オブジェクト指向, エージェント技術の動向

非会員 中谷多哉子*1 非会員 羽生田栄一*2 非会員 黒川 利明*3
 非会員 直田 繁樹*4 非会員 大須賀昭彦*5 非会員 児玉 公信*6
 正員 増村 均*7 正員 中村 正規*7 正員 田中 立二*8

Latest Trend of Object Oriented Technology and Agent Technology

Takako Nakatani*1, Non-member, Eiiti Hanyuda*2, Non-member, Toshiaki Kurokawa*3, Non-member
 Shigeki Sugata*4, Non-member, Akihiko Ohsuga*5, Non-member, Kiminobu Kodama*6, Non-member
 Hitoshi Masumura*7, Member, Masatoshi Nakamura*7, Member, Tatsuji Tanaka*8, Member

Object-oriented technology is used for the software of the large range, and its application fields are increasing. Object-oriented technology is also accepted naturally in the field of Internet application. Consequently, developments of new kinds of object-oriented technologies have been performed. Agent technology has been developed based on an object-oriented concept, and may have a great evolution with applying to applications of Internet field. In this paper, authors introduce the latest trend and the example of application of such object-oriented technology and agent technology.

キーワード: オブジェクト指向, エージェント, デザインパターン, ソフトウェアアーキテクチャ, UML, MDA, 分散オブジェクト, CORBA, XML, Web サービス, セマンティック Web

Keywords: Object Oriented, Agent, Design Pattern, Software Architecture, UML, MDA(Model Driven Architecture), Distributed Object, CORBA, XML, Web service, Semantic Web

1. はじめに

1980年代に始まったオブジェクト指向技術も実用化の段階となり, 現在の計算機システムにおいては日常的に使われるようになってきた。また産業応用分野においても実用化が進み始めている。

これに伴いオブジェクト指向技術そのものについても多

くの観点からの技術的な進展, 深耕が進みつつある。

従来からの発展の方向としては, プログラミング言語, プログラミング方法論などからソフトウェア設計, システム設計と, システム全体の設計をカバーするより上流部分への適用である。また, その適用範囲, 分野が広がるにつれ, それらの統合, 統一的なモデルを規定するためのメタモデル定義の検討が行われている。

*1 (有)エス・ラグーン
 〒272-0143 千葉県市川市相之川 4-12-14-102
 SLagoon Ltd.,
 4-12-14-102 Ainokawa, Ichikawa, Chiba 272-0143
 *2 (株)豆蔵
 〒160-0004 東京都新宿区四谷 4-3
 Mamezou Co., Ltd.
 4-3 Yotuya, Sinzyuku-ku, Tokyo 160-0004
 *3 (株)CSK R&D 室
 〒169-8620 東京都新宿区西早稲田 1-21-1 早稲田大学西早稲田ビル
 CSK Corporation, R&D,
 Nishi-Waseda Building 1-21-1 Nishi-Waseda,
 Shinjuku-ku, Tokyo, 169-8620
 *4 富士通(株) プロジェクトA-XML XML応用技術部
 〒211-8588 川崎市中原区上小田中 4-1-1
 XML Application Technology Dept., PROJECT-A XML,
 FUJITSU Ltd.
 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki 211-8588

*5 (株)東芝 研究開発センターコンピュータ・ネットワークラボラトリー
 〒212-8582 川崎市幸区小向東芝町 1
 Computer & Network Systems Lab., Corporate Research & Development Center, Toshiba Corp.
 1 Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582
 *6 (株)エクサ 技術部 兼 エンタープライズソリューション事業部
 〒212-8555 神奈川県川崎市幸区堀川町 580 ソリッドスクエア東館 15 階
 System Research and Development Dept., EXA Corp.
 15F Solid Square East Tower, 580 Hrikawa-cho, Saiwaiku, Kawasaki, 212-8555
 *7 東京電力(株) 技術開発研究所
 〒230-8510 神奈川県横浜市鶴見区江ヶ崎町 4-1
 R&D Center, Tokyo Electric Power Company Inc.
 4-1 Egasaki-cho, Tsurumi-ku, Yokohama, 230-8510
 *8 ティーエム・ティーアンドディー(株) 保護制御事業部
 〒183-0057 東京都府中市曙見町 2-24-1
 Protection Systems Div., TMT&D Corp.
 2-24-1 Harumi-cho, Fuchu-shi, Tokyo 183-0057

分散処理やインターネット分野では機能あるいは情報単位での独立性、協調性が必要とされオブジェクト指向の考え方が自然な形として受け入れられている。CORBAなどに代表される分散オブジェクトにおいてもその方向や検討の焦点が変わりつつある。また、Java、http などにより発展してきたインターネット分野においても XML を中心としたインターネットアプリケーションの統合化への方向に進みつつある。ここでは不定形、疎結合なオブジェクト連携のための新しい技術が生まれつつある。

エージェント技術はオブジェクトに知的処理の機能と移動性を持たせその概念を発展させたものと考えられる。エージェント技術はインターネット分野への適用をベースにその活躍の場が広がりつつある。エージェント技術はインターネット上の知識の統合や人間の社会活動の支援などソフトウェアの役割の高度化をはかり新しい世界を切り開く可能性を持っている。

本解説においてはこれらのオブジェクト指向、エージェント技術に関する新しい方向について以下のとおり紹介する。

- ・ソフトウェア設計方法論：デザインパターン、ソフトウェアアーキテクチャ
- ・モデリング言語 UML の新しい動き
- ・分散オブジェクト CORBA の最新動向
- ・インターネットへの展開：XML によるオブジェクト連携、サービス統合
- ・エージェント技術の新しい動き：標準化、AgentCities、セマンティック Web

また同時にこれらに関連した事例として、オブジェクト指向による新しい生産システムおよび、エージェントを用いた販売促進システムについて紹介する。

2. デザインパターン、ソフトウェアアーキテクチャ

デザインパターンやソフトウェアアーキテクチャは、利用できるように体系化された、ソフトウェア設計の知見であり、ソフトウェアパターンという範疇に入れられている⁽⁴⁾。既知の知見を利用することで、経験の有無を問わず、より適切に開発上の問題を解決できるようにすることが、ソフトウェアパターンの目標である。

技術者の知見を利用するためには、それを文書化し、共有できる形にしなければならない。パターンを記述する際に用いられる文書の構造はパターンテンプレートと呼ばれている。現在までに提案されたパターンテンプレートには、発生する問題、その問題が発生する状況、その問題に対する解決策と例、解決策を採用する根拠を提示する項目が定義されている。パターンは、開発者が何らかの問題状況に直面したときに参照され利用される。このとき、直面している問題の状況と、パターンが想定している問題状況とが類似しており、パターンの解決策を選択するための根拠が、問題解決後に目指すべき状況と合致しているとき、パターンを適用することで問題を解決できるはずである。

設計では、仕様が変更されやすい部分と変更されにくい部分とを分割し、オブジェクト指向の継承、集約を用いて、それらを組み立てる。これらのオブジェクトの分割と組み立て方法を問題状況ごとに整理してまとめたものがデザインパターンである。特に Gamma らが提案したデザインパターンが有名である⁽⁵⁾。Gamma らは、23 種類の知見をパターンとして文書化し、オブジェクト生成、オブジェクトの構造、オブジェクトの振る舞いという三種類のカテゴリに分類した。これらのパターンには、実装の際に多数の条件分岐やコードの煩雑さを発生させる仕様の多様性に対処し、要求変更に対するシステムの頑健性を向上させる設計指針が示されている。

ソフトウェアアーキテクチャとは、システムの特性に合わせて、システムを分割し、組み立てる方法を定義した知見集である⁽⁶⁾。大規模なシステムでは、システムを一つの構築物として建造せず、より小規模の部分に分割して組み立てるという手法をとる。たとえば OSI 7 階層モデルに見られるような「階層型」や、一つの入力データを順次加工し、最終的な成果物を生成するシステムで用いられる「パイプ&フィルタ」などがある。前者は、各層の部品を、隣接する層以外の層の変更から独立させる効果を生む。後者は、それぞれの加工処理の入出力データ以外、処理部品間の依存関係が存在しない構造にできるため、部品の独立性を高くさせることができる。ソフトウェアアーキテクチャは、過去に開発された様々なシステムの特性と、開発で採用されたソフトウェアのアーキテクチャとを対応付けている。そのため、新しく開発するシステムの特性が判明すれば、その特性にあったソフトウェアアーキテクチャを利用して、システムの基本構造を設計することが可能となる。

デザインパターンやソフトウェアアーキテクチャを利用する際には、開発するシステムの特性に対応づけて適応させる作業が必要となる。そのため、利用する側には、自分が解決しなければならない問題状況の把握と、問題解決後に目指される状況とを識別できている必要がある。

3. モデリング言語 UML の新しい動き

〈3-1〉 UML2.0 と UML プロファイル 米国のオブジェクト指向技術標準化団体 OMG(Object Management Group)が、オブジェクト指向モデリングで利用するダイアグラム記法の標準化を目指し規格化したのが UML(Unified Modeling Language)である。

(1) UML の経緯 1997 の UML1.0 に始まり、UML1.4 が最新版である(2003 年 1 月時点)⁽⁶⁾。

(2) UML2.0 の特徴 UML1.x でビジュアルなモデル記述言語としての仕様は固まったが、主にメタモデルが完備でなく形式性・拡張性に欠けるという課題に答えるべく、メタモデルの構成を見直し MOF(Meta Object Facility)アーキテクチャのインスタンスとして UML メタモデルを定義しなおす作業が行われており、2003 年中には UML2.0 として規格化される予定。厳密な言語定義とシーケンス表、

状態図(継承可)、アクティビティ図、ペトリネットといった新たな記法や Action Semantics を含め、振る舞い記述が強化される⁽⁷⁾。

(3) UML の拡張機構とプロファイル UML には、その言語体系を自身で拡張するための機構が備わっている。ステレオタイプ、タグ付き値(プロパティ)、制約の3つの言語機構である。これらを用いることにより UML を利用目的に合わせて拡張することができる。OMG では、いくつかの汎用的な UML 拡張セットを UML Profile として用意している。EDOC 対応、リアルタイム対応、CORBA 対応、EAI(Enterprise Application Integration)対応等がある。

〈3・2〉 MDA について 現在、OMG では MDA(Model Driven Architecture)構想に取り組んでいる⁽⁸⁾。

(1) MDA の課題 COBOL、C、C++、Java、C#といった言語の違い、様々な OS やプラットフォーム(J2EE、.Net、Palm 等)の違い、急発展するミドルウェアやコンポーネント技術(CORBA、EJB、Web サービス)の多様性に適切に対応し、ビジネスシステムの整合性・信頼性・拡張性・相互運用性を確保できるシステム開発のあり方が検討されている。

(2) MDA の特徴 MDA では、PIM(Platform Independent Model)と PSM(Platform Specific Model)を独立させることで、ビジネスシステムの表現は PIM で行い、進化する技術プラットフォーム依存のシステムアーキテクチャは PSM で表現し、その両者の情報を用いてモデルコンパイラを通すことで、実行可能コードを生成するという考え方である。現在、CORBA、EJB、Web サービスに対する PIM-PSM マッピングがプロファイル化されつつある。

〈3・3〉 UML の EDOC 対応 EDOC(Enterprise Distributed Object Computing)とは、大規模なエンタープライズ向け分散システムの機能とアーキテクチャを指す。

(1) 大規模ビジネスシステムの特徴 こうしたシステムの特徴として、分散並列処理、大規模データ管理、大量トランザクション、信頼性や可用性への高いニーズ、オープンプラットフォームと相互運用性といったものが挙げられる。こうした大規模で複雑なビジネスシステムをデザインする前提として、ビジネスモデリングが重要となる。その目的で RM-ODP(Reference Model for Open Distributed Processing)にも適宜準拠しつつ考案された UML の拡張セットが UML Profile for EDOC である。これによって記述されたビジネスモデルを、実装プラットフォームである EJB(Enterprise JavaBeans)や .NET フレームワークといったコンポーネントにマッピングすることで、ビジネスシステムが実現できる MDA 構想をも併せて目指している。

(2) EDOC のためのモデリング EDOC では、ビジネスモデル記述のための要素として、ビジネスエンティティ、ビジネスプロセス、ビジネスルール、ビジネスイベントを導入し、これらを用いて、ECA(Enterprise Component Architecture)を構成している。ECA においては、複数のビ

ジネスオブジェクトが相互作用(コラボレーション)し、その中である役割(ロール)を演じることで、ビジネスプロセスが実行されると考える。こうして、オブジェクトモデルとビジネスプロセスとがつながることになる⁽⁹⁾。

〈3・4〉 UML のリアルタイム対応 近年、オブジェクト技術の組み込み系・制御系システムへの適用が広がっている。それに合わせて、実時間制約や振る舞いに対するきめ細かいモデル記述が要求されている。また QoS(Quality of Service)としてビジネス系システムでも時間制約を含むサービスの質に関する記述が要求されている。

これらに対応し、カプセル、プロトコル、ポート、メッセージ、キューといった実時間コンポーネント要素を追加して FSM(Finite State Machine)として厳密なモデル記述を可能にした拡張セットが UML Profile for Real-time である⁽¹⁰⁾。

このように、現在、産業界での UML 適用は急速に行われつつあり、それに対応したツール類も、ビジネス向け・組み込み向けを問わず増加している。

4. OMG CORBA の最新動向

2002 年 11 月に公開された CORBA の最新版は、CORBA3.0.1 と呼ばれる。直前の CORBA2.6.x に比べて、一番大きな相違点は、Minimum CORBA 及び Real-Time CORBA がそれぞれ、独立した規格文書となり、その代わりに、Fault Tolerant CORBA と Secure Interoperability という章が追加されたことである。

これは、CORBA の位置づけのみならず OMG という団体の性格も変わってきたことを物語る。最近、OMG 会長の R. Soley は、ことあるごとに、「メインフレームはなくならない、COBOL もなくならない」という発言を繰り返している。これは、システムの中に占める分散オブジェクトの位置づけの冷静な評価であり、CORBA が企業システムの中に確かな位置を占めた自信の表れでもあり、将来への布石でもある。

すなわち、OMG は、いまや、分散処理のための CORBA の標準化団体ではなくて、システム構築を支援する MDA(Model Driven Architecture)の標準化団体であって、UML もその有力なツールだという方向付けである。

MDA においては、現在、企業(および国家などの Enterprise)の業務定義(Business Rule)からビジネスモデルを経て、システムのプラットフォーム独立モデル(Platform Independent Model、略称 PIM)、プラットフォーム特化モデル(Platform Specific Model)、さらに、最終コード配備へのオートメーションが論じられている。

PIM においては、その先のプラットフォームが ORB だと限定するわけではない。むしろ、ウェブサービスの動向などを睨んで、SOAP、J2EE、.NET などが例題としてふんだんに出てくる。

この数年で、ウェブという形で、インターネットに端を発する分散処理の枠組みが常識となった。ウェブ上では、

分散処理もオブジェクトもなんら特別なもの珍しい存在ではない。むしろ、ユーザはそれと意識することなく、分散処理を行い、オブジェクトへメッセージを送り、オブジェクトからメッセージを受け取っている。

そこで問題になるのは、システム管理者にとっては1960年代と変わらないテーマ、システムの信頼性 (Dependability)、責任適及性 (Traceability) である。だから、CORBAにおいては、耐故障性であり、安全性であり、OMGとしては、MDAを推進し、担わなければならないのである。

余り話題になっていないが、MDA文書⁽¹¹⁾では、「形式的でない規定はモデルではない」(2.2.1 Model)と述べている。すなわち、システムの信頼性や安全性を保証するためには、もはや、形式的に検証可能な基盤の上に立つしかないと見切っている。そして、そのようなモデルから、変換によって、適合プラットフォーム上に最終的なシステムの実行コードが生成されるようになることを目指している。

5. インターネットへの展開:XMLによるオブジェクト連携からサービス統合へ

XMLは、文書記述のためのメタ言語であるSGMLを、インターネットでのデータ交換を主眼として、簡略化、再構成した言語である。HTMLがデータの画面上の表現を中心としているのに対して、XMLでは、汎用的なデータの表現が可能である。

また、XMLの標準化作業では、言語仕様の標準化だけでなく、XMLを利用するための技術の仕様 (パーサの仕様 DOM, データ変換の仕様 XSLT 等)⁽¹²⁾についても標準化が行なわれた。それらに対応した処理系が、Apacheプロジェクト⁽¹³⁾や主要ベンダ等から次々と提供され、インターネット上での利用環境は急速に充実した。

これらを受けて、種々のデータ表現をXMLを用いて標準化し、異なるプラットフォーム間でのデータ交換を容易化しようとする動きが盛んに行なわれている。特に、インターネット上での電子商取引のデータ交換の標準化手段として注目され、実用化が進んでいる。また、電子政府等の公共サービスの電子化手段としても採用されている。その他、研究段階ではあるが、知識表現の手段にXMLを用いる、セマンティックWebという技術も注目されている。

このような普及が進むうちに、当初のXMLの仕様について色々な技術課題が表面化してきた。しかし、それらに対する解決策も、標準の仕様として制定され、それに基づく実装が提供されつつある。

まず、XML文書スキーマの定義手段が問題になった。当初はSGML由来のDTD(Document Type Definition)という方式であったが、これには、大規模なデータを扱うために導入された名前空間の仕様への対応ができないことや、複雑なデータ構造を定義しづらいという問題があった。これに対して、XML schemaという新しいスキーマ言語の仕様が制定された。この仕様では、データ型やオブジェクト指

向的な型階層の概念が取り入れられ、定義の記述や再利用が容易になった。但し、この仕様は複雑過ぎる、との批判もあり、RELAX NG等の別のスキーマ言語の提案も行われている。

また、HTMLは、XMLによって再定義され、XHTMLという新しい仕様が制定されたが、より汎用的なリンク機能をXMLベースで実現したいという要望が出てきた。XLink, XPointerの仕様がそれであり、標準化作業が進んでいる。この仕様では、双方向リンクを定義したり、リンク定義をXML文書とは別に持つことで、対象のXML文書の記述を変更することなく、リンクを定義すること等が可能になっている。XLinkは、財務会計報告書のための標準XBRL⁽¹⁴⁾で採用され、実用化が始まっている。

XML文書を格納/検索する技術についても、盛んに技術開発が行われ、既に、多くの商用システムが開発/利用されている。これらは、それぞれ独自の方式で行われているが、大別すると、XML専用のデータベースを用いる手法とRDBにXML格納の機能を組み込む手法とがある。これらについては、XML Queryという問合せ言語の標準化を行う動きもある。

このようなデータの標準化に加えて、さらに進んで、インターネット上で提供されるサービスの連携を標準化する仕組みとして、「Web サービス」が提唱されるようになった。これは、従来、Webブラウザを用いて人が操作していたサービスを他のソフトウェアからも自由に呼び出せるようにし、インターネット上の様々なサービスを組み合わせた新たなシステムを構築する仕組みをXMLをベースに構築することを目指すものである。

Webサービスは以下の基本技術からなっている。

- SOAP: XMLをベースとしたメッセージ交換のためのプロトコル仕様
- UDDI⁽¹⁵⁾: Webサービスの登録、検索を行なうためのレジストリ規格。Webサービスの登録や発見、記述を可能とし、サービスの電話帳のような役割を果たす。
- WSDL: Webサービスのインタフェースを記述するためのXMLベースの言語仕様。UDDIのサービスディレクトリに登録する情報はWSDLを用いて記述される。

ユーザの条件を満たすサービスを、UDDIを用いて検索し、WSDLで記述されている、その検索結果を解釈し、SOAPプロトコルによって該当するサービスの呼出しを行う、という一連の作業を自動実行可能にすることを目指している。

このような仕組みを実現するため、全世界共通のWebサービスを登録するパブリックUDDIの試行が既に行われている。しかし、パブリックUDDIから検索されたサービスを実行することには、信用上の問題があり、現実のビジネスに適用するのは難しい面がある。このため、現在では、プライベートUDDIと呼ばれる、特定会員や企業グループ内に閉じたUDDIを用いたシステムが実現されようとしている。また、この方式は、企業内システムの統合(EAI)を標

準的かつ簡易に実現する手段としても注目されている。

技術的な課題としては、セキュリティの問題がある。SOAPの当初の仕様にはセキュリティの規定がなく、HTTP等の下位のプロトコルに任せるものとしていた。しかし、下位のプロトコルだけでは、なりすまし等の不正行為を防ぐことが難しく、重要な商取引を行う手段としては不十分であった。そのため、セキュリティの仕様についても新たな標準化作業⁽¹⁶⁾が進められており、それに対応した実装も提供されつつある。

6. エージェントの新しい動き

本節ではエージェント技術に関するホットなトピックをいくつか紹介する。

〈6.1〉 エージェント技術の標準化 FIPA (Foundation for Intelligent Physical Agents)⁽¹⁷⁾は、エージェント技術の標準化を進める非営利団体である。1996年にスイスのジュネーブで登録され、50前後の企業・大学が参加して活動が続けている。活動の主な目的は、異種エージェントシステム間の相互運用性を確保する標準仕様の策定である。2002年12月に、それまでにまとめられたエージェント通信言語(ACL: Agent Communication Language)やインタラクションプロトコル、エージェント管理などに関する23の仕様がFIPA全メンバーの投票を経て標準仕様(Standard)として承認された⁽¹⁸⁾。これを受けて、今後、産業・学術分野においてFIPA仕様の活用が盛んになっていくものと予想される。

〈6.2〉 FIPA仕様に基づくエージェントプラットフォーム FIPA仕様を先取りしたエージェントプラットフォームも次々と登場している。FIPA-OS(FIPA Open Source)⁽¹⁹⁾、JADE(Java Agent DEvelopment Framework)⁽²⁰⁾、JADEを携帯機器向けに軽量化したLEAP(Lightweight Extensive Agent Platform)⁽²¹⁾が有名であるが、この他にもZeus⁽²²⁾、April Agent Platform⁽²³⁾、Bee-agent⁽²⁴⁾、Grasshopper⁽²⁵⁾、microFIPA-OS⁽²⁶⁾などが知られている。多くのプラットフォームはWebページからダウンロード可能である。

〈6.3〉 AgentCities プロジェクト FIPA仕様に準拠したプラットフォームを用いて、インターネット上でさまざまなエージェントサービスを提供する国際プロジェクトAgentCities⁽²⁷⁾が盛り上がりを見せている。AgentCitiesプロジェクトはエージェントサービスとして各地域の情報を提供することからこの名が付けられた。グローバルにプラットフォームを管理するPlatform Directory⁽²⁸⁾を覗くと、現時点で100前後のプラットフォームが登録されており、国内からもAgentCity TokyoとSendaiがこれに参加している。このプロジェクトでは、当面はエージェントサービスの相互運用技術の開発を進めるが、将来的にはセマンティックWebやWebサービス、グリッドコンピューティングなどとの連携を図り、ダイナミックで知的かつ自律的なエージェントの実現を目指していく。

〈6.4〉 セマンティックWebとエージェント セマンティックWebとは、Web上のコンテンツの意味を取り扱う技術である。これによりWebコンテンツの意味を機械的に扱うことが可能となるため、エージェントが文字通り人間の代理人となって、Web上のさまざまな仕事を代行することが期待される^{(29)~(31)}。このような知的エージェントがWebコンテンツの意味を理解する上で、オントロジーが重要となる。オントロジーを記述する言語として、W3Cでは当初DARPAで開発されたDAML(DARPA Agent Markup Language)⁽³²⁾とEUで開発されたOIL(Ontology Inference Layer)⁽³³⁾を統合したDAML+OIL⁽³⁴⁾の採用が検討されたが、その後、これをベースとする新たな言語OWL(Web Ontology Language^(注1)⁽³⁵⁾)が提案され、今後はOWL中心に検討が進むものと思われる。

〈6.5〉 Webサービスとエージェント 前節で紹介したWebサービス分野にセマンティックWebやエージェント技術を応用する試みもある。DAML+OILに基づいて定義されたWebサービスのためのオントロジーDAML-S(DAML Services)⁽³⁶⁾の活動などがその例である。また、以下のような、サービスの意味を扱うWebサービス向け検索エンジンも開発されている。

現状のUDDIでは、検索の際にサービスの意味までは考慮しないため、今後、多数のWebサービスが登録された際に、その中から適切なサービスを探し出すのが難しくなる。Webサービス向けマッチメーカーは、この問題を解決するために開発されたWebサービス向け検索エンジンである^{(21)~(23)}。サービス提供者がWSDLとOWLを併用してサービスを定義すると、マッチメーカーはオントロジーや制約条件などの意味情報を活用し、利用者の要求に合った適切なサービスを探し出す。今後、Webサービスはダイナミックにサービスを統合して活用するものになっていくと考えられるが、そこではこの種の技術が重要な役割を果たす。

7. オブジェクト指向による新しい生産管理システム

〈7.1〉 はじめに 現在の製造業は、売れるものを作るために、市場との相互作用をより敏速に行う必要に迫られている。こうした要請に対しては、製品の小さな改良を頻繁に繰り返して開発リードタイムと市場投入期間を短縮したり、生産しながらも顧客注文の状況によって製造品を動的に作りかえる方法が有効である⁽⁴⁰⁾。これは日本の製造業が実現してきたことであり、日本の製造業の強みでもあった。

しかし、情報技術の進展にもかかわらず、多くの製造業の生産管理システムは、いまだに大量生産時代のMRP(Material Resource Planning)をベースにしており、敏速な生産を支えられない。以下では、上述の方法を支える情報システムとして、オブジェクト指向技術による新しい

(注1) 略称がWOLではなくOWLとなったことより、これに合わせてOntology Web Languageと定義する文献もある。

部品表管理システム (SPBOM) と、それを利用した生産管理システム (Rock'n Planner) を紹介する。

〈7・2〉 新しい部品表管理システム SPBOM 製品の多様化は、製造設備を大きく変更せずに、サイズや色などの属性値を違えたり、多くのオプションを組み合わせることで実現することが多い。これを MRP 型の部品構成表 (Bill of Materials) で扱うには、大多数の部品が同じで、ごく一部の部品が入れ替わっただけという重複の多いデータを用意する必要がある。現在、これが管理上の大問題となっている。多様性をうまく扱うには、次のような新しい考え方を導入しなければならない。

(1) 加工方法の記述 製品の多様性は、投入品目の違いと、切断や塗装、研磨、穿孔などの加工内容の違いによって生まれる。部品構成表では、製造品が製造プロセスで受ける加工内容の累積的な違いを疑似品目として表さざるをえない。疑似品目にまでコードを振ると、それは容易に爆発する。要するに、品目コードが多様性を扱いにくくしている。

1つの加工内容は、①投入される品目、資源、情報、②処理内容、時間、コスト、③産出される品目、副産物から構成され、これらを連鎖することで製造方法が導出される。導出された連鎖を、筆者たちは工程部品表 (Bill of Manufacturing) と呼んでいる⁽⁴¹⁾。

(2) 品目コードに替わるもの 多様性を扱えない品目コードに替わるものとして、品目群名とその属性値のリストを用いる。品目群とは、その製造工程で製造可能な品目からなる集合である。

品目群はその要素を特徴づける属性種を持つ。各要素は属性種ごとに固有の値を持つことで個体識別される。このとき、属性値は固定的な値だけでなく、「未定」の状態と、取りうる値の範囲 (連続値) や値の組 (離散値) として「半決定」の状態も表現できるようにする。こうすることで、曖昧な品目の識別や製造プロセスにおける製造品の変遷を記述できる。

(3) 属性値の伝播 子品目 (投入品) の属性値を、親品目 (製造品) の属性値から決定できる場合がある。これを属性値の伝播と呼ぶ。属性値が「未定」や「半決定」の状態でも、一定のルールを与えることで同様に扱える。こうすることで、サイズ違いや色違い品を少ない品目定義で表現できる。

〈7・3〉 新しい生産管理システム Rock'n Planner

柔軟な生産活動を支える新しい生産管理システムを、SPBOM を利用して開発した。このシステムの特徴は、製造品の属性値が曖昧な状態でも、先行的に着手し、作業進捗を見ながら製造指示を動的に変更していくことにある。この方法だと、顧客が注文を決定してから製品を入手するまでの時間を実質的に短縮できる。旧来の生産管理システムでは、品目コードが与えられるまで生産を開始できなかったのと対照的である。

製造品の属性値 (未定、半決定を含む) が与えられると、

図1 SPBOMのシステム構成

Fig. 1. System architecture of SPBOM.

工程部品表を導出し、それに基づいて在庫の引当、資源の割り当て、スケジューリングを行い、製造指示を発行する。注文変更や製造実績の変動によって属性値が変化すると、それを反映して、工程部品表が生成しなおされ、製造指示が再発行される。

〈7・4〉 オブジェクト指向による実装 工程部品表を実装するデータ構造は非常に入り組んだものとなる。データを集合として扱う関係データベースでは、このような状況を実現することは難しい。

品目をコードで識別するのではなく、抽象的な品目クラスと属性種クラスおよび属性値の協調としてとらえることが重要である。そして、属性値を加工プロセスと関連づけることで、製造品の属性値が、「未定」、「半決定」の状態から加工を受けるたびに変化していくものとして、製造指示を動的にスケジューリングできる。

SPBOM は、オブジェクト指向データベース上に JAVA 言語を用いて実装された (図1参照)。クライアントとの通信は CORBA を用いた。

〈7・5〉 柔軟な生産管理システムの実現 Rock'n Planner は、現在、化学プロセス系の企業と精密組立系の企業で本格的な導入作業を行っているところである。特に、化学プロセス系の産業では、製造品の品質が安定せず、作りすぎが問題となっている。Rock'n Planner の動的なスケジューリングによって作りすぎがなくなり、製造法の改善結果 (設計変更) が部品表に反映されて、より妥当な計画が立てられると期待される。

8. エージェントを用いた販売促進モデル

〈8・1〉 まえがき 近年、企業と顧客の間にてインターネットを含むオープンなネットワーク上での商品情報配信および顧客からの要望集約といったコミュニケーションが進展しつつある。WWWを用いた手法として、マーケティングが運用するバナー広告・クロズドな電子懸賞を用いた手法、顧客向けプレミアムとして音楽データ・画像データを配信する手法、バイラル⁽⁴²⁾な電子メール配信手法等があるが、契約成立時のみを目的とするのではなく、契約後の企業・顧客間の関係を維持・向上させる手法についての考察が進んでいる⁽⁴³⁾。商品を購入した各顧客向けに個別化されたWWWページやメーリングリストを用意し顧客向けに付

加的情報サービス等を配信する手法が見られるが、如何に顧客をそのような情報サービスに引き留めるかは常に課題となっている。ネットワーク上に、企業と顧客間の信頼で繋がつたコミュニケーション空間を形成する必要がある。

(8・2) エージェント環境 人間が行うタスクを補助するための手法の一つとして、ソフトウェアエージェントを利用するという、エージェント指向のアプローチが挙げられる⁽⁴⁴⁾。エージェントが具備すべき基本的機能としては、常にバックグラウンドで実行され、実行環境の変化やユーザの指示に応じ相応しいアクションを取るという連続性と、与えられた目標を達成するために適切なアクションを取るという自律性がある。ネットワーク上で企業と顧客間のコミュニケーション環境をエージェント環境と呼んでいる。エージェント環境は、チャンネル生成エージェント、チャンネルエージェント、ユーザエージェントで構成されている。各エージェントの機能を述べる。

(1) チャンネル生成エージェント 実環境からインプットされた顧客別購買情報に基づきエージェント環境内にチャンネルエージェントを生成する。

(2) チャンネルエージェント エージェント環境内で提供されるサービスと顧客間のパスを生成する。

(3) ユーザエージェント 顧客に送られてくる情報の中から、必要な情報は残しそれ以外を除外したりすることで、ユーザの情報認知を支援するエージェントである。

(8・3) 販売促進モデル 顧客との接点創出および販売促進を目的とし、チャンネルエージェントを商品販売促進用プレミアムとして用意し、ユーザが商品を購入すると商品に添付されたチャンネルエージェントが起動し、顧客一企業間の接点が創出される。顧客一企業間の関係は、チャンネルエージェントで継続的に維持され、チャンネル生成エージェントにより、購買履歴に応じたサービスが、顧客に提供される。また、企業は、顧客単位に顧客の購買活動を

継続的に収集でき、精度の高いマーケティング情報を収集できる。また、このモデルは顧客間でチャンネルエージェントを授受することが可能であるため、例えば同じ興味を持つ顧客同士がチャンネルエージェントを授受した結果、各顧客のエージェント環境にチャンネルエージェントが常駐するという形で、クチコミネットワークという強い紐帯(クリーク)の存在が表出化されるという性質を持つ⁽⁴⁵⁾。マーケターは特定の傾向を持つ顧客集合であるクリークに対し顧客向け情報を配信することが可能である。

図2はチャンネルエージェントのシステムアーキテクチャである。エージェント環境はサーバサイド及びクライアントサイドに分散配置されるJavaプログラムにて実現している。

(注) 本論文に掲載の商品の名称は、それぞれ各社が商標として使用している場合がある。

(平成15年1月24日受付)

文 献

Design Pattern, Software Architecture

- (1) F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal: "Pattern-Oriented Software Architecture: A System of Patterns", John Wiley & Sons (1996) (in Japanese)
金澤典子他訳: ソフトウェアアーキテクチャ, トップラン (1999)
- (2) J. Coplien: "Software Patterns", <http://www1.bell-labs.com/user/cope/Patterns/WhitePaper/SoftwarePatterns.pdf>
- (3) E. Gamma, R. Helm, R. Johnson, and J. Vlissides: "Design Patterns", Addison-Wesley (1995) (in Japanese)
本位田真一・吉田和樹監訳: デザインパターン改訂版, ソフバンク, (1999)
- (4) 中谷多哉子・青山幹雄・佐藤啓太: ソフトウェアパターン, 共立出版 (1999)
- (5) M. Shaw and D. Garlan: "Software Architecture", Prentice Hall (1996)
- UML, OMG, CORBA
- (6) OMG: "UML specification Version 1.4", <http://www.omg.org/technology/documents/formal/uml.htm> (2001)
- (7) OMG: "UML 2.0 Infrastructure/Superstructure RFP", <http://www.omg.org/uml/> (2002)
- (8) OMG: "OMG Model Driven Architecture", <http://www.omg.org/mda/> (2002)
- (9) OMG: "UML Profile for EDOC", <http://www.omg.org/cgi-bin/doc?pte/2002-02-05> (2002)
- (10) OMG: "UML 1.5 with Action Semantics", http://www.omg.org/technology/documents/modeling_spec_catalog.htm (2002)
- (11) OMG: "Model Driven Architecture (MDA)", ormsc/2001-07-01, (2001)

XML, Web Service

- (12) <http://www.w3.org/> W3C(XMLやSOAPの基本仕様の標準化団体)
- (13) <http://www.apache.org/> Apache Software Foundation
- (14) <http://www.xbrl.org/>
- (15) <http://www.uddi.org/>
- (16) <http://www.oasis-open.org/> OASIS(e ビジネス関連等の上位技術の標準化団体)
- Agent
- (17) <http://www.fipa.org/>
- (18) <http://www.fipa.org/repository/standardspecs.html>
- (19) <http://www.nortelnetworks.com/products/announcements/fipa/>
- (20) <http://jade.cse.lt/>
- (21) <http://leap.crm-paris.com/>
- (22) <http://more.btexact.com/projects/agents/zeus/>
- (23) <http://www.nar.fujitsulabs.com/aap/>

図2 システムアーキテクチャ
Fig. 2. System architecture.

- (24) <http://www.toshiba.co.jp/beegent/>
- (25) <http://www.grasshopper.de/>
- (26) <http://www.cs.helsinki.fi/>
- (27) <http://www.agentcities.org/>
- (28) <http://www.agentcities.net/>
- (29) <http://www.semanticweb.org/>
- (30) 萩野達也編:「特集セマンティック Web」, 情報処理学会誌, 43, 7, pp.708-750 (2002-7)
- (31) 溝口理一郎編:「特集 Semantic Web とその周辺」, 人工知能学会誌, 17, 4, pp.383-416 (2002-4)
- (32) <http://www.daml.org/>
- (33) <http://www.ontoknowledge.org/oil/>
- (34) <http://www.daml.org/2001/03/daml+oil-index.html>
- (35) <http://www.w3.org/2001/sw/WebOnt/>
- (36) <http://www.daml.org/services/>
- (37) M. Paolucci, T. Kawamura, K. Sycara, et al.: "Semantic Matching of Web Services Capabilities", in Proc. First Int. Semantic Web Conf. pp.333-347 (2002)
- (38) T. Kawamura, T. Hasegawa, A. Ohsuga, et al.: "Proposal of Semantics-based Web Service Matchmaking", in Proc. Int. Conf. on Computational Intelligence and Multimedia Applications, pp. 87-92 (2001)
- (39) 川村隆浩・長谷川哲夫・大須賀昭彦:「XML Web サービスのマッチメーカー」, 東芝レビュー, 58, 2 (2003-2)

Applications

- (40) MASP コンソーシアム編:「製造業の新アーキテクチャ開発/研究報告書」, MASP コンソーシアム研究会 (1998)
 - (41) E. M. Goldratt: "The Haystack Syndrome: Shifting Information Out of the Data Ocean", North River Press, 222/228 (1990)
 - (42) S. ゴーティン: バイラルマーケティング, 翔泳社 (2001)
 - (43) P. コトラ: マーケティングマネジメント, ダイアモンド社 (2001)
 - (44) S.Yashida, K.Kamei, T.Ohguro, and K.Kuwabara: "Shine: An Agent-oriented Framework of Network Community Support Systems", Technical Report of IEICE OFS2001-13, AI2001-18 (2001) (in Japanese)
- 吉田 仙・亀井剛次・大黒 毅・桑原和宏:「Shine: ネットワークコミュニティ支援システムのエージェント指向フレームワーク」, 電子情報通信学会技術報告 OFS2001-13, AI2001-18 (2001)
- (45) 安田 雪: ネットワーク分析, 新曜社 (1999)

中谷 多哉子 (非会員) 1980 年東京理科大学理学部応用物理学科卒業。1994 年筑波大学大学院経営・政策科学研究科経営システム科学専攻修士課程修了。1998 年東京大学大学院総合文化研究科広域システム科学系博士課程修了。博士(学術)。富士ゼロックス情報システム(株)を経て、(有)エス・ラグーンを設立。和歌山大学客員教授。オブジェクト指向分析・設計方法論、要求工学に興味を持つ。共編著に「ソフトウェアパターン」(共立出版)など。情報処理学会、電子情報通信学会、日本ソフトウェア科学会、ACM、IEEE Computer Society 各会員。

羽生田 栄一 (非会員) 株式会社豆蔵代表取締役社長 CEO。技術士(情報工学部門)。オブジェクト指向分析設計モデリング・開発プロセスとその適用に関するコンサルティング・教育に携わる。

黒川 利明 (非会員) 1948 年生まれ。株式会社 CSK フェロー。OMG では、UML 翻訳、MDA、Business Rule などに関わる。

直田 繁樹 (非会員) 1984 年、京都大学大学院工学研究科情報工学専攻修士課程修了。同年、富士通(株)入社。(株)富士通研究所にて、ソフト開発支援技術の研究に従事。2001 年より富士通(株)にて、XML 適用システムの開発支援技術の研究開発に従事。ソフト工学、システム性能分析等に興味をもつ。情報処理学会、ACM、IEEE Computer Society 各会員。

大須賀 昭彦 (非会員) 1958 年生。1981 年上智大学理工学部数学科卒業。同年(株)東芝入社。1985~1989 年(財)新世代コンピュータ技術開発機構(ICOT)に出向。現在(株)東芝 研究開発センター コンピュータ・ネットワークラボラトリー主任研究員。早稲田大学博士(工学)。2002 年より電気通信大学大学院 客員助教授ならびに大阪大学大学院 非常勤講師を兼任。主としてソフトウェアのためのフォーマルメソッド、エージェント技術の研究に従事。1986 年度情報処理学会論文賞受賞。情報処理学会、電子情報通信学会、日本ソフトウェア科学会、IEEE Computer Society 各会員。

児玉 公信 (非会員) 1978 年 3 月 東京都立大学人文学部(心理学)卒業。石油会社、北海道大学工学研究科(情報工学専攻)受託研究員などを経て、1992 年(株)エクサ入社。オブジェクト指向技術の社内外向けコンサルなどに従事。情報処理学会教育委員会生涯教育委員。技術士(情報工学)。

増村 均 (正員) 1968 年 12 月 26 日生まれ。1992 年 3 月山梨大学工学部卒業。1994 年 3 月山梨大学工学研究科博士前期課程修了。現在、東京電力株式会社技術開発研究所勤務。ソフトウェアエージェント、情報コミュニケーションなどに関する研究に従事。

中 村 正 規

(正員) 1952年5月12日生まれ。1975年3月日本大学生産工学部電気工学科卒業。同年4月東京電力(株)入社。現在、同技術開発研究所に勤務。主として、データベースシステム、キーワード抽出システムなどの研究開発に従事。

田 中 立 二

(正員) 1971年3月早稲田大学理工学部数学科卒業。同年4月(株)東芝入社。現在、ティーエム・ティーアンドディー(株)に勤務。分散処理、データベースなどのプラント監視制御システムへの適用に関する研究開発に従事。情報処理学会、人工知能学会、IEEE Computer Society, Power Engineering Society 各会員。