

会計の基礎理論と情報構造からみた会計システムのアーキテクチャ —— SAP と Oracle EBS の比較を通じて

The Architecture of Accounting System on Basic Accounting Theory and
Information Structure
—— via Comparison between SAP and Oracle EBS

藪原弘美, 緒方朱実

要約 企業の情報システムは、サービスやソリューションを選択し繋ぎ合わせ利用していく時代になってきた。このようなサービスやソリューションの選択に当たって、“機能”や“サービス”を重視し、アーキテクチャがあまり重視されないことが多い。しかし、アーキテクチャの違いはデータ連携、情報活用、保守において様々な形式で顕在化してくる。

企業の会計システムは、比較的システム化しやすい分野であった。しかし、国際会計基準を初めとした様々なニーズへの対応により年々複雑化している。利用されるソリューションも SAP 社の SAP ERP (以下 SAP) や Oracle 社の Oracle EBS など大規模なものも多くなってきており、これらのソリューションはそれぞれアーキテクチャが異なる。

たとえば、帳簿体系は、Oracle EBS は帳簿会計を、SAP は伝票会計を採用している。管理会計では、目的別に機能モジュールが豊富に存在する SAP に対し、Oracle EBS は財管一致を重視し、モジュールを持たず、財務会計の仕訳明細の項目で処理する。本文店会計においても、一つの取引を一つの伝票にすることを重視する SAP と、データ連携も考慮した柔軟な帳簿体系を重視する Oracle EBS とで、データ構造やアプリケーションの構成は大きく異なるのである。

加えて、リアルタイムという観点から情報システムのアーキテクチャを考える。システム連携を重視し、処理の頻度を上げてタイムリー性を上げる Oracle EBS と、もともと全てのモジュール利用を前提としてリアルタイム会計を追求する SAP とでは、会計システムのアーキテクチャは全く異なる。さらには、販売や在庫/購買システムとの連携においても、全体アーキテクチャ決定に大きな影響を及ぼすのである。

このように、ソリューションの開発思想の違いはデータ構造やアプリケーション構成に大きな影響を与える。ソリューションの選定あるいは、これらのシステムとのデータ連携・活用に当たっては、全体の開発思想を深く理解することが必要である。

Abstract In the era that the enterprise information system is becoming linked together with selected services and solutions has come. On selecting these services and solutions, the “function” and “service” which can be realized are focused on, but the architecture not on in many cases. However the differences of architectures become visible in some cases like data interface, data utilization, and maintenance.

The field of cooperate accounting was relatively easy to systematize. But various needs such as IFRS convergence, the accounting system have been becoming complex from year to year. The used solutions, such as SAP ERP and Oracle EBS, are becoming large, and these solutions have different architecture.

For example, under the ledger system, Oracle EBS adopts ledger accounting system, while SAP adopts voucher accounting system. Under the management accounting, SAP has many subjective modules, while Oracle EBS focuses on the financial and management accounting consistency has no management accounting modules and operates with the attribute of journal entries in the management accounting. The inter-office accounting differs considerably in the data structure and application architecture, between SAP focusing on one slip per transaction, and Oracle EBS focusing on the flexible accounting books system considered the data coordination. In addition, we make a study on the architecture of information system on realtime standpoints. The accounting system architecture is completely different between Oracle EBS and SAP, the former emphasizes the system coordination and enhances the number of transactions to increase the timelines, the later pursues the realtime accounting assuming the use of all of subjective functional modules. These differences affect decision of application architecture as a whole not only on the accounting system architecture but also on system coordination with sales, stock and purchase systems.

Thus, the differences of development philosophy for solution have much effect on data structure and application configuration. It is important to understand development philosophy as a whole prior to selection of solution and/or data coordination and utilization with these systems.

1. はじめに

従来の会計システムは、伝票を入力し財務諸表を作成するというシンプルな構造であり、システム化しやすい分野であった。しかし1990年代後半に入ってから急激な環境変化、管理会計、会計基準の国際化など、多くのニーズへの対応が求められており、会計システムの構造は年々複雑化している。利用するソリューションもSAP社のSAP ERP（以下、SAPと表記）や、Oracle社のE-Business Suite（以下、Oracle EBSと表記）など、大規模かつ広範囲なものとなっている。

ERPパッケージは、そのアーキテクチャすなわちデータ構造やアプリケーション構成を変えて導入することは推奨されない。このため、表面に現れる“機能”や、利用方法に着目して導入可否の検討を行い、アーキテクチャ面からの分析は通常行われない。しかし、ユーザ企業の導入目的、業務方針、文化、システム環境の違いから、選択したソリューションのベースとなる考え方が馴染まないケースも散見される。よって、当該ソリューションのアーキテクチャとそのベースとなる考え方を十分に理解しなければ、データ連携や情報活用、システムの全体最適はもとより、業務の全体最適も実現できない。

本稿では、まず、伝票会計・帳簿会計の理論、本支店会計、管理会計をその構造面から考察するとともに、これらにおける考え方の違いがデータ構造やアプリケーション構成に及ぼす影響をSAPとOracle EBSの比較分析を通じて明らかにする。次に、期間損益計算と取引別損益計算の理論を構造面から考察し、データ構造やアプリケーション構成への影響を示す。

これらの検討を通じて、会計理論の違いがアーキテクチャに与える影響やアーキテクチャ検討の重要性を説くとともに、情報システム構築時に検討すべきポイントの具体例を提示する。

2. 会計の基礎理論と会計情報の構造

本章では、会計の基礎理論を財務会計と管理会計に分け、はじめに財務会計における帳簿や伝票の考え方や決算における方式の違いが会計システムの構造に及ぼす影響を述べる。次に管理会計すなわち企業内部の業績管理の考え方が会計システムの構造にどのような影響を及ぼす

かを考察する。

2.1 財務会計と情報の構造

財務会計で論点とされるテーマのうち会計システムの構造に大きな影響を与えるテーマである帳簿会計・伝票会計と支店会計を取り上げ、その論点を解説した上で SAP と Oracle EBS との違いを述べ、それぞれの違いに起因するアーキテクチャの差異を考察する。

2.1.1 帳簿会計・伝票会計と情報の構造

財務会計における主要情報の構造を論じる場合、まず論点とされるのが帳簿会計か伝票会計かという点である。

帳簿会計とは、すべての取引を都度仕訳帳に記入し、総勘定元帳に転記する帳簿ベースの会計処理をいう。帳簿会計の帳簿とは「仕訳帳」と「総勘定元帳」からなる主要簿と、補助簿の二つの帳簿組織から成り立っている (図 1)。

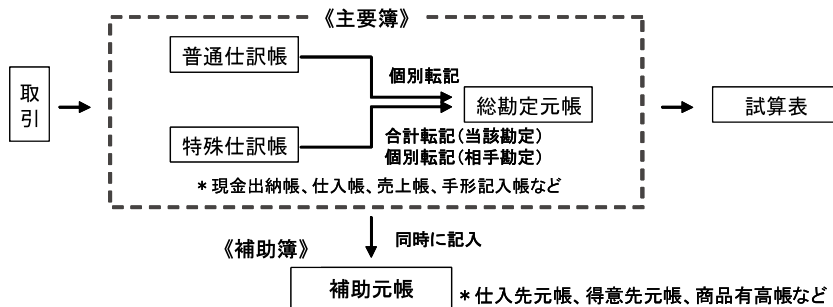


図 1 帳簿会計の体系

一方、伝票会計は、会計取引を伝票によって記録し、伝票を集計することで総勘定元帳や補助簿を作成する仕組みである。伝票会計により伝票を様々な現場で起票することが可能になり、日計表からの合計転記により記帳や集計を迅速に行える (図 2)。

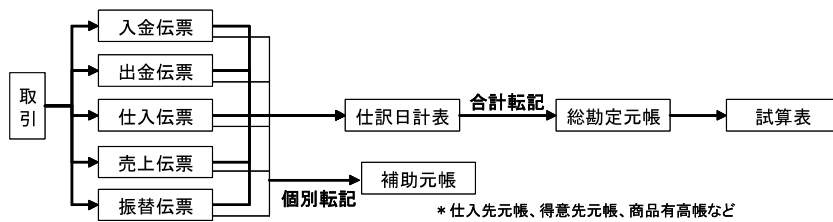


図 2 伝票会計の体系

Oracle EBS は帳簿会計をベースにしており、図 3 のような仕訳を中心にした転記処理を行う帳簿体系を採用している。仕訳データと残高データを会計帳簿と呼ぶ。

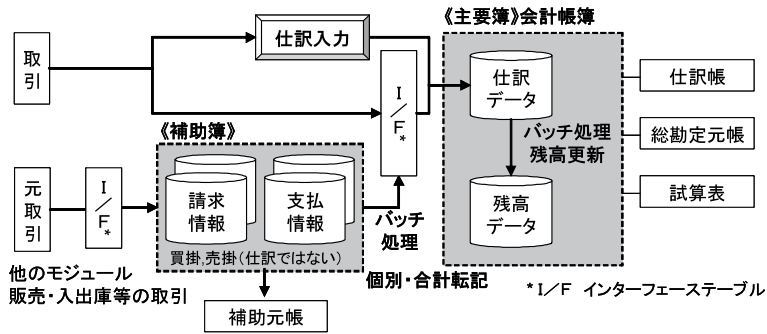


図3 Oracle EBSの会計帳簿

SAPは伝票会計を採用しており、図4のような仕組みになっている。SAPでは取引別に起票された伝票明細がすべての帳簿・処理の基礎データとなる（大福帳システム）。

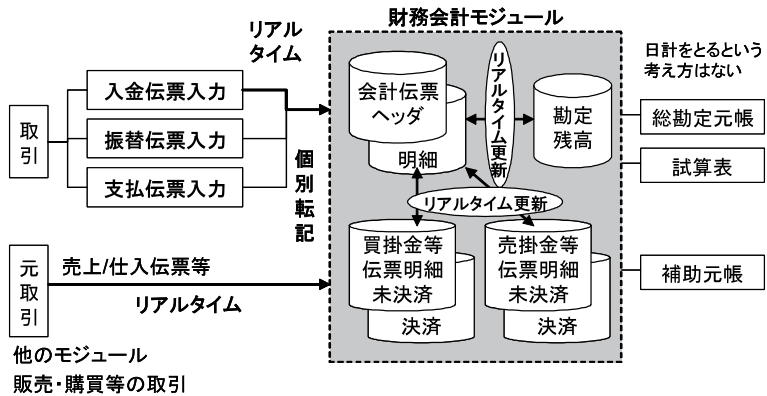


図4 SAPにおける会計伝票と帳簿

そもそもの伝票会計と帳簿会計の考え方は、手作業による記帳をベースとしていた。システム化が進んでいる現代では、いずれの方式であっても帳簿や伝票を情報システム上で共有することで、記帳の分業や集計の省力化は図れる。しかし、これら二つの考え方に起因するデータの持ち方と更新順序の違いは会計業務に大きな影響を与える。たとえば、伝票は取引単位に起票するものだが、仕訳は必ずしも取引単位である必要はなく、この違いは3.1節で述べる取引別損益計算と関連し、リアルタイムな業績管理に影響する。また、帳簿会計は転記前に承認や変更を行うことができ、帳簿の独立により補助簿からの転記を締めることで柔軟な締めを行うことも可能である。

伝票会計に立脚するか、帳簿会計に立脚するかがアーキテクチャに与える影響は大きい。当該ソリューションのテーブルや更新タイミングを把握し、いずれの方法を採択しているかの判断が必要である。また、仕訳が取引ベースなのか、サマリーか、そのボリュームの分析、取引別損益管理やリアルタイム管理の重要性と採択可否の確認、補助簿の単位と管理・データ連携方式の確認、他のシステムも含めた承認や締めの全体方針の策定などを行うことが必要である。

2.1.2 本支店会計と情報の構造

財務会計における主要情報の構造を規定しているものに本支店会計がある。本支店会計と

は、支店が独自の帳簿に記帳し、決算も支店独自で行う会計の仕組みである (図5)。支店を独立して評価する目的で古くから行われている。

本支店会計は以下の順番で行われる。

1. 支店内取引は当該支店の帳簿のみに記帳
2. 本支店間取引は振替先・振替元双方に伝票起票、双方で承認後それぞれの帳簿に記載
3. 決算時は、本店・支店勘定を照合し、現物や通知が未到達の取引の処理を実施
4. 全社合算後、支店の本店勘定と本店の支店勘定を相殺
5. 社内仕入・社内売上を照合・相殺

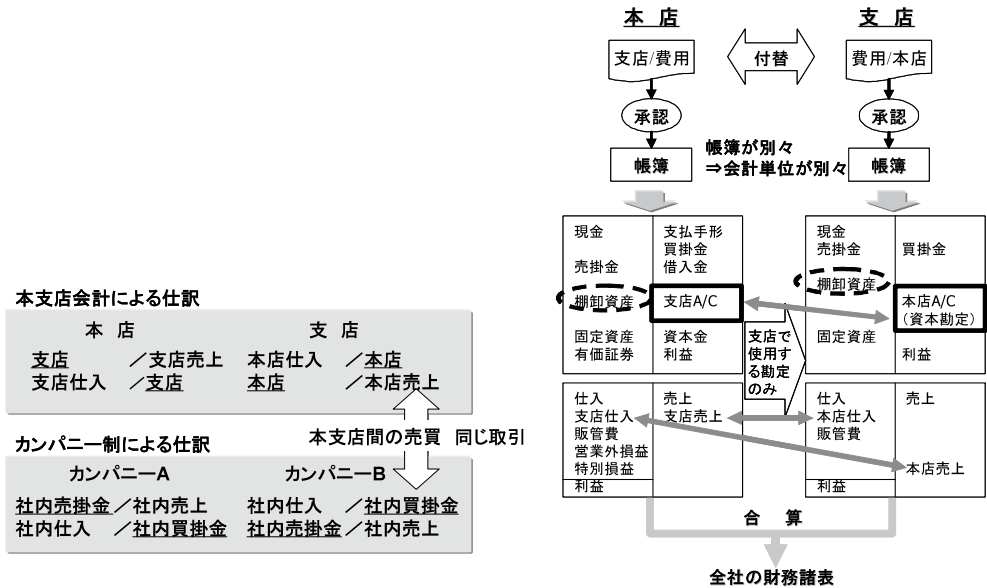


図5 本支店会計の構造

本支店会計は広く解釈すると、会社内部に会計単位を設けて損益計算書のみならず貸借対照表も用いて管理し評価する仕組みである。そこでの本支店勘定は、社内債権債務、社内資本、利益などの概念が混在しており、貸借対照表を利用した評価 (ROI, ROA など) や意思決定は厳密に言えはできない。

紙での帳簿の考え方ではない現代は、帳簿を分けなくても仕訳明細データに多彩な切り口を持たせることで迅速に集計できるようになった。このため、本支店勘定は会計単位間で貸借対照表のバランスを保つためだけのものとなり、本支店会計の役割は採算・評価目的より、勘定管理・決済・決算などの会計業務としての便宜に重心を移した。

SAPは、ワンファクト・ワンプレース*1を重視しており、本支店間取引を二つの取引に分けて伝票起票する考え方は馴染まない。双方での伝票の承認は、リアルタイムの考え方にも相容れない。このため、SAPの財務会計モジュールは主に会社単位のB/S, P/Lを出力するというシンプルな機能であり、次節で述べる管理会計モジュールによりP/Lをベースとした損益計算書を重視する。管理会計では組織としての切り口・事業としての切り口などの多様な観点からのフレキシブルな比較を重視する (図6)。

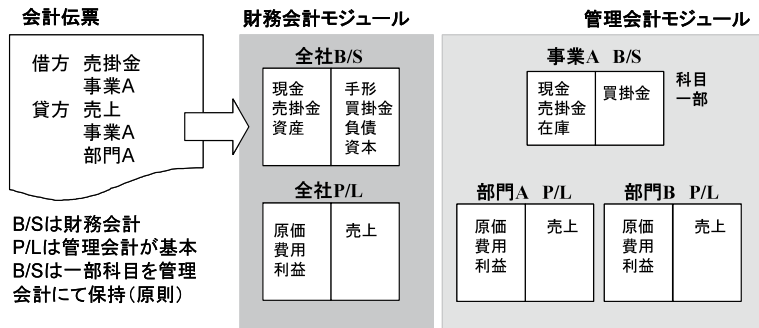
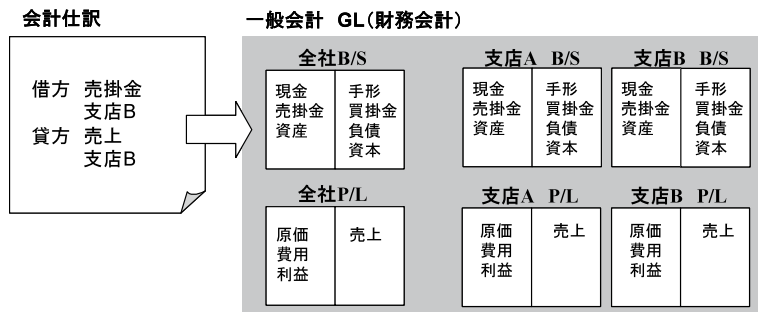


図6 SAPによる組織別情報の管理体系

一方 Oracle EBS においては、SAP より帳簿の運用が柔軟であり、一つの帳簿内に会計単位の情報を持ち、会計単位毎に貸借一致を保証する。管理会計の P/L において片仕訳*2 も許容する SAP に比して本支店会計のベースは整っていると思われる。ただし、いずれの場合においても、厳密に本支店会計を行う場合、振替先での承認や締めやの制御、勘定相殺などの機能の追加開発が必要である。



B/S、P/L双方とも貸借一致した状態で財務会計で保持可能

図7 Oracle EBS における組織別情報の管理体系

なお、近年の国際会計基準へのコンバージェンスの一環で、貸借対照表の重要度が増してきている。これにともない、財務会計の機能の中で、会社内部の組織で貸借対照表を作成する機能が重視されつつある。SAP においても、近年のバージョンでは、会社内部の組織単位での貸借対照表機能も拡充されており、差異は小さくなっている。

本支店会計は、財務会計と管理会計の機能分割や会計システムによるデータの持ち方に影響を与える。このため、ビジネスとしての貸借対照表の重要度、ワンファクト・ワンプレースやリアルタイム性の重要度を確した上で、支店毎のアプリケーションの配置、本支店取引のボリュームとパターン、支店内の会計と業務システムの関連性などを把握することが必須となる。

2.2 管理会計と情報の構造

会社内部の業績評価・業績管理いわゆる管理会計のデータをどのように管理するかという問題も会計システムの構造を論じる上で大きな課題である（本支店会計もその一部であるが、決算の便宜という観点からこちらは財務会計の部分で検討した）。

管理会計とは経営者のための企業内部の管理を目的とする会計である。管理会計には様々な手法・分野があり、近年では多彩な管理の仕組みが開発されてきた(表1)。

表1 多彩な管理会計の仕組み

<p>【伝統的管理会計】 原価計算, コストコントロール, 予算管理</p> <p>【新しい管理会計】 事業部・カンパニー制会計 バランスド・スコアカード(BSC) 活動基準原価計算(ABC/ABM) 意思決定会計(プロダクトミックス, 価格決定, 投資決定)</p>
--

SAPにおいては、管理会計機能モジュールはコスト管理用、利益管理用といった「利用目的別」に分かれ豊富に存在する。管理会計のデータは財務会計とは別に独立して存在し、管理会計独自のデータの投入や、シミュレーションが可能になり、簡便かつ柔軟に集計・分析ができるようになっている(図8)。

SAPは「利用目的別」にDBを分割することにより取り扱う項目・データ構造を個別に定義し、専用機能の装備を充実することで、それぞれの分野に特化した細かな業務サポートを行う。

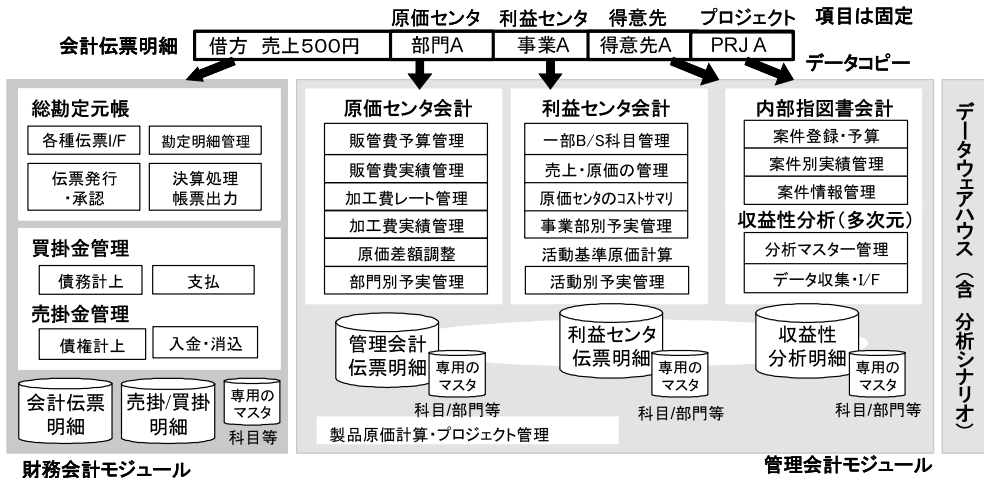


図8 SAPにおけるモジュール構成とデータ構造

Oracle EBSには、管理会計モジュールというものは存在しない。会計データは財管一致の前提があり管理会計のデータとして特に独立を要さず、管理会計における集計項目は、すべて財務会計の仕訳に管理項目として持つ。Oracle EBSの仕訳明細の項目はSAPのように項目が固定ではなく、会計フレックスフィールド(AFF: Accounting Flex Field)というユーザ定義可能な領域に自由に定義して利用する。この方法は非常に柔軟かつ簡易で判りやすく、Oracle EBSの最大の利点である(図9)。しかし、管理会計で利用する情報を、財務会計の仕訳明細のどの項目に格納するかも原則としてユーザ企業ごとに決定するため、まず「管理会計の利用目的(何を何の目的でどう管理するのか)」から定義しなければならない。

利用項目をユーザ選定に任せているため、目的を固定した各種の専用機能の提供は難しくなる。しかし、シンプルなデータ構造により、仕訳を通した情報の統合・格納および、活用支援に重点が置かれており、各種機能を提供している。これにより、SAPのような、データ連携時に仕訳を管理会計と財務会計に振り分ける複雑なロジックや目的別DBの管理、専用マスタの管理が不要になり、データ連携や活用が容易である。

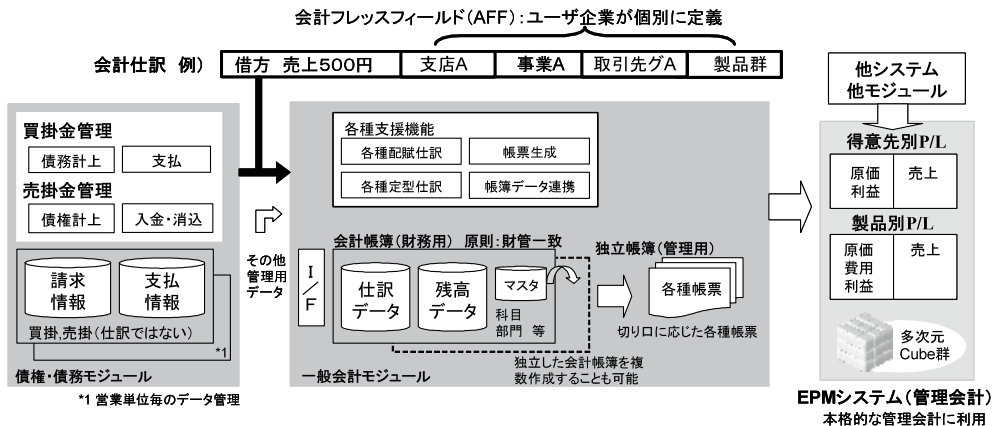


図9 Oracle EBSにおけるモジュール構成とデータ構造

管理会計システムの構築を行う場合、管理会計の「利用目的」、すなわち業務支援に重点を置く（利用目的別）か、データ活用・連携に主眼を置く（財管一致）か、さらに情報系システム（シミュレーション、ダッシュボード）をどのように活用するのかを十分に検討する必要がある。

3. リアルタイム会計と会計情報の更新

タイムリーな損益管理を重視する企業は多い。第3章ではリアルタイム会計と会計情報の更新として、まず期間損益計算という会計の前提と取引別損益計算の必要性を述べ、商品売買の勘定記入の方法である三分割法、各種在庫評価方法の特徴を説明した上で、リアルタイム損益管理という観点よりSAPとOracle EBSの比較を行い、リアルタイム性の重要度と情報システムのアーキテクチャに及ぼす影響を考察する。

3.1 期間損益計算・取引別損益計算

会計の仕組みは、清算することを前提にした場合と、今後も事業継続することを前提にした場合では、会計上の考え方も価値そのものも異なってくる。事業継続を前提とした現代の企業では事業を一定の期間に便宜的に区切って収支の算出を行うことが求められる。この仕組みのことを期間損益計算という。

期間損益計算の実現だけであれば会計システムは期次または月次のバッチ処理で足りる。しかし、環境変化の激しい昨今においては、日次での利益管理を行いたいというニーズも多い。また、業種によっては取引毎にその利益を把握したいというニーズもある。リアルタイム損益管理を行うには、取引ごとの損益把握がベースとなる。

これらを実現するために、企業は業務プロセスの見直し、I/Fタイミングやコンピュータの

処理速度の向上などの対策を施すことが多いが、それだけでは実現できない。それは制度会計そのものが、期次・月次ベースの期間損益計算を前提に作られているからである。リアルタイムで業績の把握を行うには、売上原価対立法による勘定記入、継続記録法による在庫管理、標準原価法あるいは移動平均原価法の採用が必須になる。

3.2 勘定記入の方法と会計情報の更新

商品売買を行った際の記帳方法に三分割法を利用している企業は多い。三分割法において商品売買は、繰越商品、仕入、売上の三つの勘定を利用して記帳される。三分割法によると繰越商品は前期末の在庫額が記載され、期末にまた当月末の期末在庫額が仕訳されるまで更新されない。商品を仕入れた場合は都度、仕入/買掛金という仕訳を起こす。期末に仕入/期首繰越商品、期末繰越商品/仕入という振替仕訳を行い、仕入勘定の最終貸借差額が売上原価の金額になる。簡便な方法ではあるがバッチペースでの運用となる(図10の左)。

一方、売上原価対立法は、商品の入庫の都度、商品/買掛金の仕訳を起こす。また、売上時に売掛金/売上という仕訳のほかに、売上原価/商品という仕訳を起こす(図10の右)。その際、在庫の払出単価(=売上原価額)を都度決定する必要がある。売上と原価がタイムリーに仕訳され、関連付けを行うことにより、取引別の損益が都度把握できる。ただし、この方法を採用するには、商品の入出庫を都度記帳・把握(継続記録法による在庫管理)し、商品払出単価の決定が都度必要である(3.3節参照)。

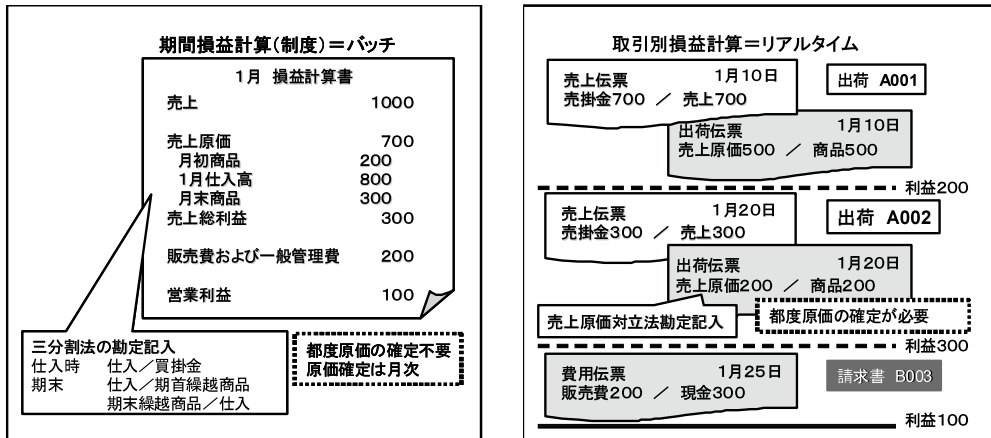


図10 期間損益計算と取引別損益計算

3.3 在庫評価方法と会計情報の更新

3.2節の勘定記入の方法と会計情報の更新の部分で、売上原価対立法の採択には在庫の払出単価の決定が都度必要と述べたが、払出単価を都度決定するためには、標準原価、移動平均原価法を採択する必要がある。

在庫の払出単価の決定を行う最も一般的な方法は先入先出法であるが、先入先出法は期末棚卸品を先に仕入れたとみなす、いわゆる「みなし計算」である。期末に残った在庫が後に仕入れた物と仮定し、払出単価を計算するバッチペースの考え方である(図11)。このため、どの原価のものが、どの取引だったのかを都度仕訳上で関連づけることができず、取引単位の原価の把握ができない。取引単位の原価を把握するためには標準原価、移動平均原価法の採択が必須である。

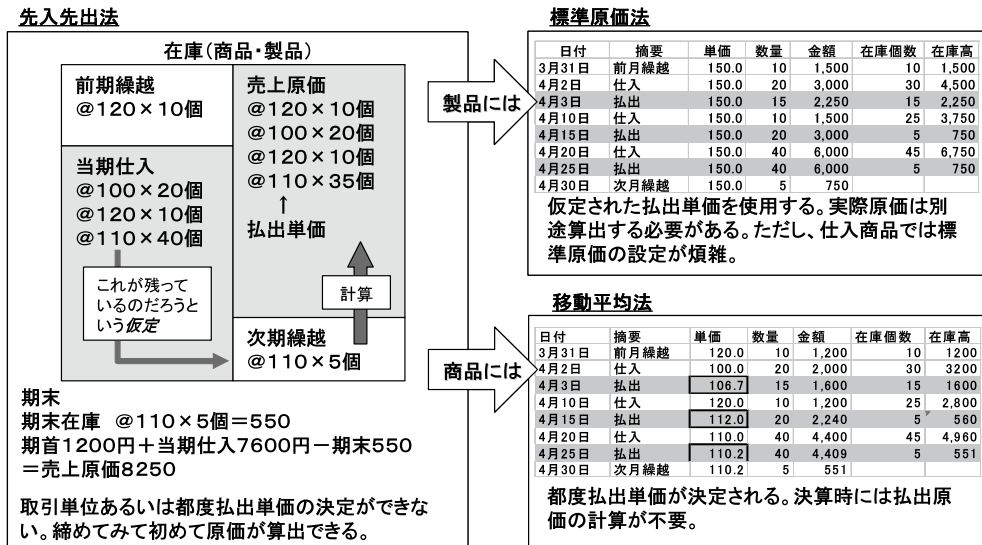


図 11 先入先出法と標準原価法・移動平均法

一般的なバッチベースのシステムから真の意味でのリアルタイムシステムへの移行は、業務的にも技術的にも難易度が高い、なぜなら、勘定記入において会計システム、在庫払出単価都度決定において在庫管理システム、取引単位の利益把握のための販売伝票と出庫伝票の紐付けにおいて販売管理システムの基本構造などに大きなインパクトを与えるからである(図12)(図13)。

Oracle EBSは、独立したモジュールをI/F経由でデータ連携し、最終的に仕訳で管理するというバッチベースの考え方に立脚している。I/Fの実施間隔を短くすることで擬似リアルタイムは可能であるが、厳密な意味でのリアルタイム損益把握の実現は難しい。Oracle社の場合、データベース技術のみならず、強固な技術基盤が、これら各種のアプリケーション連携を支えている。よって、販売・購買機能へのモジュール利用は、顧客独自開発システムを含め幅広い選択肢を想定しており、Oracle EBSではモジュール間の疎結合を重視している。

SAPは、業務統合を目指した密結合のパッケージであり、在庫元帳と会計がリアルタイム統合されており、移動平均原価を含め、取引別リアルタイム損益把握が可能である。しかし、他システムとの連携、あるいは単価後決めや一括請求などの商慣行の観点からも、完全なりリアルタイム業績把握を実現するには他システムの改良や業務改善も必要である。

以上のように、会計以外のアプリケーションのアーキテクチャが業績把握のタイミングの制約になり、会計システムのアーキテクチャ決定に影響を及ぼす。このため会計以外のシステムのアーキテクチャの分析も必要である。さらには密結合のシステムから疎結合のシステムへの変換も負荷が高いことから、今後はアプリケーション連携や統合の基盤はどうするかといったインフラ面でのアーキテクチャの検討も必ず行う必要がある。

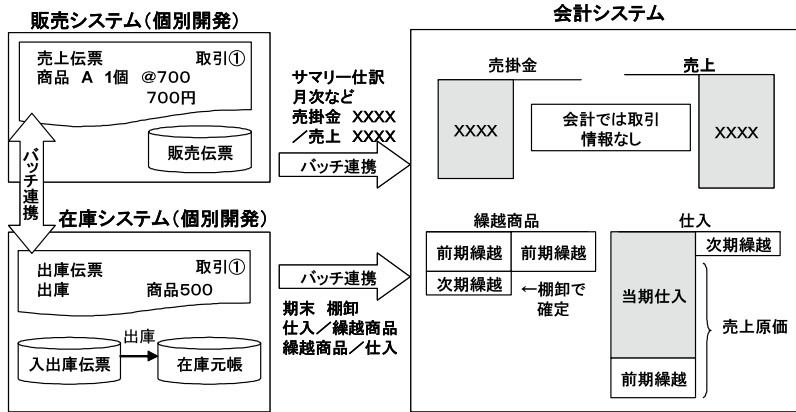


図 12 バッチベースのシステム

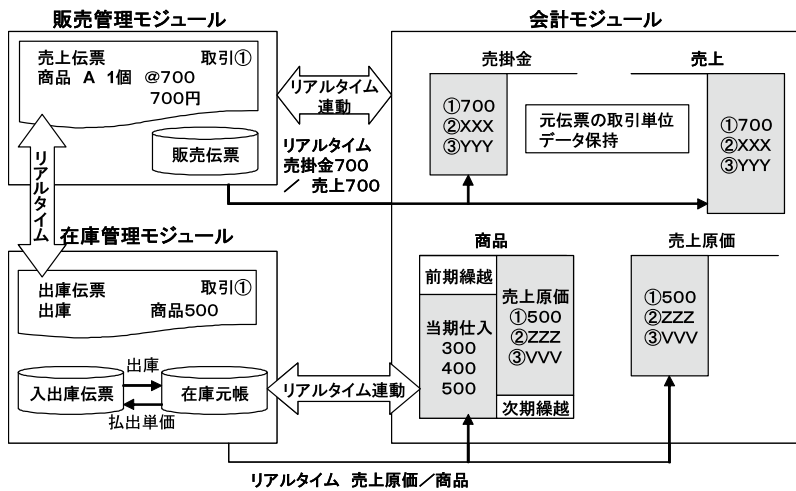


図 13 リアルタイムベースのシステム

4. おわりに

ERP (enterprise resource planning) は企業内のあらゆる経営資源を統合的に管理し、最適に配置・配分することで効率的な経営活動を行っていくという経営手法である。ERPパッケージとして“機能”比較の対象となることが多いSAPとOracle EBSであるが、採択する基礎理論および、その開発思想すなわちベースとなる考え方には大きな違いがある。

これらベースとなる考え方はソリューションのアーキテクチャ（データ構造やアプリケーション構成など）に大きな影響を与える。表面的に表れる機能のみを重視して導入すると、結果として導入目的に合致しないシステムが構築されることが多い。これを防止するためにはアーキテクチャの分析を行った上で、ソリューション全体の開発思想を推察・類推することが必要である。一般にこういった分析には、会計理論における知見の他に、複数パッケージの経験、ソリューション開発経験が必要となってくる。パッケージのパラメータの表層的な理解だけでは、とりえず稼働させることはできたとしても、稼働後の変化への対応、データ活用、システム拡張などの局面において全体最適は実現し得ない。

本稿では、主に総論をベースに検討をした。これは筆者らが従来より取り組んできたビジネスにおける知見の集大成であり、一応の成果は達成できた。しかし、各論部分や管理会計の詳細については、紙面の関係上あまり深く言及することができなかった。これらについては今後さらに研究を深めたい。

最後に、執筆のご指導をいただいた方々に感謝を申し上げ、本稿の結びとしたい。

-
- * 1 事実 (Fact) が1ヶ所だけに記録されているという意味。一つのデータは一つしか存在させないという考え方。
 - * 2 貸借バランスしない仕訳。通常はシステム内で入力できないように制限がかけられている。

- 参考文献**
- [1] 櫻井通晴, 「管理会計第二版」, 同文館出版, 2000年
 - [2] 古村彰・吉田博文・中尾宏, 「連結会計の導入と実践」, かんき出版, 2000年
 - [3] 森本朋敦, 「四半期開示に対応した連結経営管理マニュアル」, 税務研究会出版局, 2003年
 - [4] 広瀬義州, 「財務会計第8版」, 中央経済社, 2008年

執筆者紹介 籾原 弘美 (Hiromi Yabuhara)

1996年中央大学大学院商学研究科博士前期課程修了。1997年より会計コンサルタントとして主に製造業においてSAPのフィージビリティスタディ (SAPとEBSの比較評価・選定支援)、導入～保守に従事。

2008年10月よりERP統括プロジェクトにてOracle EPM (旧Hyperion社の経営管理ソリューション) を担当。



緒方 朱実 (Akemi Ogata)

1984年日本ユニシス(株)入社。ホスト間のシステム移行 (言語の文法変換, データ変換)、上流CASEツールの普及を経て、1997年よりOracle EBSの会計コンサルタントとして導入～保守に従事。2007年よりSAPとEBSの比較評価・選定を支援。現在は、知財活用センター知財化・適用支援室に所属。

