるいは非定常なランダム性や間欠性を持っていたりする。増加する需要に対して計画情報システムによって適応的に生産を増やすアプローチをとる場合に、最適的な目標在庫配置をすることによってサプライチェーン全体の平均在庫を減少させながら需要増加に追従できることが観察されている[4]。そうしたプロセスに対しても、平衡状態を前提とし在庫の適正配置を行うという本論文の設計法で理論的に説明できるのか、また、どこまで有効に使うかを明らかにする課題が残されている。

参考文献

- [1] ゴールドラット(1992), 三本木訳、ザ・ゴール – 企業の究極の目的とは何か, ダイヤモンド社, 2001.
- [2] Ho, Y.-C. (1989), Special issue on dynamics of discrete event systems. Proceedings of the IEEE, 77-1.
- [3] Hopp,W.J. and Spearman, M.L., Factory physics : foundations of manufacturing management, Irwin/McGraw-Hill, 2001
- [4] 川西亜矢子,佐藤亮,住田潮: 計画情報共有システムを用いるサプライチェーン管理の構造的特徴について, Department of Social Systems and Management Discussion Paper Series No.1135, University of Tsukuba, 2005.
- [5] Little, J.D.C., "A proof for the queuing formula: $L = \lambda W$," *Operations Research*, 9, 383-387, 1961
- [6] Mesarovic, M. D. and Takahara, Y. , *Mathematical foundation of general systems theory*. Academic, 1975.
- [7] 中根甚一郎 : 総合化 MRP システム , 日刊工業新聞社 , 1984.
- [8] Sato, R. and Praehofer H. "A discrete event model of business system - A Systems Theoretic for Information Systems Analysis: Part 1." *IEEE Trans. Systems, Man, and Cybernetics*, Volume 27, 1997, pp.1-10.
- [9] Sato, R.: Realization theory of discrete-event systems and its application to the uniqueness and universality of DEVS formalism, *Int. J. of General Systems*, 30-5, 2001, pp. 513-549.
- [10] 佐藤亮: ビジネスプロセスの離散事象モデリングと業務取引ペトリネットによる特性分析, システム/制御/情報、45-8、pp470-479、システム制御情報学会、2001.
- [11] 佐藤亮 , 「仕事はイベントで動いていく」, 飯島・佐藤編 『システム知の探究 2』 , 日科技連出版社 , 1997.
- [12] Scheer, A.W, *Business process engineering-Reference models for industrial enterprises*, 2nd ed, Springer, 1994
- [13] Silver, Pyke, Peterson: *Inventory Management and Production Planning and Scheduling*, 3rd edition, John Wiley & Sons, 1998.

- [14] Stein, R.E.: Re-Engineering the manufacturing system – applying the theory of constraints, Marcel Dekker, 1996.
- [15] 田中一成：生産物流統合管理システム D/SNS, 日刊工業, 1988.
- [16] 鳥羽登：SE のための MRP、日刊工業新聞社、1995.
- [17] Vollmann, TE, Berry, WL, Whybark, DC, Jacobs, F.R. (2005), Manufacturing Planning and Control Systems for Supply Chain Management, 5th edition, McGraw-Hill.
- [18] Zeigler, B. P. *Theory of modelling and simulation*. John Wiley, 1976.
- [19] Zeigler, B. P. *Multifaceted modeling and discrete event simulation*. Academic, 1984.