小売業における消費者の購買構造解析に対するシミュレーション

堀江育也

1．はじめに

現代の消費者要求は多様化し、同じものを求める時代から一人ひとりが自分の好みに合ったものを求める時代へと変わりつつあり、小売業だけでなく製造業やその他の産業に属するすべての企業が、ビジネスのあり方を問われるようになってきている。消費者は、良いものを安くだけでなく、欲しいときに手に入れられることをより重要視し、対応の素早さが企業や商品を選択する条件の一つとなっている[1)]。特に、小売業においては、品切れによる機会損失が、消費者の再来店意欲を減退させる要因の一つとなっていることが知られている。さらに、消費期限の短い属性を持つ商品は、期限切れによる廃棄が頻繁に生じるため、発注者は、機会損失および廃棄損失を抑えた、最適な発注量を決めるための意思決定が困難なものとなっている。

最も成功した小売業の業態にコンビニエンスストア（以下コンビニとする）がある。消費期限の短い商品である弁当や調理パン、惣菜などはFF（Fast Food）と呼ばれ、コンビニの売り上げの30％を占める重要なカテゴリーとなっている。FFは、購入後すぐに消費される食品であり、消費期限は他の商品よりも短く、弁当などは数十時間で期限切れとなるため、毎日、一日のうちに数回発注し何度か納品される。また、コンビニの商品寿命は短く、一年後には、半数以下の商品しか残っていないと言われる。従って、需要を維持もしくは拡大し、最大の利益を得続けるためには、消費者の要求にあわせた品揃えと、日々の過不足無い発注作業が、コンビニの存続において重要な課題の一つとなっている[2),3)]。各コンビニの発注者は、統一化された発注方法を使って発注を行っているわけではなく、限られた過去のデータ、

つまり、売上や発注、売れ残りなどのデータを参考に、その都度、経験的に発注量を決めているのが現状である。

従来、発注方法に関する研究は、在庫理論の定期発注法や定量発注法（発注点法）の枠組みの中で行われてきた。発注頻度は頻繁ではなく、経済発注量の決定には、発注費用や在庫費用などが使われている。このような在庫理論の枠組みの中でも、実需要に近い需要分布の研究がなされるなどの理論的な発展もみられる[4)]。しかしながら、その典型的な対象商品は、使用期限が長く、工場が発注するようなものと思われる。コンビニのように、短い消費期限を持つ商品を扱う小売店においては、無在庫が理想とされ、発注費用も固定的に考えられている。コンビニの弁当などの発注方法に近い問題を扱う在庫管理法には、「新聞売り子の問題」があげられる[5)]。新聞売り子の問題では、消費期限の短い商品を対象に、商品の小売価格、仕入れ価格に加えて、機会損失と廃棄損失の金額を仮定し、さらに、一定の需要分布を想定して最大の粗利益をもたらす発注量を決定している。ただ、現実においては、前述したとおり、直接把握することが困難な機会損失が需要に負の影響を与え、さらに商品の持つライフサイクルによって、需要分布は年間を通して常に変化している[6)]。

近年、コンビニで販売される商品に対する、需要予測が重要視されてきており、菅野ら[7)]によって新商品の売れ行きに対する新しい予測手法が提案されている。さらに、従来のマーケティング理論だけでなく、シミュレーションの必要性も示されている[8)]。特にシミュレーションは、対象とする事象のモデルが構築できれば、将来の予測だけでなく、様々な角度から仕組みを理解するための有効な手段ともなる。多様化する消費者の要求に、迅速に応えるためには、経験や勘だけでなくシミュレーションとの連携により、需要と供給の構造理解を深めることが、より発注時の意思決定の質を高めることにつながると思われる。ただ、人間の行動が含まれたモデルの構築には、シミュレーション結果をもとにした、モデルの変更や拡張の試行錯誤による、モデルの精緻化の作業が必要となることが多い[9)]。試行錯誤の過

程では、モデルやプログラム、結果だけでなく、メモやコメントなどが作成され、統一した表現形式の無いまま、それぞれが断片的な形で蓄積される場合が殆どであり、容易に再利用できる状態では管理されていない。従って、開発過程で得られたモデルやプログラムだけでなく、結果およびドキュメントなどとも関連づけて蓄積し、柔軟に活用できる環境が、新たなシミュレーション開発の負担を減らすだけでなく、さらに知識の共有を円滑にし、より妥当性の高いモデルの構築が可能となると思われる。

本稿では、はじめにシミュレーション開発支援環境のコンセプトを提案する。また、需要と供給構造を解析するシミュレーションの一例として、短い消費期限を持つ商品を対象に、消費者行動の影響を考慮した小売業発注モデルを構築し、シミュレーション結果の考察に至る過程を示し最後にまとめを行う。

2．シミュレーション開発の現状

シミュレーションは、工学、産業、経済など多方面の分野において、分析・予測を行う上で重要な技術の一つとなっている。シミュレーションを行うためには、対象とする事象のモデリングを行い、既存ツールの利用もしくはプログラムなどを作成して行う。既存の代表的な汎用シミュレーションツールの一つとして、米国 Rockwell Automation 社が開発した Arena[10),11)]があげられる。Arena は、GUI 環境を備え、モデルの作成から結果の視覚化まで行うことができ、様々な分野で実用的なシミュレーションツールとして広く用いられている。専用のシミュレーションツールを用いた場合、プログラミングの知識が無くとも、シミュレーションを行うことができるが、各ツール独自の操作方法を身につける必要があり、汎用性の高いツールになるほど操作方法もより複雑化することとなる。一方、専用のツールを用いない場合は、汎用言語を用いたプログラミングによる方法がある。代表的な汎用言語として FORTRAN があげられる。高速な計算

が可能であり、科学技術計算で広く用いられているが、モデルとプログラムとの関連性が高いとは言えず、また、新たなモデルに対しての再利用性は低く、はじめから作り直す場合が殆どである。従って、対象とするモデルが高度になるにつれ、モデルとプログラムとの対応関係が取りづらくなり、さらに、プログラミングミスも増大することとなる。また、汎用言語の他にシミュレーションのための専用言語も開発されており、GPSS、SIMSCRIPT、SLAM、SIMAN などがあげられる。これらの言語は、シミュレーション開発に対する種々の枠組みを提供したことで、従来の汎用言語よりも開発が容易となったが、各言語の特徴を知り開発方法を学ぶためには、やはり、直接関係の無い知識習得に多くの時間が必要とされる。このように、シミュレーション開発には、様々な開発環境が提供されているが、それぞれの環境間での可搬性は低く、作成されたプログラムや結果等を他の環境で柔軟に活用できるとは言えない。

他方、ハードウェアの高性能化とともに、基本ソフトウェアだけでなく、応用ソフトウェアも高機能化が進んでいる。開発に必要なプログラムの行数も数百万行から数千万行規模にいたるものもあり、開発効率のよいプログラミング言語だけでなく、ソフトウェア開発全体を含めたオブジェクト指向技術が、今日ではソフトウェア開発において不可欠な技術の一つとなっている。従来型のソフトウェア開発は、ウォーターフォールモデルが一般的であり、要求の把握、分析、設計、プログラミング、保守にいたる、上から下へといった一方向的なアプローチが主流であった。従来の開発方法では、各段階で用いるツールや考え方が異なり、ソフトウェアの大規模化だけでなく、多様化する利用者の要求にあわせた、小回りの効く開発ができなくなっていた。これらの問題に対して、オブジェクト指向は、前述の各段階において、一貫したオブジェクト指向の考え方を基本とし、小さなプログラムを作成し、設計を見直しながら成長させていく、スパイラルモデルを用いたアプローチが容易となった。また、オブジェクト指向開発におけるモデリングにおいては、従来、設計の表記法として 50

以上の規格が乱立していたが、Rational Software 社の Grady Booch、James Rumbaugh、Ivar Jacobson らの３人によって UML（Unified Modeling Language）が開発され、OMG（Object Management Group）[12]によって標準として認定された。現在、オブジェクト指向技術は、ビジネスアプリケーションソフトなどの開発だけでなくシミュレーション開発にも積極的に応用されてきている[13),14)]。

本稿が対象とする、消費者の購買行動を考慮したシミュレーションの場合、はじめからはっきりとしたモデルがわかっていることは少なく、単純なモデルからシミュレーション結果をもとに、モデルだけでなくプログラムの修正および拡張を行いながら、シミュレーションの有用性を高めていく必要がある。実際にシミュレーション開発を行ってみると、基本的なモデルからより現実的なモデルへと発展させていく過程で、現実と理論や予測していた結果などとの相違から、思いこみや勘違いなどに気づくきっかけが得られることが多い。しかしながら、このときの途中経過は、アーカイブ化されず、されたとしても多様な形式であり、基本的には最終的なシミュレーション結果だけに注意が向けられていることが多いと思われる。その結果、他の開発者が、同じようなシミュレーションを行う場合、同じ失敗や思いこみに気づくまでに時間を要したり、プログラム等の誤った再利用を招いたりすることが生じる。従って、シミュレーション結果を得るためだけでなく、モデルやプログラム、さらに、結果およびドキュメントやコメントを、開発環境に依存しない形で有機的にそれぞれを結合し、管理および活用できる開発支援環境が、効率的なシミュレーション開発だけでなく、蓄積された開発時のコメントやドキュメントの有用な参照を可能とし、より深い理解へとつながることと思われる。

３．シミュレーション開発支援環境のコンセプト

本稿では、図１に示すようにシミュレーションの開発過程をモデリング、プログラミング、リザルティングの三要素のサイクルとして考

える。各要素と開発サイクルの過程で生じるコメントやドキュメントとの有機的な結合を行うために、柔軟なデータ記述が可能なXML[15]による表現のコンセプトを提案する。

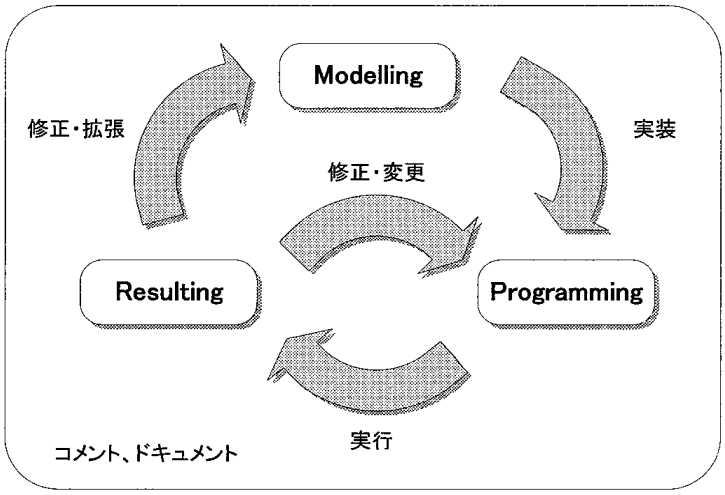


図1　シミュレーション開発の3要素

図2から図6に示したように、モデリング、プログラミング、リザルティングをそれぞれModellingML、ProgrammingML、ResultingMLとし、また、各要素に対するコメントおよびドキュメントをCommentML、全体をIMSSD（Intelligent Management System for Simulation Development）としてXMLで表す。本稿で提案するシミュレーション開発支援環境は、オブジェクト指向技術をベースとし、モデリングはUMLの使用を想定している。すでにUMLは、OMGが提唱するXMI（XML Metadata Interchange）によりMOF（Meta Object Facility）を基盤とした、モデル情報をXML形式へ容易に変換することが可能である。また、プログラミングにおいては、オブジェクト指向言語の一つである、Java言語の使用を考えている。Java言語もまた、G. J. BadrosがJavaMLによるソースコードのXML表現が提案されている[16]。ただし、モデリングおよびプログラミングにおいては、上記で示したものに限定するのではなく、最終的にXMLによって表現できればよい。また、シミュレーションの開発過程で生じる、各要素に対するコメントやドキュメント等の表現方法も図5に示すようにタグをつけて表す。このように、タグをつけて表

現することで、バックグラウンド上では、論理的な構造を表すことができ、コンピュータにおいても、タグ情報をもとに、各要素への容易なアクセスが可能となる。このような考え方は、Web サービスにもみられる。特に、現在 SOA が注目されており、ビジネスプロセスの構成単位を、標準的なインタフェースを持つソフトウェア部品とし、ネットワーク上に公開することで、俊敏な企業間の相互連携を可能とする、システムアーキテクチャが考えられ実用化されつつある。

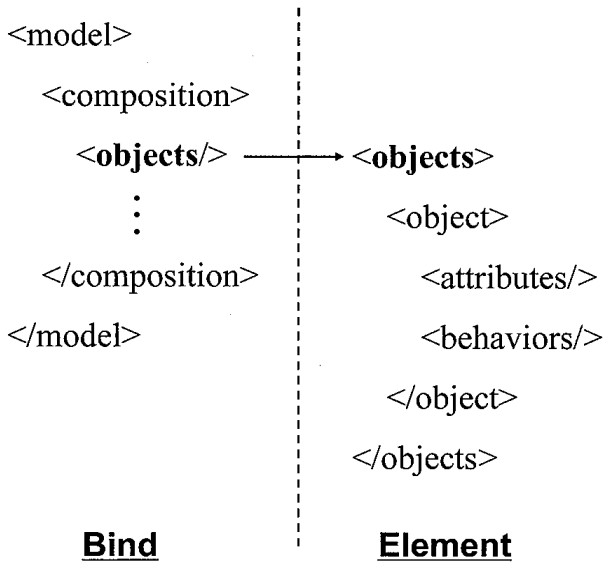


図２　モデルを表す ModellingML

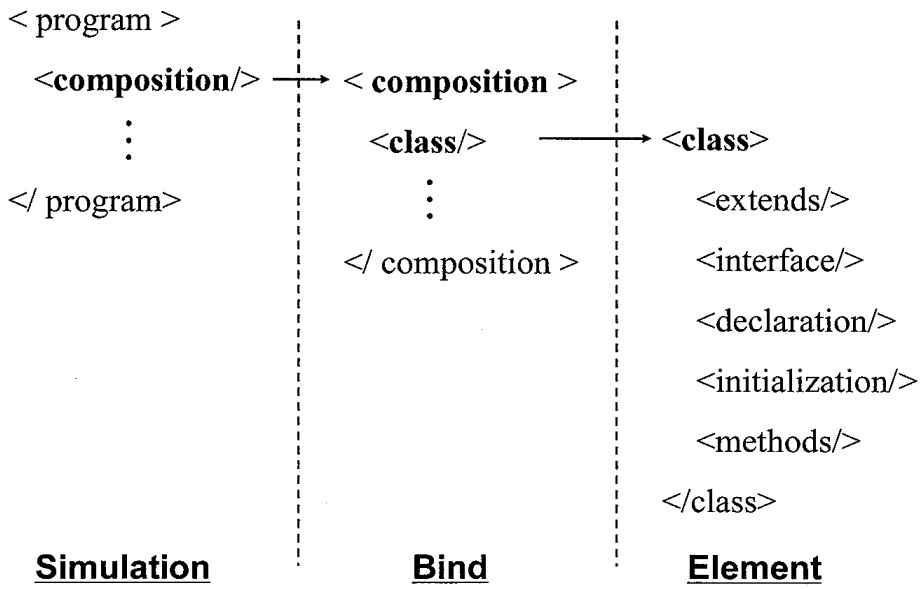


図３　プログラムを表す ProgrammingML

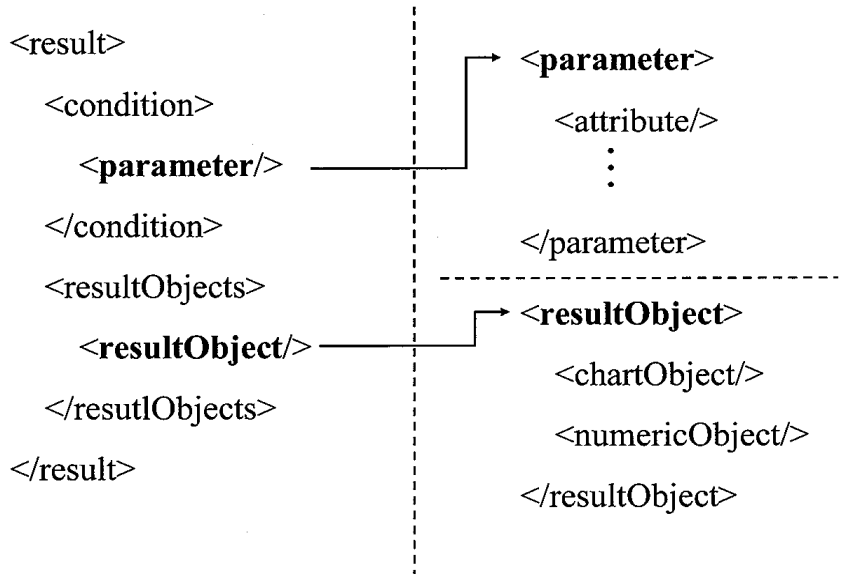


図４　結果を表す ResultingML

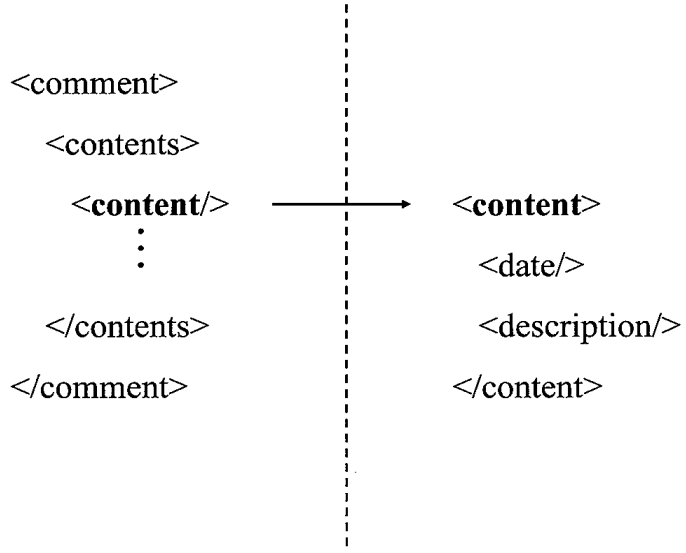


図５　コメントおよびドキュメントを表す CommentML

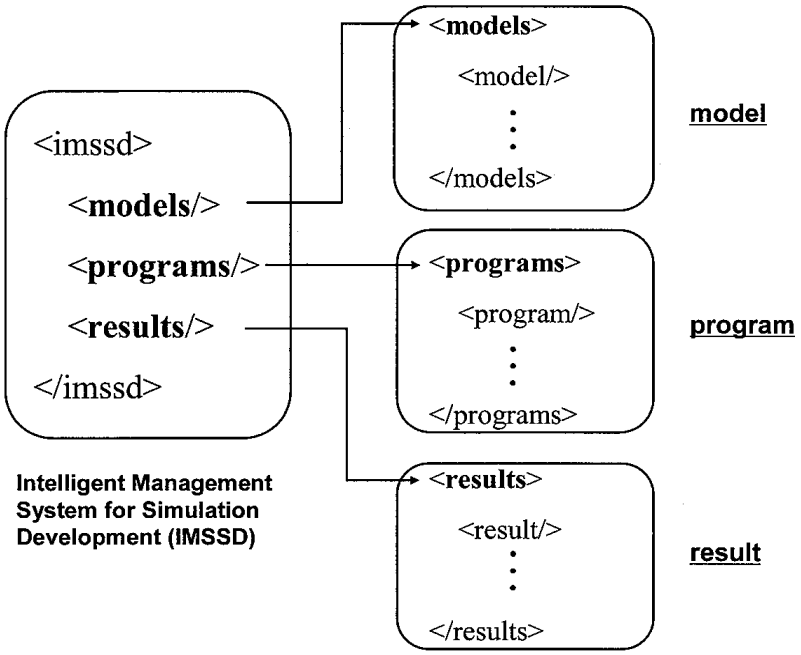


図６　シミュレーション全体をまとめる IMSSD

図７は、シミュレーションの開発支援環境のコンセプトを示したものである。各要素やコメント、ドキュメントに対する手作業によるタグ付けは容易でない。従って、入出力インタフェースとしてオーサリングの概念を導入する。一般的にマルチメディア要素を編集し、統合して一つのタイトルにまとめることが、オーサリングと呼ばれている。しかし、ここでは各要素やドキュメントをオブジェクトの階層構造で表現するために、すでに提案を行っているハイパーオーサリング[17]の概念を適用することを想定している。前述したように、シミュレーションは、修正や改良などを繰り返しながら進め、その過程で各要素やコメントなどが新たに作成されたり更新されたりすることとなる。従って、オブジェクトの階層構造で表現されたモデリング要素、プログラミング要素、リザルティング要素は、それぞれ過去に作成された要素を更新および統合して、既存のネイティブ XML データベースへ蓄積する。また、コメントやドキュメントは、ナレッジベースへ蓄積する。活用時においては、例えば、モデリングを行うとき、ナレッジベースが、モデリングおよびリザルティング要素が蓄積された

データベースをもとに、知的な検索を行い開発作業の支援を行う。また、過去に行ったシミュレーションを再現する場合、リザルティングにモデルやプログラム、パラメータ情報等をメタレベルで結合し、記録しておくことで可能となると思われる。さらに、外部接続インタフェースを設け、実験データや実際の販売管理データ等の柔軟な読み込みも想定している。以上のことから、提案する開発支援環境により、モデルの妥当性を高めやすくし、より実用的なシミュレーションを行うことができるものと思われる。

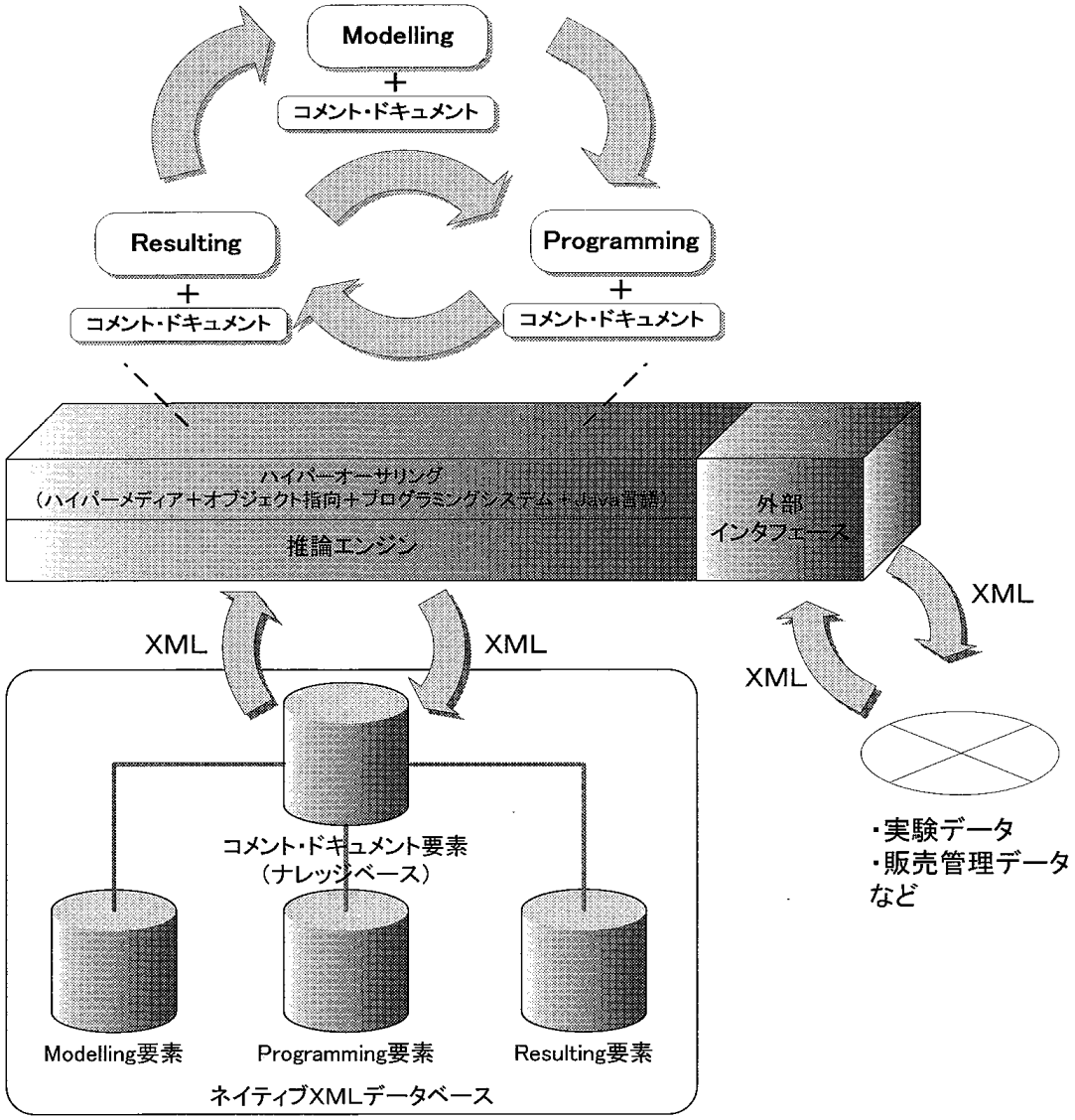


図7　シミュレーション開発支援環境のコンセプト

次に簡単な商品販売モデルをもとに、各要素のXMLによる表現の一例を示す。はじめに、基本的なモデルとして、「メーカ」、「小売店」、「顧客」の関係を図8に示す。図9は、図8で示したモデルをModellingMLで表したものである。すべてを表現しているわけではないが、モデルを構成する三つの要素がobjectsタグで一つにまとめられ、さらに上位のcompositionタグで名前付けしてある。また、必要に応じて作成者や作成日などの情報は、さらに任意のタグをつけて表せばよく、メタ情報を持たせた階層構造で表現が可能であることを示している。図10は、図8で示したモデルに対するシミュレーションプログラムをProgrammingMLで表したものであり、同様に、図11、図12は、結果とコメントをそれぞれ、ResultingML、CommentMLで表している。図10で示したProgrammingMLでは、declarationタグで、クラス内部で使用する変数および外部クラスをそれぞれvariablesタグとclassesタグでまとめており、クラスが持つメソッドはmethodsタグでまとめて表現している。図11は、シミュレーション結果をResultingMLで表しており、conditionタグでは、実行時のパラメータをまとめ、resultObjectsタグでは、グラフ作成時に必要な情報を、さらに詳細なタグで表したものを、一つにまとめていることを示している。図12は、CommentMLで、シミュレーション開発時のコメントを表しており、commentタグ内に、さらに詳細にタグをつけて表現している。ここで示した例は、完全ではないが、いずれも、階層構造で表現され、最終的には各要素を示す何らかのタグで表すことが重要である。通常、ビジネスソフト等で作成されたデータは、各ソフトウェアが画面上に表示するために必要な情報しかもっておらず、表示されているオブジェクト間の意味的情報は保持されていない。しかし、前述の通り、タグをつけて表現しておくことで、タグのトレースが可能となり、任意のタグに容易にたどり着くことができる。また、タグのトレースにおいて、ナレッジベースを活用することで、データベースが肥大化しても、容易なアクセスが可能となると思われる。

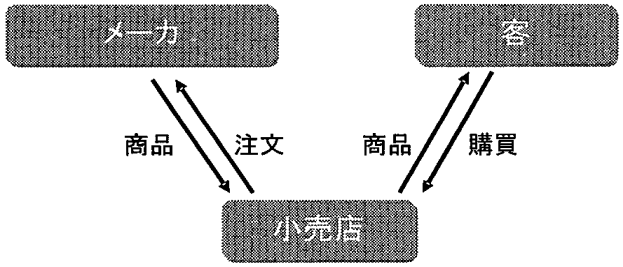


図８　商品販売モデル

```
<model>
  <composition name="Commodity sales simulation">
    <objects>
      <object name="store"/>
      <object name="customer"/>
      <object name="wholesaler"/>
    </objects>
  </composition>
</model>
```

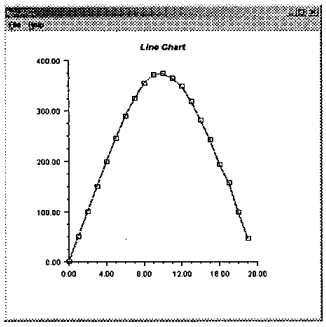
図９　商品販売シミュレーションにおける ModellingML の表現例

```
<class name = "simCostSell" >
  <declaration/>
  <methods/>
</class>

<declaration>
  <variables>
    <variable name="sell"/>
    <variable name="cost"/>
    <variable name="m"/>
  </variables>
  <classes>
    <object name = "p" instance_class="poisson"/>
  </classes>
</declaration>
```

```
<methods>
  <method name="sales">
    <variable name="nTry" value = "10000"/>
    <variable name="sum" value = "0.0"/>
    <branch>
      <condition logic= " number of customers & gt; Quantity of goods laid in">
        sum += x * (sell-cost)
      </condition>
      <condition logic=" number of customers & lt; Quantity of goods laid in">
        sum += g*sell - x*cost
      </condition>
    </branch>
    <return>
      sum/nTry
    </return>
  </method>
</methods>
```

図10　商品販売シミュレーションにおける ProgrammingML の表現例

```
<result>
  <condition>
    <parameter/>
  </condition>
  <resultObjects>
    <resultObject/>
  </resutlObjects>
</result>

<parameter>
  <attribute name = "purchase " value ="10"/>
  <attribute name = "customer " value ="10"/>
  <attribute name = "cost" value ="50"/>
  <attribute name = "sell" value ="100"/>
</parameter>
```




```
<chartObject>
  <Chart>
    <ChartTitle>
      <Attribute name="Title">Line Chart</Attribute>
      <Attribute name="FontSize" value="14"/>
      <Attribute name="FontStyle" value="3"/>
    </ChartTitle>
    <AxisXY>
      <Data y="{ 0.0, 50.0, 99.98, 149.7, 198.64, 246.05, 288.86, 324.57, 354.81, 371.75, 373.96, 364.93,
348.02, 318.82, 281.62, 241.77, 192.92, 156.13, 98.11, 46.93}">
        <Attribute name="LineColor" value="blue"/>
        <Attribute name="LineWidth" value="2.0"/>
        <Attribute name="DataType" value="DATA_TYPE_LINE|DATA_TYPE_MARKER"/>
        <Attribute name="MarkerType" value="MARKER_TYPE_HOLLOW_SQUARE"/>
        <Attribute name="Title" value="Blue Line"/>
        <ToolTip/>
      </Data>
    </AxisXY>
  </Chart>
</chartObject>
```

図11　商品販売シミュレーションにおける ResultingML の表現例

```
<composition>
  <class name="simCostSell">
    <comment/>
  </class>
  <class name="poisson"/>
  <class name="rndGene2"/>
  <class name="rnd_MT19937"/>
</composition>

<comment>
  <contents>
    <content>
      <date>
        2006/12/10
      </date>
      < description>
        The class which stocks and sells
        are trade.
      </description>
    </content>
  </contents>
</comment>
```

図12　商品販売シミュレーションにおける CommentML の表現例

４．小売業発注モデルのシミュレーション

より具体的な小売業発注モデルを構築し、シミュレーションを行い考察に至るまでの一例を以下に示す。

■シミュレーション概要

○商品の特徴と発注リードタイム

対象とする商品の消費期限は短く、１日３回定時の発注を想定する。店頭在庫以上の商品を持たないとし、発注費用や在庫費用は考えない。また、消費期限と発注リードタイムがほぼ同じ場合を考える。従って、発注時点での在庫は、納品までに販売あるいは廃棄のどちらかとなるものとする。

○需要の前提

シミュレーション期間は、１年間を想定し、最初はある需要からはじめ、徐々に増加し最大需要をとり、やがて減少していく図13に示したようなライフサイクルを持つものとする。また、欠品があった場合、需要曲線に影響を及ぼす場合を想定する。つまり、顧客が来店しても希望する商品が品切れの場合、一定の間再度来店しない場合を仮定する。

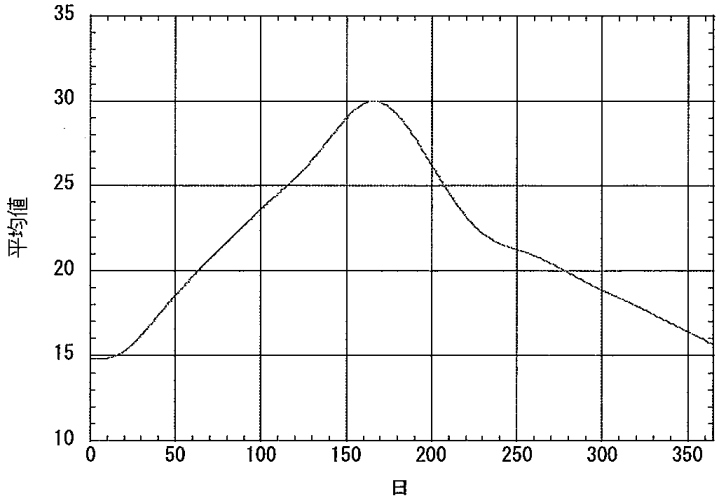


図13　商品のライフサイクルの一例

○発注方法

機会損失と廃棄損失をできるだけ出さない発注方法が最適とする。

本稿では、具体的な発注方法として、固定的な数を毎回発注する場合と、過去の販売実績をもとに発注する場合を想定する。

○機会損失と廃棄損失

需要が発生しても在庫がない場合は、欠品が発生し機会損失とする。消費期限が切れた商品は廃棄とし、仕入原価をそのまま廃棄損失とする。

以上の内容をもとに、シミュレーションを行うことを想定する。通常、はじめから詳細なモデルを構築するのではなく、図14に示したような、段階的なモデルの精緻化を行っていくこととなる。

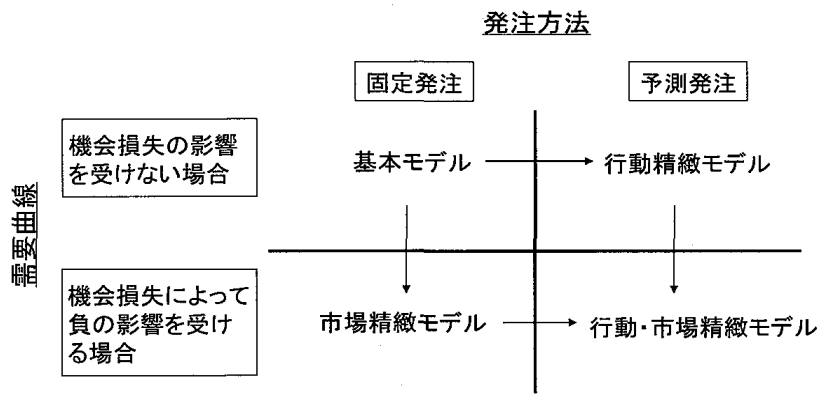


図14　モデルの精緻化の一例

図15は、図14に示したように、基本モデルをもとにいくつかのケースに分けたものであり、各ケースのシミュレーション結果の比較を行い考察する。

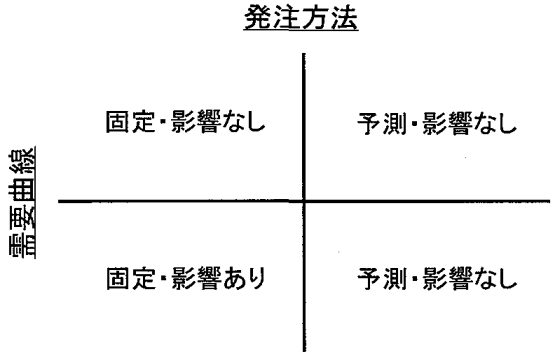


図15　モデルのケース分け一例

本稿では、各ケースの共通条件およびそれぞれの条件をもとに、モンテカルロシミュレーションを行うこととする。

■シミュレーション条件および方法

●共通条件

①試行回数　　：20,000
②期間　　：1年間（365日）
③初期在庫　　：15個
④リードタイム：3ステップ（1日を3分割）
⑤賞味期限　　：3ステップ（1日を3分割）
⑥発注：1日3回

●固定・変動なし（基本モデル）

①発注個数：20個
②需要は、機会損失に影響を受けない

●固定・変動あり（市場精緻モデル）

①発注個数：20個
②需要は、機会損失に影響を受ける
　需要曲線－(過去1週間の機会損失の平均×0.5)

●予測・変動なし（行動精緻モデル）

①発注量は過去2週間の販売結果をもとに予測※
②需要は、機会損失に影響を受けない

●予測・変動あり（行動・市場精緻モデル）

①発注量は過去2週間の販売結果をもとに予測※
②需要は、機会損失に影響を受ける
　需要曲線－(過去1週間の機会損失の平均×0.5)

※最小二乗法により納品時点の需要を予測し、任意の安全係数をかけ、予測在庫量を引いて求める。

■シミュレーション結果および考察

図16から図19に、各ケースのシミュレーション結果を示す。はじめに、図16は基本モデルの結果を示しており、発注方法を固定した場合である。仮定した商品のライフサイクルによって増加する需要量

を、発注量が70日目あたりから下回り、廃棄損失は発生しなくなるが、それに対して、機会損失が現れているのが確認できる。このモデルは、機会損失が生じても需要に対する影響は無いため、欠品がそのまま機会損失となって現れている。一方、図17は、図16に示したモデルをもとに、機会損失が需要に負の影響を与えるように拡張したモデルの結果である。70日目までは、機会損失の影響はみられないが、70日目から270日目の間で、需要が減少しているのが示されている。これは、固定した発注方法のため、途中から在庫量が需要量を下回り、機会損失を生じたことで、需要に負の影響を与え、需要量が減少すると共に、機会損失も基本モデルよりも減少していることが示されている。図18は、基本モデルをもとに、需要を予測しながら発注量を決定するモデルに拡張して得た結果であり、商品のライフサイクルの変化に沿って、適正な発注量が求められているため、機会損失を出さずに需要にあわせた販売ができていることが示されている。図19においては、さらに機会損失の影響を考慮しているが、図18で示した結果と、ほぼ同じ結果が得られている。また、いずれの結果においても、グラフが階段状を示しているのは、1日（3ステップ）の間では、需要は変化しないものとしているためである。

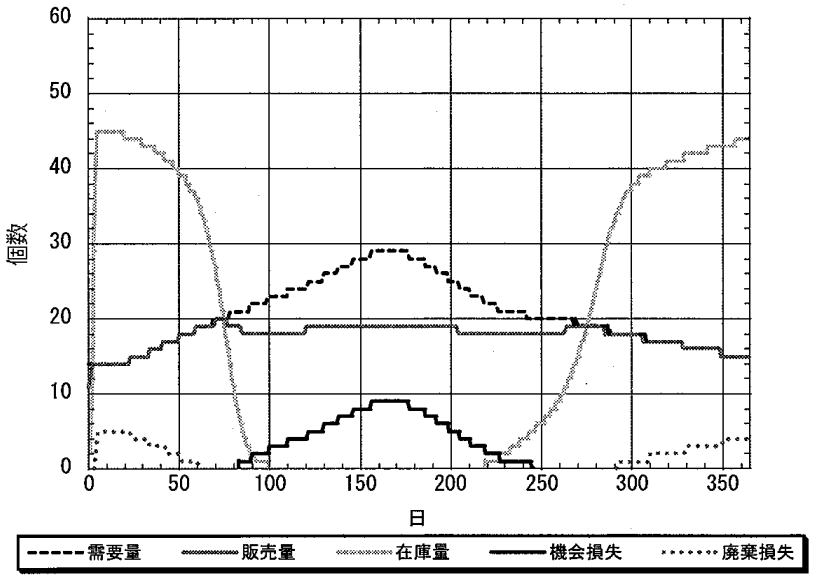


図16　固定数発注（基本モデル）

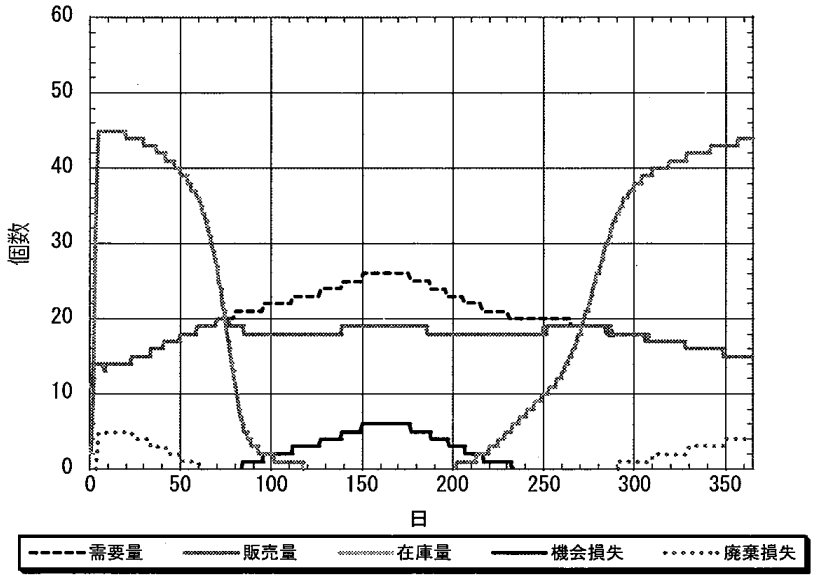


図17　固定数発注（行動精緻モデル）

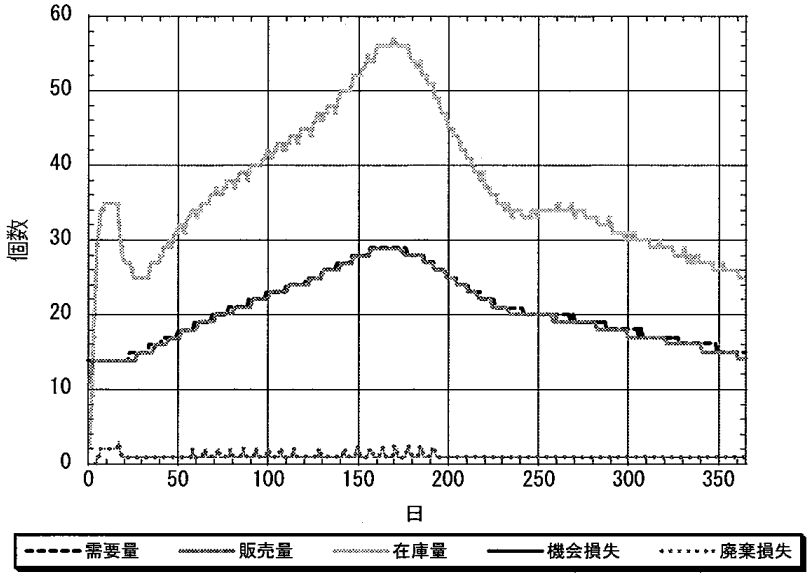


図18　予測発注（市場精緻モデル）

ここで得られた結果は、実際の現象を忠実に表せているわけではないが、シミュレーションを行うことで、現実では知ることが困難な機会損失や、商品のライフサイクルに対する在庫などの変化の仕組みを、具体的に知ることができる。さらに、仮定したモデルの結果と現実との比較を行うことで、消費者行動を考慮に入れた、需要と供給の構造をより深く理解するための手がかりとなると思われる。さらにモ

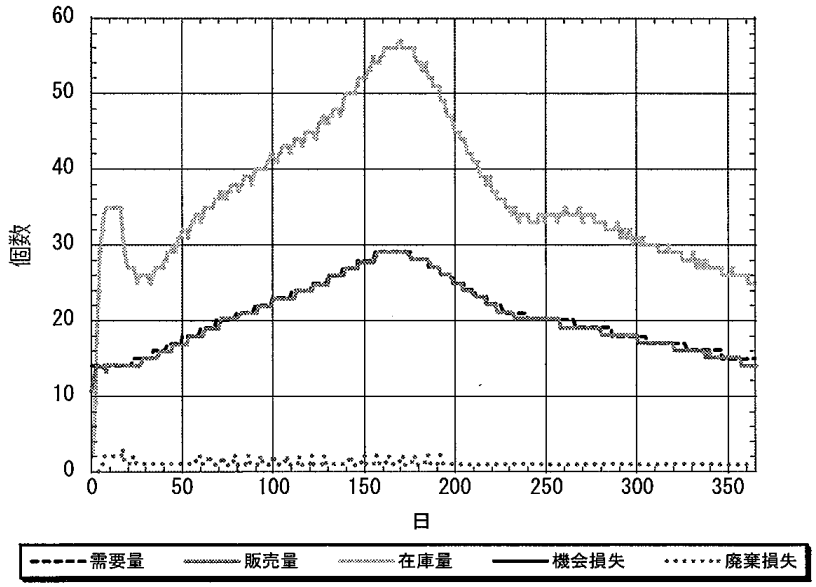


図19　予測発注（行動・精緻モデル）

デルの妥当性を高めていくことで、より現実に即したシミュレーションができるようになるとも思われる。ただ、上記の単純なモデルをもとにしたシミュレーションの開発だけでも、ここでは示していない様々な結果や考察時のメモやプログラムなどは、離散的に保存され再利用性は高いといえない状況になることが多い。これは、シミュレーションを行う目的が、結果をもとに考察することにあり、開発時のメモや一時的な結果、モデルの改良や拡張時の情報を逐一、再利用できる形で管理することは負担が大きいことがその原因の一つである。これは、開発者の怠慢とも言えるかもしれないが、既存の方法ですべての情報を保存したとしても、従来の検索方法では、必要な情報に対するアクセス時間も増大する。従って、様々な要因で変化する消費者行動を対象としたシミュレーションを行う場合、モデルを徐々に精緻化していく継続的な開発が必要であり、前述したようなシミュレーション開発支援環境が有用となると思われる。

5．おわりに

コンビニは、競合する他社の店舗だけでなく、同じチェーン店が、

近距離に複数出店している場合が多くみられる。消費者にとって、店舗を選ぶ自由度は高く、いつでも希望の商品が買えるという欲求を満たすことができない店舗は、存続が危ぶまれる状況になってきている。現場では、直接知ることのできない、消費者の行動を予測しつつ、消費期限切れによる廃棄損失を抑え、かつ品切れによる機会損失を出さないような発注は、きわめて重要な作業の一つとなっている。実際の発注作業は、経験や勘に頼って行われ、後継者や新規に始めようとする経験の浅い者にとって、適切な発注量を決定することは容易ではないのが現状である。本稿で、極めて簡略化したモデルをもとにした結果を示したわけであるが、シミュレーションは頭の中だけで考えるだけでなく、基本的なモデルの構築や修正、拡張などの試行錯誤により、実際の意志決定における思考の手助けとなると考えられる。しかし、一般的には、シミュレーションの開発は容易ではなく、誰もが簡単に利用できるとは言えない。本稿の前半で提案した、シミュレーション開発支援環境のように、単純な開発支援だけでなく、経験者などの結果に対するコメントや、試行錯誤の過程で得られるメモなどを、容易に参照し、任意のパラメータを与え、結果をもとに考察するだけでも、消費者行動をより深く理解できるようになると思われる。そのためには、モデル、プログラム、結果などと有機的に結合し、開発サイクルを効果的に循環させ、常に更新し全ての情報を活用できるようにしておくことが大切であると考えている。シミュレーションは、将来を予測するだけでなく、販売構造などを理解するための有効な手段の一つとなると考えられる。今後は、本稿で提案した開発支援環境をより具体化しつつ、より妥当性の高いモデルをもとにしたシミュレーション結果を示せるよう研究を進めていくことを課題とする。

本研究は、平成17年度札幌大学研究助成制度による研究成果である。

参考文献およびURL

1）黒田重雄，菊地均，佐藤芳彰，坂本英樹，“現代マーケティングの基礎”，千倉書房，(2001)
2）矢作敏行，“コンビニエンス・ストアシステムの革新”，日本経済新聞社，(1994)
3）矢作敏行，小川孔輔，吉田健二，“生・販統合マーケティング・システム”，白桃書房，(1993)
4）村瀬康比古，増田士郎，福田収一，大塚哲久，“広帯域な需要モデルに対するB1モデルの改良”，日本経営システム学会誌，pp.1-8，vol.21，no.1，(2004)
5）加藤豊，小沢正典，“ORの基礎”，実教況出版社，(1998)
6）Philip Kotler，“Marketing Management: Tenth Edition”，Prentice-Hall，2000)（恩蔵直人監修，月谷真紀訳，“コトラーのマーケティング・マネジメント　第10版”，ピアソン・エディケーション，(2001)）
7）菅野憲明，佐々木雄大，高山毅，池田哲夫，“コンビニにおける新商品発売時の売上データマイニング—目的変数と説明変数の導出—”，情報処理理学会，pp.545-552，(2004)
8）北中英明，“複雑系マーケティング入門”，共立出版，(2005)
9）堀江育也，北守一隆，佐藤芳彰，大森義行，“市場予測における知的シミュレーション開発環境に関する研究”，オフィス・オートメション学会第49回全国大会予稿集，pp.103-106，(2004)
10）W.D.Kelton，R.P.Sadowski，D.T.Sturrock，“シミュレーション　－Arenaを活用した総合的アプローチ－　第3版”，コロナ社，(2005年)
11）Rockwell Automation，http://www.arenasimulation.com/（2006年12月現在）
12）The Object Management Group（OMG)，http://www.omg.org/（2006年12月現在）
13）井庭崇，中鉢欣秀，松澤芳昭，海保研，武藤佳恭，“Boxed Economy Foundation Model: 社会・経済のエージェントベースモデリングのためのフレームワーク”，情報処理学会，Vol.44，No.SIG14（TOM9)，pp.20-30，(2003)
14）柴山司，西俊文，皿井伸明，松岡達，天野晃，松田哲也，野間昭典，“オブジェクト指向に基づいた生体機能シミュレーションモデル構築環境，”電子情報通信学会技術報告 MBE2003-05，pp.23-28，(2003)
15）The World Wide Web Consortium（W3C)，http://www.w3.org/（2006年12月現在）

16) G. J. Badros, "JavaML - An XML-based Source Code Representation for Java Programs", http://www.cs.washington.edu/homes/gjb/JavaML, (2000)
17) 堀江育也, 鈴木卓真, 北守一隆, "ハイパーオーサリング機能を持つ知的資産管理へのXML活用の視点", 日本生産管理学会論文誌, Vol.10, No.1, pp.67-72, (2003)