

平成15年度

製造業における情報技術活用促進補助事業
調査報告書

平成16年3月

財団法人 製造科学技術センター

序

わが国の製造業発展の原動力は、モノづくりによる付加価値の創造、それを実現した技術力及び優秀な人材にありました。しかしながら、製造業のグローバル化を背景に1地域・1国にとどまらない技術開発環境が求められており、この開発・実用化で後れをとれば、国際競争力の面で大きな影響を及ぼすことになりかねません。モノづくりを経済の基盤とする日本としては、わが国の優れた製造技術(MT)と最新の情報技術(IT)の連携を図り、製造に関わる情報統合・情報連携を企業における製造活動全般においてシームレスに行える技術を確立し、製造技術・製造活動の高度化を推進することが重要であるとともに、その国際競争力を維持・発展させてゆくためには、この基盤技術の国際的な標準化戦略を進めることも重要であると言えます。

このため、当財団では、製造業の産業競争力強化に資することを目的として、情報統合・情報連携のための技術環境及び標準化に関する調査研究について調査、検討致しました。

本報告書は、これらの調査結果をとりまとめたものであり、今後の製造業の持続的な発展に寄与できれば幸いです。

また、本調査事業の遂行にあたり、ご支援をいただきました日本自転車振興会ならびに経済産業省に御礼申し上げますとともに、調査にご協力をいただきました皆様に対し、深く感謝を申し上げます。

平成16年3月

財団法人 製造科学技術センター

理事長 亀井 俊郎

目次

| | |
|--|----|
| 第1章 製造業におけるXMLの利用に関する調査研究..... | 1 |
| 1.1 事業運営体制 | 1 |
| 1.1.1 組織構成 | 1 |
| 1.1.2 委員名簿 | 2 |
| 1.1.3 委員会活動状況..... | 2 |
| 1.2 XMLとは..... | 3 |
| 1.2.1 XMLの歴史的背景 | 3 |
| 1.2.2 XMLの特徴..... | 5 |
| 1.2.3 XMLとデータベース..... | 7 |
| 1.3 XMLの関連技術..... | 8 |
| 1.3.1 関連規格と応用規格..... | 8 |
| 1.3.2 Webサービスと関連技術..... | 10 |
| 1.4 XMLを用いた開発事例 | 15 |
| 1.4.1 XVL | 15 |
| 第2章 製造業におけるXML情報連携実証モデル技術に関する調査研究..... | 23 |
| 2.2 情報連携モデル案 | 25 |
| 2.2.1 概要..... | 25 |
| 2.2.2 ドメインオブジェクトの説明 | 28 |
| 2.2.3 MESアプリケーション連携モデル | 30 |
| 2.2.4 MES-装置間情報連携のモデル..... | 46 |
| 2.2.5 装置エンジニアリング情報連携モデル | 56 |
| 2.3 分野別モデリング対象調査..... | 62 |
| 2.3.1 加工ラインシステム..... | 62 |
| 2.3.2 機械加工 | 66 |
| 2.3.3 専用機 | 71 |
| 2.3.4 半導体製造..... | 80 |
| 2.3.5 ユーティリティ監視..... | 81 |
| 2.4 実装アーキテクチャの検討 | 84 |
| 2.4.1 モデル共通操作..... | 84 |
| 2.4.2 モデル共通操作ツール | 85 |
| 2.4.3 ツール開発における優先順位 | 85 |
| 2.4.4 ゲートウェイ | 86 |
| 第3章 製造業における製造システム技術基盤構築に関する調査研究 | 95 |
| 3.1 CADコンテンツ流通のための共通基盤構築 | 95 |
| 3.2 生産設備のインタフェース共通基盤の確立 | 96 |
| 3.3 遠隔作業支援システムのための技術基盤構築 | 97 |
| 3.4 実証実験..... | 98 |

第1章 製造業におけるXMLの利用に関する調査研究

20世紀の産業革命は、モノ（ハード）を効率的に生産するオートメーションによってもたらされた。そして21世紀は情報（ソフト）を効率的に伝達・加工する、ITによる産業革命の始まりで、「IT革命」と呼ばれる。「IT革命」の要因として次が挙げられる。

- 世界中にインターネットが広まり、パソコンや携帯電話から簡単に接続可能となった。
- マイクロソフト社のWindowsに代表されるパソコンOSが誰でも使用可能なものとなり、これらが普及した。
- 安価で高速・大容量なパソコンが普及した。
- 通信プロトコルが国際標準化され、世界中の企業と電子データ交換（EDI）が可能となった。

このインターネットの普及と電子データ交換が可能となったことにより、企業では業務形態が大きく変化し、新しいビジネス形態（ビジネスモデル）を容易に創造することができるようになった。経営革新も含めたこれらの代表されるものとして、eコマース、SCM、ERPなどがあり、これらによって取引形態も大きく変化している。これは電子商取引と呼ばれ、これらの略語として、BtoB、BtoC、BtoE、CtoCなどがある。BはBusiness（企業）、CはConsumer（消費者）、EはEmployee（雇用者）を意味する。具体的には、BtoBは「企業間取引」、BtoCは「企業と消費者の取引」、CtoCは「消費者レベルのやりとり」、BtoEは「企業内における社内情報伝達システム」を表している。現在の電子商取引市場のほとんどはBtoBが占めている。これらの代表的な例として、SCM（BtoB）「楽天市場」などの電子商店街や電子店舗（BtoC）がある。これらの実現のためには企業・組織の枠組みを越えた柔軟な情報の連携が必要となり、また、各種データ・情報の統合化及びデータの相互変換が必要になる。この手段としてXMLは世界中から注目され、現在では多く使用されるようになってきている。また、SOAP、WSDL等の周辺技術も整備や国際標準化が進められ、これらを使用したWebサービスが普及してきている。

一方、近年、人、モノ、金、情報に続く、第五経営資源として知識資源が脚光を浴びている。企業が新しいビジネスを創造し、自己革新を行うには、ナレッジの活用が欠かせない。ナレッジマネジメントが注目される所以である。これを実現するシステムがナレッジマネジメントシステムであり、ナレッジマネジメントシステムにはその特徴からXMLの情報共有・交換技術が有効であると考えられる。

1.1 事業運営体制

1.1.1 組織構成

財団法人 製造科学技術センター内に製造業XML検討委員会を設置し、製造業におけるXMLの活用に関する基礎調査を実施した。一部調査については委託により実施している。

1.1.2 委員名簿

● 委員長

新 誠一 東京大学大学院 情報理工学系研究科 システム情報学専攻 助教授

● 副委員長

西岡 靖之 法政大学 工学部 経営工学科 教授

橋向 博昭 (株)山武 アドバンスオートメーションカンパニー C P事業本部
マーケティング1部 部長

● 委員

福田 好朗 法政大学 工学部 経営工学科 教授

柿 良幸 クオリカ(株) インダストリアルビジネス事業部
ソリューション営業部 チーフエンジニア

佐藤 知一 日揮(株) 産業プロジェクト本部 ビジネスソリューション部
SCMグループマネージャ

田辺 繁美 オムロン インダストリアルオートメーションビジネスカンパニー
技術統括センタ 企画室 主幹

谷岡 雄一 清水建設(株) エンジニアリング事業本部 情報ソリューション本部長

茅野眞一郎 三菱電機(株) 先端技術総合研究所 制御システム技術部
コントローラグループ グループマネージャ

濱口 猛智 マイクロソフト(株) 製造ソリューション本部 製造ソリューション部
ソリューションデベロップメントマネージャ

水谷 正道 (株)東芝 電力・社会システム社 府中電力・社会システム工場
計測制御機器部 主幹

村上 正志 (株)デジタル 企画開発本部 V E C事務局 リーダー

● 事務局

瀬戸屋英雄 (財)製造科学技術センター 専務理事

岡宗 秀一 (財)製造科学技術センター F Aオープン推進室 課長代理

1.1.3 委員会活動状況

| | | | | |
|-----|-------|-----|--------|-------------|
| 第1回 | 2003年 | 4月 | 8日(火) | 14:00~17:00 |
| 第2回 | 2003年 | 5月 | 7日(水) | 14:00~17:00 |
| 第3回 | 2003年 | 6月 | 12日(木) | 14:00~17:00 |
| 第4回 | 2003年 | 7月 | 11日(金) | 13:00~15:00 |
| 第5回 | 2003年 | 8月 | 29日(金) | 14:00~17:00 |
| 第6回 | 2003年 | 9月 | 29日(月) | 14:00~17:00 |
| 第7回 | 2003年 | 10月 | 27日(月) | 14:00~17:00 |

| | | |
|------|----------------|-------------|
| 第8回 | 2003年11月27日(木) | 10:00~12:00 |
| 第9回 | 2004年1月19日(月) | 14:00~17:00 |
| 第10回 | 2004年2月23日(月) | 13:00~15:00 |
| 第11回 | 2004年3月23日(火) | 14:00~16:30 |

1.2 XMLとは

XML (eXtensible Markup Language) は、SGML、HTML をベースに開発された文書式定義用の言語である。近年、インターネットを中心に広く普及してきている。その歴史は、図 1 - 1 に示したように 1969 年に IBM が開発した GML を発端に 1998 年に W3C (World-Wide Web Consortium) がその仕様を発表した。本章では、XML の歴史的背景や XML の言語的概要およびその特徴を解説する。

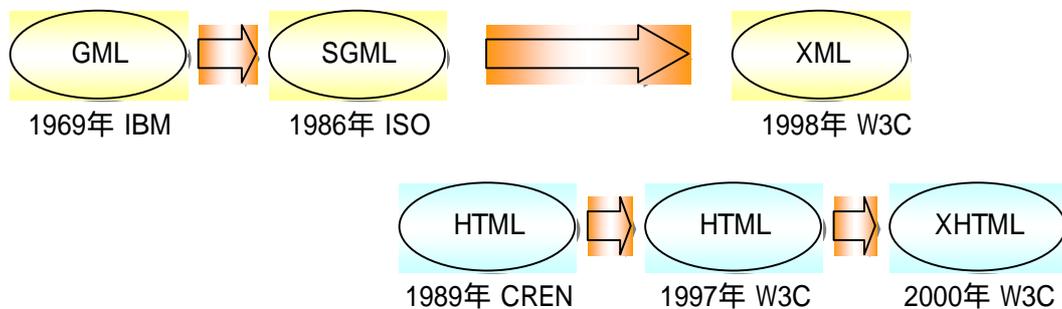


図 1 - 1 XML の歴史

1.2.1 XML の歴史的背景

(1) SGML

SGML (Standard Generalized Markup Language) (ISO 8879 の国際標準) は、文書全般をコンピュータで取り扱うための「汎用マークアップ言語」として、1986 年に登場する。さらに SGML の前身である IBM 社の GML (Generalized Markup Language) は 1969 年に考案され、その歴史は古い。SGML の標準化作業は 1970 年代後半から始まり、1986 年に ISO で規格化された (ISO 8879)。日本においても 1992 年に JIS X 4151 として規格化された。

その特徴は文書を表示情報と内容情報に分け、内容情報を構造化することにより、編集再利用を可能にした点で、画期的な意義を持つものである。そして全世界の全文書が対象となっており、取り扱う文字種や文字コードに制限がない。また、あらゆるデータに対応するための工夫として、記法宣言という仕組みが規定されている。記法宣言とは、外部のファイルとして参照するエンティティが XML 以外の記法 (TeX、EPS、GIF、BMP など) を使っている場合、その記法を識別するための名前などを指定、すなわち宣言することである。

このような優れた点を持ちながら、SGML はインターネット登場以前の仕様でありかつ複雑で

あるということから、あまり一般的に普及しなかった。

また、記法宣言は XML においても可能となっているが、しかし、インターネットでは、例えば HTTP を使用する場合は Web サーバがファイル形式を通知し、SMTP (メール用プロトコル) の場合ならば MIME によってファイル形式を通知する仕組みがすでに存在する。さらにアプリケーションで自動判定がある程度可能となってきた。したがって記法宣言をする必要は、実質的にはほとんどないと言っても過言ではない。

(2) HTML

次いで 1989 年に登場した HTML (Hyper Text Markup Language) は、SGML の流れを受け継いだもので、ネットワークを使った文書のユニバーサルなブラウジング方法を提供しようというティム・バーナーズ=リー (Tim Berners-Lee) の提案によるものである。SGML の仕様が非常に複雑であるのに対し、HTML には最小限の構造しかないこと、表現についての記述であることという点から、誰でも 1 時間もあれば見栄えのよい HTML 文書の作成法を学ぶことが可能である。

HTML より文書内容は、特定のコンピュータシステム環境に依存したり制約されたりすることなく、インターネットでコンピュータを結び、共通の約束に従って文書と文書をハイパー構造に連携することが実現された。このようなハイパー構造を持つ文書のことをハイパーテキストというが、このハイパーテキストの概念は 1945 年にアメリカのパネバー・ブッシュが論文「As We May Think」で提唱されたものである。また、この仕組みが WWW (World Wide Web) と呼ばれ、インターネットが爆発的に普及することとなった。

HTML は文書を表示することが主な機能であり、非常にシンプルな形で、習得が容易である反面、文書の内容を表現するには大きな制約があった。1990 年代後半に HTML が大規模で複雑なアプリケーションに適用され始めると、HTML の限界がささやかれるようになった。HTML は柔軟性に欠け、表現方法だけに限定されている。典型的な例としては、Altavista、Lycos、Excite のような Web クローラー (巡回型検索エンジン) が HTML の特定要素の「意味」を識別できないことが挙げられる。

そこで W3C は 1998 年に、インターネット上で構造化文書を縦横に利活用することを目標として XML の仕様を勧告した。これにより SGML の文書内容を多様に扱う手法がインターネット上で可能とされることとなったのである。

(3) XML の誕生

1996 年、W3C では、Web 機構に対する適合性を持ち、かつ人間側からみてもコンピュータ的にみても SGML より扱いやすい言語 XML の開発を開始した。その目的とする点は、HTML を基礎におく現状の Web 構造の限界を鑑み、Web 機構をより高度な情報交換・共有機構が可能な構造にするというものである。端的に言うと、図 1 - 2 に示したように SGML と HTML の利点を取り込んだということで、例えば、ネットワーク上に展開される膨大な量の情報源より、必要な

情報やサービスをより効率的な探索を実現すること、同じコンテンツをさまざまなメディアや体裁で表示可能とする機構の形成にある。併せて、利用者がネットワーク上の情報を自分のアプリケーションによって直接アクセスを可能とする点にある。

設計に当たって採用した原則は以下の通りである。

- 文書の構成要素を、「コンテンツ」と「文書構造」と「文書体裁」に分離する。
- XML は「システム対システム」インタフェースを確立するための規約とし、「システム対人間」のインタフェースは、利用者側アプリケーションに委ねる。(アプリケーションを特定しない。)
- SGML 規約の煩雑性、なじみにくさを排除する。
- HTML が抱える文書構造表現上の制約を取り払う。

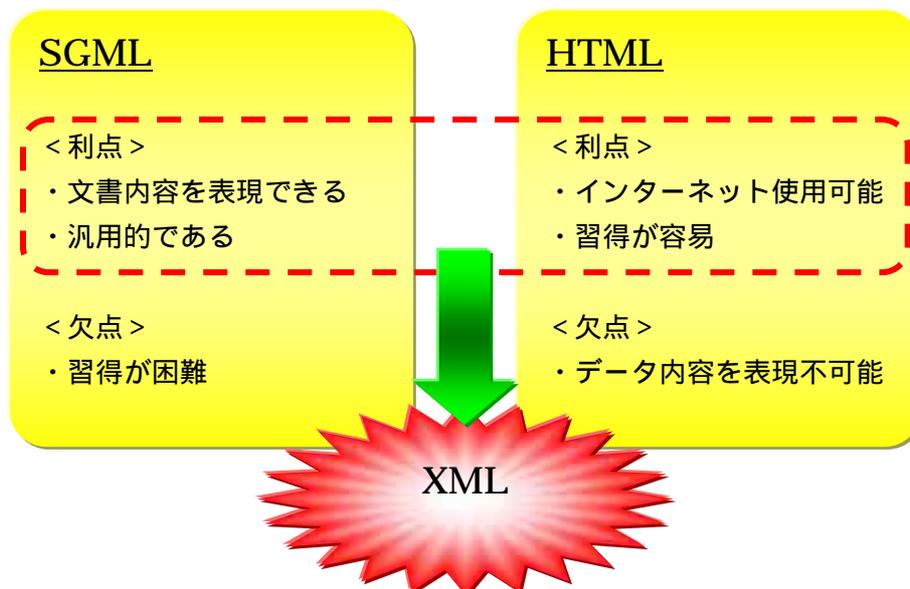


図 1 - 2 XML の誕生

1.2.2 XML の特徴

XML の文書はタグとテキストデータで構成されるため、プラットフォームやアプリケーションに依存しないデータ形式である。SGML をベースに開発されているが、SGML よりも簡単で、一貫性があり、曖昧さがなく、取り扱いを困難にしていた SGML の飾りや難解な拡張を取り除いている。SGML の構造化についてのすべてのアイデアは XML の中に明確に活かされている。また、XML の学習は SGML に比べて非常に容易で、その敷居ははるかに低くなっている。

XML の主な特長として次の点が挙げられる。

テキストデータであるため、汎用性が高い。

タグを利用して階層構造を表現可能なため、データを構造化して独自の属性を定義可能であり、データの検索性、再利用性に優れている。

(a) テキストデータであるため、汎用性が高い

テキストデータは、文字コードさえ分かれば、ハードウェアや OS を問わず、処理可能である。これがテキストデータの最大の利点である。図 1 - 3 に示すようにワードプロセッサの場合を例に挙げると、文字による文章の他に文字の装飾や罫線など、書式に関する様々な情報が加わってくる。このため、通常は保存されたデータはテキストデータではなく、特定のソフト専用の形式（バイナリデータ）となっていることが多い。テキスト形式であれば異なるプラットフォームでもデータを読み込むことが可能となる。

XML 文書はテキスト形式で記述されるため、プラットフォームの壁を越えてデータ交換が可能である。しかし、テキストデータは情報のわりにはデータのサイズが大きくなるという欠点もある。

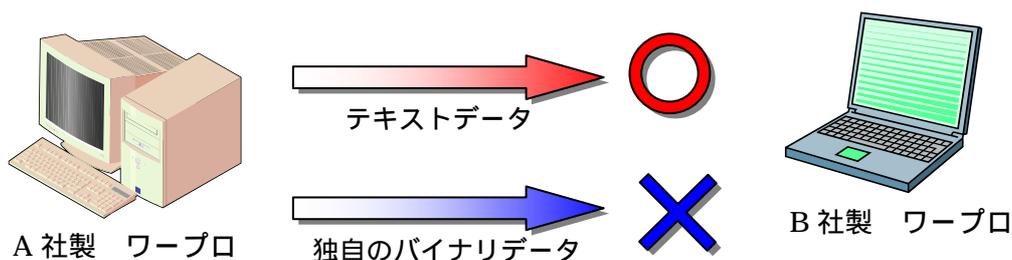


図 1 - 3 テキストデータの汎用性

(b) タグを利用して階層構造を表現可能なため、データを構造化して独自の属性が定義可能であり、データの検索性、再利用性に優れている。

XML は「マークアップ言語」である。マークアップとは印を付けることであり、XML では“<”と“>”を使用する。この“<”と“>”で囲まれた部分を“タグ (tag)”といい、マークの先頭を示すタグを“開始タグ”、終わりを示すタグを“終了タグ”という。また、開始タグと終了タグではさまれた部分を“内容 (content)”といい、タグと内容を合わせて“要素 (element)”という(図 1 - 4)。要素には“属性 (attribute)”を付けることができる。

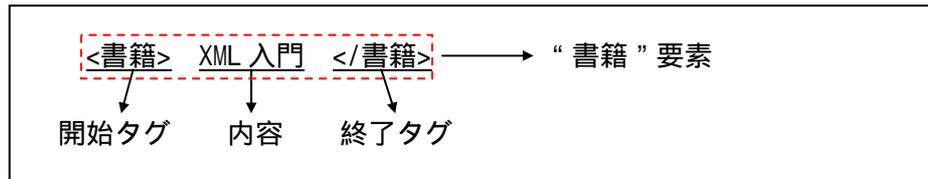


図 1 - 4 タグと要素

タグを使った記述方式を採用することで、データの意味やデータ構造を保持したまま、インターネット上でデータ交換が可能となる。さらに仕様変更や異なるシステム間でのデータ交換に柔軟に対応できるようになる。HTMLもマークアップして文書を記述する言語だが、XMLがHTMLと大きく異なるところは、使用するタグ名を“名前”とか“氏名”など、ユーザが決められる点だ。タグ名を要素内容の意味を反映させたものにすれば、そのXML文書を読む人間やシステムに、データの内容だけでなくデータの意味も伝えることができる。

さらにXMLは、タグでマークアップされた要素を入れ子構造にすることでデータ構造を表現することも可能である。これにより、タグ名と要素の入れ子構造というシンプルな方法で、データの意味とデータ構造を保持したままでのデータ交換を可能にしている。

データにどのような要素があるか、要素と要素にはどのような関係があるのかといったことを定めたものを文書型定義(DTD: Document Type Definition)という。文書型を定義することで、“データがどのような構成となっているか”を示すことが可能である。文書型を定義するには、スキーマ言語(schema language)が、必要でDTD、XML Schema、RELAX NGなどがある。

XMLは汎用のデータ記述言語として優れた特徴を持っている。しかしXMLは、プログラミング言語ではなく、あくまでもデータを表現する技術にすぎない。従って、XML文書を表示したり、文書構造を変換したり、アプリケーションプログラムから操作したりするXML関連技術が必要になる。

1.2.3 XMLとデータベース

情報の集中管理を実現するために、データの入力・更新・検索などの手順を提供し、様々なデータを体系的に整理して蓄積したものをデータベースといい、データベースを管理するソフトウェアをDBMS(DataBase Management System)という。

XMLのデータをデータベースとして管理するものをXMLデータベースといい、様々なソフトウェアがリリースされている。XML文書をデータベースに登録することで得られる利点は、データベースの特徴の一つである検索や追加、更新などが高速かつ容易になることである。また、バックアップ機能やログの管理、レプリケーション機能などを備えたデータベースであれば、データベースの保守も容易となる。

XMLデータベースには、「XMLネイティブデータベース(XML文書をそのまま保存できる)」と、「リレーショナルデータベース(リレーショナル形式にマッピングして保存する)」に大別さ

れる。

XML ネイティブデータベースは、XML 文書にインデックスを付け、DOM ツリーに変換するなど、XML 文書内の目的のデータを素早く検索、更新できるような工夫がされている。インデックス方式は、XML 文書の文書構造に沿ったインデックスを付ける方法であり、大規模なデータベースとなった場合でも、インデックスによって高速検索が可能である。DOM ツリーへの変換は、XML 文書を「ノード」に分解してツリー構造に組み変えることであり、このツリー構造を DOM ツリーという。DOM ツリーは、DOM ツリーから XML 文書への復元や、ツリーによる検索や置換操作も容易に行える特徴がある。

一方、リレーショナルデータベースは XML 文書のある部分を特定のテーブルの特定のフィールドにマッピングし、データベースに格納する。製品によってマッピング方法が異なるが、XML 文書の特定の要素（開始タグから終了タグまで）と、テーブルの特定の列を対応させてデータを取り出し、データベースへ格納させる方法が一般的である。また、同様の方法で、データベースに格納されているデータを XML 文書に組み立てて出力する機能も持つ。

XML ネイティブデータベースは、データベースから XML 文書を復元可能であるが、リレーショナルデータベースは完全な復元は不可能である。これは、XML 文書から特定の要素だけが抜き取られて保存されるため、元の XML 文書の構造はいったん破棄されるためである。一方、XML ネイティブデータベースには、問い合わせ機能が存在しない。一長一短があるようである。

1.3 XML の関連技術

1.3.1 関連規格と応用規格

XML が提供しているのは単なる構文であるため、アプリケーションとして活用するためこと、または XML をどのように利用するかは利用者（スキーマを定義する設計者に相当）に委ねられている。XML を活用するための関連標準や応用標準が多く存在する。これらについての概要を表 1 - 1 及び表 1 - 2 に示した。

表 1 - 1 関連規格

| 名称 | 内容 |
|---------------|--|
| Canonical XML | XML 文書を正規化するための仕様。 XML Signature で使用される |
| DOM | Document Object Model XML 文書にアクセスするための API の 1 つ |
| Namespace | Namespaces in XML XML 文書で使われるタグを目的に合わせて識別するための規格 |
| SOAP | Simple Object Access Protocol ネットワーク経由でオブジェクト間の通信を行う軽量のプロトコル |
| XForms | HTML で提供しているフォーム機能を分離独立させ、さらに強化することを目的とした規格 |
| XHTML | The Extensible HyperText Markup Language HTML 4.0 を、XML 1.0 の構文で再定義した規格 |
| XLink | XML Linking Language ほかの XML 文書と併用して、ハイパーリンクを表現するためのリンク言語 |
| XML Schema | XML のスキーマ言語の 1 つ |
| XML Signature | XML 文書に対して電子署名を行う書式を規定 |
| Xpath | XML Path Language XSLT や XPointer で利用され、XML 文書内の特定の部分を指し示す構文を提供 |
| XPointer | XML Pointer Language XLink とともに使用されるリンク言語で、特定の XML 文書の、特定の場所を指し示す構文を提供 |
| XSL | Extensible Stylesheet Language XML 専用のスタイルシート言語 |
| XSLT | XSL Transformations XML 文書の構造変換を行う規格 |
| XML Infoset | XML Information Set XML 文書の情報構造を明示的に扱うための定義 |

表 1 - 2 応用規格

| 名称 | 内容 |
|-----------|--|
| CHTML | Compact HyperText Markup Language 携帯電話や PDA などの携帯情報端末で閲覧可能な Web ページを作成するための記述言語。 |
| MathML | Mathematical Markup Language 数式を記述するためのマークアップ言語。 |
| RDF | Resource Description Framework メタデータの処理を扱うための規格。 |
| SMIL | Synchronized Multimedia Integration Language 時間に関するコントロールや、スクリーン上のプレゼンテーションのレイアウトなどを行うことを目的とした、マルチメディア記述用の規格。 |
| SVG | Scalable Vector Graphics ベクターグラフィック言語。 |
| Voice XML | Voice Extensible Markup Language 音声認識を行うための規格。 |
| WIDL | Web Interface Definition Language Web 上でのインタラクションを統一された方法で操作可能にするための規格。 |

1.3.2 Web サービスと関連技術

(1) Web サービスの概要

Web サービス (Web Service) の目的は、インターネット上に分散するアプリケーションの提供する機能を、クライアントのアプリケーションにプログラミング可能なインタフェースとして提供することである。アプリケーションの提供する機能をサービスといい、Web を利用してインターネット上で展開される様々な「サービス」を連携できるようにする仕組みやアプリケーションそのものを Web サービスという。また、XML Web サービスと呼ばれる場合もある。

Web サービスの基礎的技術には、SOAP、WSDL、UDDI があり、すべて XML がベースとなっている。これらを用いた Web サービスの構成図を図 1 - 5 に示した。このとき、検索及び登録は UDDI を使用する。また、接続には SOAP を用いる。

SOAP (元は Simple Object Access Protocol の略称であったが、現在は固有名詞として扱われている) はネットワークに接続したプログラム同士がメッセージ通信するためのプロトコルである。SOAP で用いるアプリケーションに渡す引数や戻り値は、XML 形式で定義されている。また、このアプリケーションを利用するためのインタフェースを記述するための言語が WSDL (Web Service Description Language) である。一方、UDDI (Universal Description Discovery, and Integration) はどのようなサービスが利用できるかを検索できる仕組みのことであり、Web サービスを公開するためには UDDI に登録する必要がある。

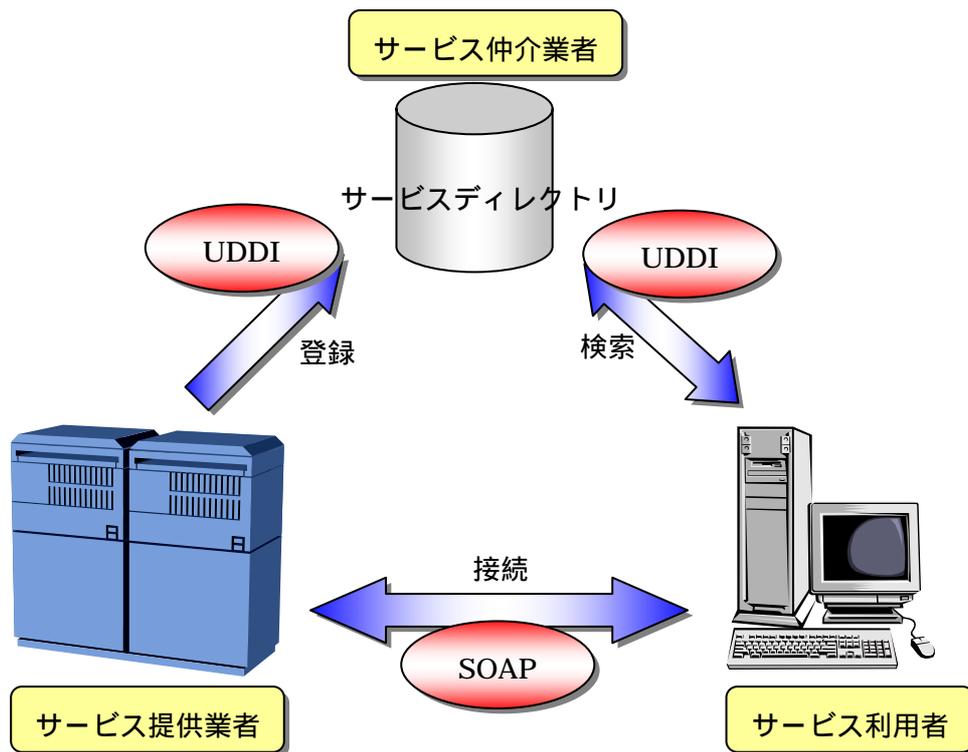


図 1 - 5 Web サービスの構成図

(2) Web サービスの特徴と利用

XML 関連仕様の中で最も注目されているのが Web サービスであろう。それは Web サービスによってインターネット上に分散されているサービスを自社のプログラムの一部として利用できるからであろうと推測する。Web サービスで使用されるサーバは、Web サービス毎で分散されており、サーバが分散されていることで回線の負荷も低減される。また、サーバが分散されていることでシステムの信頼性も向上される。複数のサービスをうまく連携させることで自社にはない新しいシステム、あるいは新しいビジネスの構築が可能となり、その可能性は今後も広がっていくであろう。

分散システムの構築する技術には、RPC、CORBA、DCOM などがあり、それぞれ実際に活用されている。これらと Web サービスの違いはプログラム（サービス）間の結合度にある。Web サービスは、個々のプログラムの独立性が高く、連携方法も非常にシンプルである。必要なときに必要な要求（リクエスト）を出すだけで、所用の結果を得ることが可能である。これは Web サーバと Web クライアントの関係に似ており、このシンプルさが利点でもある。

Web サービスの利用に関する概要を図 1 - 6 及び図 1 - 7 に示した。その特徴は以下の通りである。

利用者が目的に応じた Web サービスを UDDI ディレクトリから検索する

Web サービス提供サイトで公開されている Web サービスの情報を WSDL から取得する。

通信プロトコルは、SOAP を用いる。

SOAP は XML で要求を受け、XML で応答を返す。

標準仕様の XML による通信なので、プラットフォームや言語に依存しない。

ここで、旅行予約を例に Web サービスと従来の Web サイトによるシステムの違いを比較する(図 1 - 8)。従来、申込者は旅行代理店に電話するか航空会社、ホテルなどの Web サイトから予約していた。Web サービスを利用すると、航空会社やホテルなどの予約用 Web サービスを利用し、これらを統合した予約システムを構築可能である。このような予約システムを旅行代理店の Web サイトに設置すれば、申込者はこのサイトより様々な旅行プランの立案から予約まで可能となる。これによって非常に利便性の高いシステムとなる。

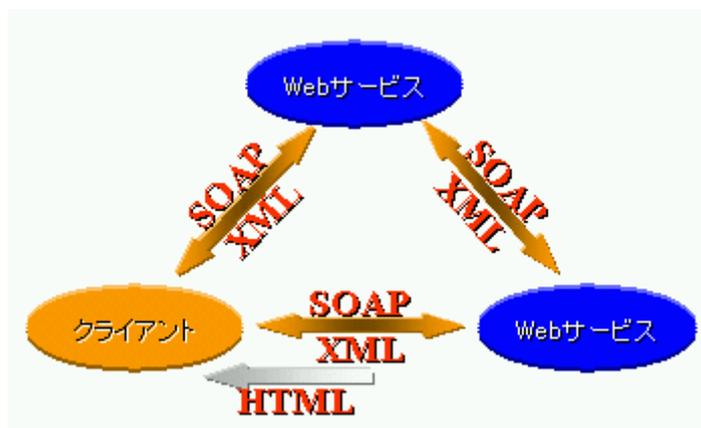


図 1 - 6 Web サービスにおけるメッセージのやり取り

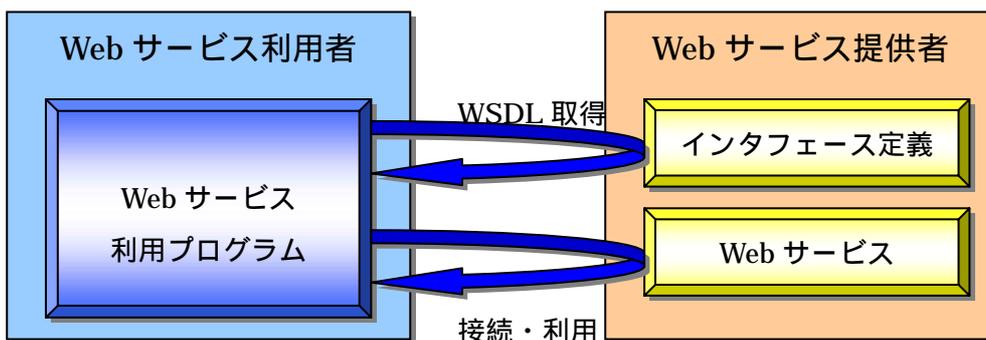


図 1 - 7 Web サービスの構成図

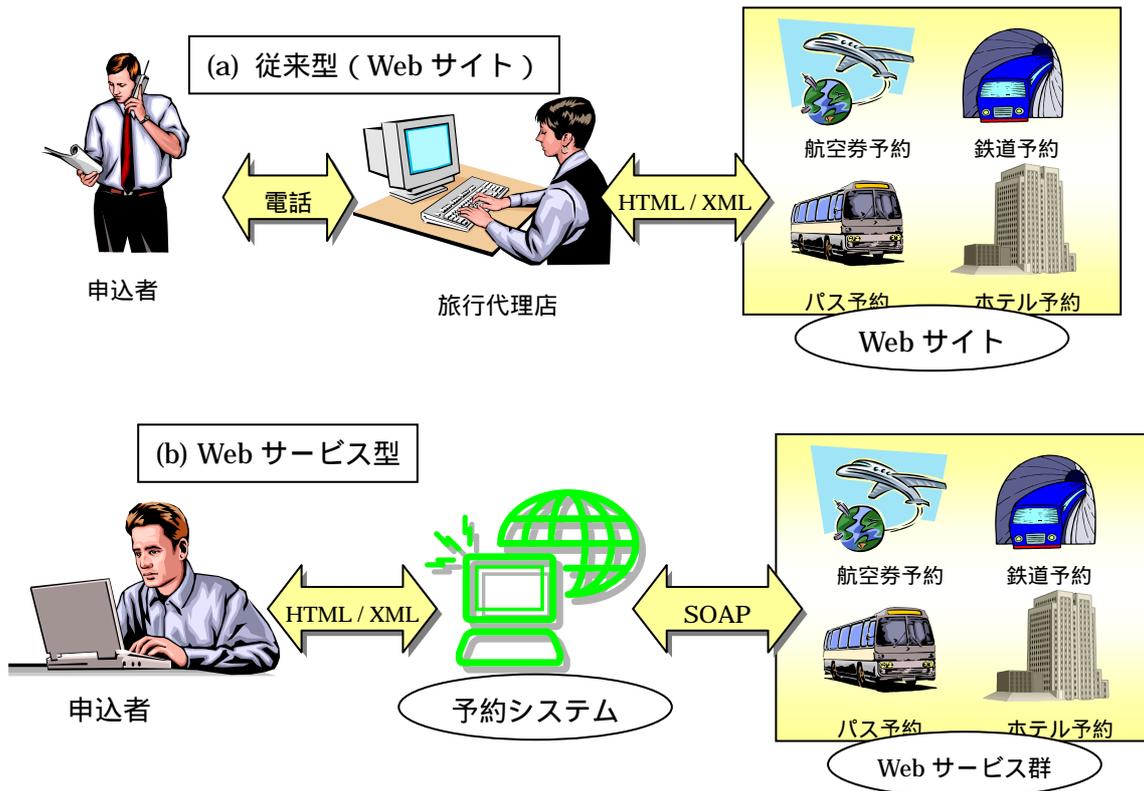


図 1 - 8 従来型システムと Web サービスによるシステムの比較

(3) SOAP の概要

SOAP は XML によるメッセージ交換プロトコルである。元は Simple Object Access Protocol の略称であったが、現在は固有名詞として扱われている。インターネットに接続したソフトウェア同士が相互にアクセスするためことを目的として作成された。通常は HTTP と組み合わせて使用するが、他のプロトコル、例えば SMTP などでも利用できるようになっている。

プログラム同士でメッセージを交換する場合、XML のデータだけでは不十分であるため、SOAP には XML データにエンベロープと呼ばれる付加情報を追加している。SOAP と送信用のプロトコルと組み合わせる部分はプロトコルバインディングヘッダと呼ばれる(図 1 - 9)。SOAP エンベロープが SOAP 本体であり、SOAP ヘッダは SOAP を処理するソフト用の情報となる。送りたいデータは SOAP メッセージ本体部に記述されている。



図 1 - 9 SOAP のデータ構造

(4) UDDI の概要

UDDI は Web サービスを公開し、発見させるためのディレクトリサービスである。Web サイトで例えると、Yahoo や Google などの検索サイトであり、検索という意味で同様な役割がある。

Web サービスに関する情報は、UDDI レジストリ (UDDI registry) と呼ばれる場所に保管されている。UDDI レジストリにあるサービスを検索する場合は、UDDI の照会 API (Inquiry API) を使用する。UDDI の API は SOAP メッセージの形式であるため、UDDI レジストリ自体も Web サービスとして提供していることになる。

(5) WSDL の概要

WSDL は Web サービスのインタフェース部分を記述するための言語である。Web サービスを利用するための仕様を XML 形式で記述する。WSDL では SOAP でのメッセージ交換情報を抽象的な部分と具体的な部分に分けて定義する。これによって定義内容を再利用しやすくしている。

UDDI によって発見した Web サービスのインタフェースを WSDL で定義することによって Web サービスは使用可能となる。表 1 - 3 には WSDL の構造とその説明を示した。

表 1 - 3 WSDL の構造

| 定義 | 説明 |
|------------------------|--|
| タイプ (types) | 交換されるメッセージのデータ型の定義 |
| メッセージ (message) | 伝送するメッセージを定義する。メッセージは論理的なパートで構成され、そのそれぞれが何らかの型システムの定義に関連付けられる。 |
| オペレーション (operation) | 操作を定義する。それぞれの操作は、入力メッセージと出力メッセージを参照する。 |
| ポート タイプ (portType) | 操作の組み合わせを定義する |
| バインディング (binding) | 操作と具体的なプログラムとの結びつきを定義する。 |
| ポート (port) | 通信の端点のアドレスを定義する。通信端点のアドレスとは、実際に Web サービスを提供するサーバの URLなどを指す。 |
| サービス (service) | サービスの所在地などをまとめた情報。 |

1.4 XML を用いた開発事例

1.4.1 XVL

(1) XVL 誕生の背景

21 世紀の製造業では、製品品質向上やコスト削減を目指して設計の 3 次元化 (CAD / CAM / CAE の導入) が行われている。また、インターネット網の整備などの IT 革命は、拠点間の距離といった制約無くし、様々な場面での「情報」の共有化を促した。

『情報』を意味する英語には「Information」と「Intelligence」の 2 種類がある。

- Information ... 断片的な只の情報
- Intelligence ... 意味を持った情報の結合

設計の三次元化と IT 革命によるインターネットの普及は膨大な “Information” の蓄積をもたらした。しかし、あまりにも多くの情報がありすぎて、必要な人が必要な情報を取り出すことが難しくなっている。また、もし取り出せたとしても、情報が複雑化し分かりにくい。これらを解決するには、information を整理・結合し知識 (=Intelligence) に変えていき、情報をビジュアル化し分かりやすく表示することが必須である。

現在の 3 次元 (3D) データは、設計に 3D - CAD が利用されるに伴い増加して設計に関しては有効に活用されてきている。ただし、この 3D データが利用されているのは、大部分は設計部門にのみに限定されている。これが設計・製造現場でのみ 3D データを活用する「エンジニアリング 3D」である。

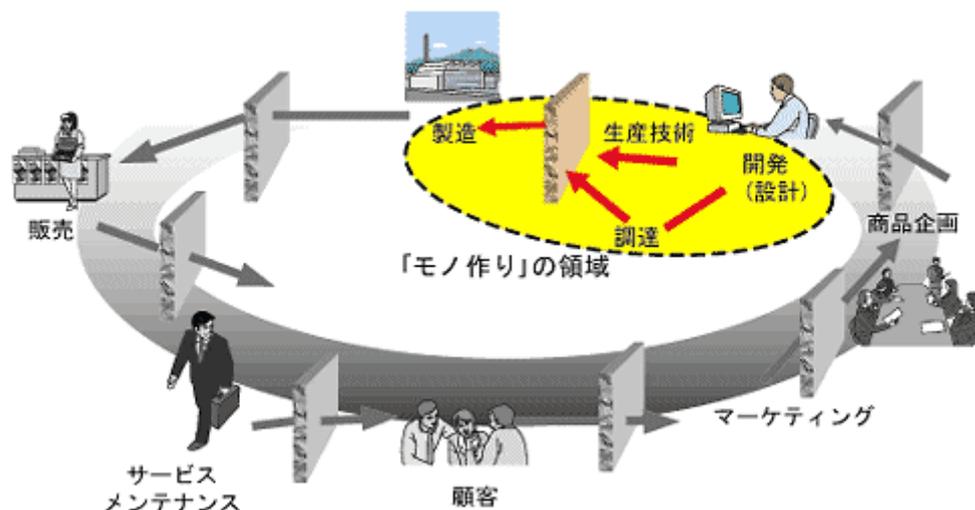


図 1 - 10 エンジニアリング 3D

この限られた分野だけでしか 3D データが活用されていないという原因は以下のような要因に起因する 경우가一般的である。

- PC の高性能化に伴って設計者が作成した 3D データは“巨大化”し、全社で利用するような幅広いネットワーク網で利用が不可能である。
- 設計という高度なタスクを行うため、“高価な”ハードウェアを必要とする。また、CAD の操作方法が“難解”なために、一般の PC ユーザが手軽に利用できない。
- 部門ごとの特性などにより異なる“多様な”CAD が導入され一般的に CAD ごとの専用ツールが必要である。

上記のような要因により、3D データは、組織間の壁を越えられず、設計部門のみなど限定された場所でのみ増加しているのが現状である。

XVL は、インターネットでの利用が可能になるほどの、従来にはない桁違いに軽いデータサイズで 3 次元の形状を表現することが可能です。また、特定のハードウェアに依存せず、様々な CAD・CG ソフトへのインタフェースを持つ XVL 製品群を利用することにより、今まで各社のオリジナルフォーマットでは実現できなかった、異なるソフトウェアで作成したデータを同じ利用環境で活用することが可能になり、今までの壁を取り去ることができます。こうして、設計部門で生み出されたエンジニアリング 3D のデータを、製造現場だけではなく、工場・保守部門・営業部門などでデザインレビュー・ペーパーレス 3D マニュアルに展開することで、データを有効に利用することが可能になります。全社での利用はもちろん、関連会社・一般ユーザまでを巻き込むことによって今までのプロセスを大幅に改善する、これが XVL を利用した 3D データの共有すな

わち「カジュアル3D」の世界である。

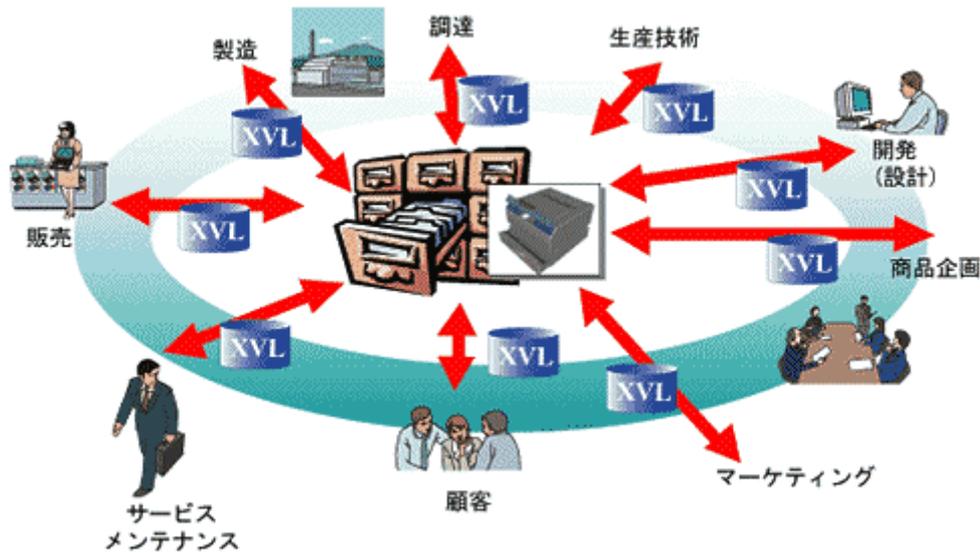


図 1 - 1 1 カジュアル 3D

前述の通り、今までの CAD や CAM の生成するスタンドアロンな環境にのみ蓄積されてきた設計製造用の 3D レガシーデータを XVL 化することで、インターネット網などを利用して流通を可能にする。また、PDM などを用いて整理することで、広くコミュニケーションの手段とすることで、“情報 (= Information)” を分かりやすくまとめ “知識化 (= Intelligence)” を行うことが可能になる。

(2) XVL 技術とは

XVL (eXtensible Virtual world description Language) とは、データサイズが大きくなりがちな 3D データを最大数百分の一以下に軽量化し、インターネット環境で活用することを目的として開発された。ここでは、XVL を構成するテクノロジーについて解説する。

(a) 精度と軽量化

XVL は、慶応義塾大学 環境情報学部 千代倉弘明 教授(現ラティス・テクノロジー株式会社 会長)の考案した、単純な形状から滑らかな曲面を表現するというラティス構造を利用した形状表現をベースとしている。ラティス構造とは、多数のポリゴンから構成されるポリゴンメッシュに比べ、大まかな形状をラティス格子として定義し、Gregory パッチ(曲面間の接平面連続 [G-1 連続]の表現に優れた曲面表現)と呼ばれる自由曲面を内挿することによって、滑らかな曲面表現を行う手法である。滑らかな曲面形状を高精度のままポリゴンメッシュで表現する場合、非常に多くのポリゴンを必要とするためデータサイズが膨大になるが、ラティス構造では、曲面データを

ラティス構造として単純化して扱うため、データはとてもコンパクトになる。また、制御格子を操作することにより、自由曲面を変形させることも出来るため、モデリング手法としても利用され、XVL モデリングツールである XVL Designer として製品化されている。また、形状を単純化して扱っているために大量のポリゴンを操作するモデラーでは実現できなかった、直感的な形状操作が可能になっている。

- 既存データからのデータ軽量化

IGES、Parasolid などの一般的な CAD で用いられる中間ファイルや CATIA、I-DEAS などの 3D - CAD データや、また、従来インターネットでの 3D データ利用を想定して開発された VRML などは、マシンの高スペック化、ソフトウェアの高機能化に伴い数百メガバイトにもなってしまうことが日常的である。この巨大なデータサイズの 3D データについてネットワーク環境を利用して活用するには、大幅な軽量化が必要である。XVL ではどのような手法を用いて、3D データの軽量化が行われているか解説する。

- CAD XVL について

多くの汎用的な 3D データ表示ソフトウェアでは、CAD データであっても、VRML や STL などのポリゴン化したデータ形式に変換して活用されているのが一般的である。このような曲面からポリゴンへの変換を行うと、精度を保つためにはデータ容量が大きくなってしまい、逆にデータサイズを小さくすると精度が損なわれてしまう。

しかし、XVL の CAD データの軽量化の手法は、精密な製造データをより多くのシチュエーションで活用するといった目的での開発が行われているため、曲面のまま軽量化するという、まったく違ったアプローチ方法で軽量化されている。通常の CAD モデルは 10 - 5 ~ 10 - 6mm という精度をもっているが、設計ではなく、ビューイングを目的とした場合これほどの厳密な数値は必要ではない。XVL では曲線・曲面の次数の引下げや、不要な制御点の削減を行う、それに加えトリム曲面をラティス構造の内挿曲面へと変換することによって、大幅な軽量化を実現している。

このように非常に CAD データに近い曲面形式のデータでありながら 10 - 1 ~ 10 - 3mm という高い精度をもったまま軽量化を行うことで、3D データを設計だけではなく、製造・検査など幅広いフィールドでの 3D データの利用を可能にしている。

- ポリゴン XVL について

CG ソフトをはじめとするポリゴン系 3D データは、映画やテレビ放映用に作成されるなど、ポリゴンデータとして市場に多く存在している。また、3D スキャナなどの点群から作成される、ポリゴンデータは非常に緻密に作成されているため非常に大きなサイズになってしまい、一般的なハードウェアでは取り扱うことさえ困難な場合さえある。一般的にはポリゴンデータの軽量化はポリゴンを間引くことによって軽量化するポリゴンリダクションであるが、軽量化するほど元データの原型が崩れ精度を失ってしまう。また、表示処理に関してもポリゴンデータでは、隣り合

うポリゴンの法線ベクトルを平均化することで、スムーズなシェーディングを行い、擬似的に滑らかに表示しているがXVLでは、ラティス構造を利用して曲面を表現しているため、データサイズが小さいながらも滑らかな表示を行うことが可能になっている。また、点群測定器等で生成した、巨大なポリゴンデータであっても"ポリゴン格子変換"を行うことでラティス構造に変換することが可能である。こうして作成されたXVLデータは、IGES形式などに変換することで、設計を行う参考形状として、CADソフトウェアにインポートするなどの利用も可能になり、大きなメリットが生まれる。

(b) 合理性

3Dデータを生産設計、製造、保守、購買、営業などで活用することにより、文字情報や平面的な2Dの図面・画像だけでは伝えにくかった形状の情報伝えることが容易になる。また、製品を購入する顧客に対してはカタログやマニュアルの中で3Dを利用して、情報を提供することは平面では伝わらない奥行きや製品の特徴を効率的に伝えることを可能にする。しかし、今まで3Dデータを設計以外の様々なシーンで利用する場合にはCGソフトなどで、新たに作成することが主流であった。これでは、せっかく3D-CADを利用して3Dデータを作成しているにもかかわらず、CGソフトでまた3Dデータを作成するということは二重の手間が発生する。3Dデータが流量できずさらに多額のコストが発生してしまう事が、情報伝達に3Dデータを用いることが効果的であることは分かっているにもかかわらず3Dデータの利用が広まらない原因となっている。XVLでは、3D-CADデータやCGソフトで既に作成されている3Dデータがあれば、あらかじめアプリケーションに組み込まれたI/OやXVL Converter製品群などのツールを利用することで容易にXVL化することができる。今まで蓄積された資産である3Dデータをインターネット上での利用可能な形式に変換する。これが"Web 3Dエンコーディング"である。XVLを変換することで、設計のみに使用されていた3D-CADデータを、設計分野でのコラボレーション作業はもちろんのこと、製造、保守、購買、営業などの製品ライフサイクル全般におよぶ企業全体の幅広いフィールドで3Dデータを活用が可能になる。

(c) 表現力

● アニメーション

XVLは形状表現だけではなく、形状に対して組立の工程や、製品の動作などのアニメーションをXVLファイルの一部として定義でき、さらに分かり易い情報の提供を可能とする。これを利用して、インターネットの双方向性を利用しムービーなどにはないインタラクティブなコンテンツを提供することも可能である。また、XVLは、ファイル内部に定義するアニメーションだけではなく、Webの世界で一般的に使用されているJavaScript等を利用して、柔軟なアニメーション定義を行うことも可能である。

スクリプトによるアニメーション定義は、現在Windows版XVL Player for Internet Explorer

のみサポート。

- Web ドキュメントリンク機能

形状を構成する立体・面・稜線・頂点などから、Web 上のあらゆるデータ（2D イメージ、図面情報、HTML ドキュメント等）へのリンクを設定することができる。3D データをインデックスとして、様々なデータをネットワーク上で連携することが可能になる。

- 3D 寸法注記機能

XVL の一部として 3D 注記や 3D 寸法表示、交差情報を書き込むことができる。3D - CAD の持つ情報をそのまま XVL に変換すれば、設計だけではなく下流工程でのモデルの情報を利用が可能になる。XVL の編集ソフトを利用すれば、新たに寸法情報や注記を追加・編集することも可能であり、形状だけでは不足している情報を付加することが可能である。

- 表示属性

情報伝達において、視覚情報は重要なポイントをしめる。XVL はマシンスペックに応じて曲面の分割数を切り替え、表示クオリティを選択できる機能やマテリアル（形状の色）を設定し特定の部分のみ強調するといったような表示が可能であり、テクスチャのアンチエイリアシング機能や環境マッピング機能を利用して表示がよりリアルなリアルタイムレンダリングを行うことも可能である。さらに、多彩なテクスチャ表現を可能にするアルファブレンディングや、操作時に半透明色を簡易的に表示する機能など強化が行われている。

- 構成情報

XVL は、単一形状の表現だけではなく、CAD と同じように形状（部品）の位置関係やアセンブリ構造を XVL データ内に定義することが可能である。このデータを利用して部品表を手軽に作成することができる。また、部品の形状データとアセンブリ構造のデータを別々のファイルとしても持つことができ、同一部品を配置する場合には、インスタンス（仮想部品）として形状を定義することで、より軽量化して形状を扱うことができる。

（3）XVL の標準化

インターネットにかかわる規格を決定する国際規格団体の W3C ワーキンググループの Web3D コンソーシアムでは、VRML に変わる次世代の Web での利用を目的とした 3D 規格として、X3D（eXtensible 3D）の策定が行われている。この X3D においても XML ベースという拡張性は考慮されているが、データ構造としては、VRML 同様、ポリゴン構造を踏襲した物がベースとなっている。これは、X3D でも 軽量であるということと精度がよい表現という問題は解決していないということの意味している。VRML は 1997 年に国際標準規格としては、ISO に認定

されたが、残念ながら、実用レベルでの利用はごく限られた分野でしか利用されておらず、XVLが目指す、「誰でも気軽にその場で活用できる環境を構築」するには至っていない。投票で決まったVRMLの仕様が失敗してしまったWeb3Dコンソーシアム(旧VRMLコンソーシアム)では、新たなフォーマットを開発するのではなく、市場のデファクトスタンダード(=市場で多く利用されている)技術を採用することにより、標準化しようとしている。

市場を急速に広げ普及が進んだXVL技術は、先の問題点ははじめとする様々な基本セットだけでは実現不可能な問題点を解決するX3D拡張形式のひとつとして有力候補となっており、ラティス社では、X3Dに不足している曲面表示の拡張形式として提唱しており、その仕様が以下のサイトに公開されている。

(参考)XVL、VRML、IGESとの比較

なぜXVLが今まで出来なかったような改善が出来るのか、現在、市場で利用されている一般的な3Dデータ(IGES、VRML)と比較する。

● IGES (Initial Graphics Exchange Specification)

一般的に設計製造用の汎用3Dデータとして利用されている。多くのCADで標準的に入出力が可能であり、主に異種CAD間のデータ交換用データとして利用されているため、曲面構造を厳密に表現できるという特徴がある。しかし、IGESは汎用に利用する事を重視したために、データの冗長性が大きく、データ量が大きいという問題点がある。またこの冗長性が異種のシステムからのデータを受け取ったときなどに、“面落ち”や“データ欠落”といったCADごとに方言(癖)を持つという欠点にもなる。

また、データ配信、ビューイングを目的としたフォーマットではないために全社で利用するといったような利用には適しているとは言えない。

● VRML (Virtual Reality Markup Language)

VRMLは主にWebでの三次元形状の表示に特化した、ファイルフォーマットである。基本的にはポリゴンという三角形や四角形の平面の集まりによって形状を表現している。このフォーマットもまた、多くのCADやCGソフトから出力することが可能だが、ポリゴンで形状を近似して表現しているため、精度を要求される製造分野での利用は難しく、また見ることに特化したe-Commerceなどの利用を想定して、詳細なモデルを表現しようとする、大量のポリゴンが必要となりデータ量が膨大になってしまう。エンジニアリング用途かつITでの利用に必須とされる「軽量」と「精度」という矛盾する条件は、今までのフォーマットで同時に満足していなければ、活用範囲は限られてしまう。

表 1 - 4 XVL、IGES、VRML 比較

| 種別 | XVL | IGES | VRML |
|----------------|---------------------------------------|---------------|---------------------------------------|
| 開発元 | ラティス・テクノロジー | ANSI / ISO 標準 | ISO 標準 |
| ターゲット市場 | カジュアル 3D 市場 | 設計製造データ交換 | Web3D |
| データ構造 | サーフェス、ソリッド | サーフェス、ソリッド | ポリゴン |
| データサイズ | | | |
| (転送速度) | | × | |
| 既存データ変換 | | | |
| XML ベース | | × | × |
| ソフトウェア開発環境 | | | |
| オーサリングツール | | | |
| 表示品質 | | | |
| アニメーション再生 | | × | |
| ブラウザでの表示 | (Internet Explorer, Netscape 対応) | × | (Internet Explorer, Netscape 対応) |
| CAE / CAM 利用 | | | |
| DTP (イラスト制作利用) | | | |

第2章 製造業におけるXML情報連携実証モデル技術に関する調査研究

本章は、FAオープン推進協議会（FAOP）/XML情報連携実証モデル専門委員会（FAOP-XML）の協力の下、同委員会が進めている製造システムにおけるXML技術を活用したオープンな情報連携仕様の内容および取り組みについてまとめたものである。

製造業におけるXML情報連携実証モデル技術に関する調査研究に関し全体としての方向付けを行い、3つの検討課題に対し開発期間を3つに分け各フェーズの終了時点における目標とする成果をまとめた。

表2-1 XML情報連携実証モデル技術に関する調査研究

| 検討課題 | フェーズ1 (2004/03終了) | フェーズ2 (2004/10終了予定) | フェーズ3(最終) |
|------|--|---|----------------------------------|
| D1 | MES領域サブシステム連携 インタフェース仕様 MES装置API仕様 | 管理仕様 | ISO15745 プロセスプロファイル 提案まとめ |
| D2 | 既存ミドルウェア調査 (XML化など) 新ゲートウェイアーキテク チャ仕様 | ゲートウェイコンセプトモデル、 エンジニアリングモデル、 ME装置APIゲートウェイマ ッピング仕様 | ISO20242へのFAミドルウェア としての逆提案まとめ |
| D3 | 日産オブジェクトモデル エンジニアリング仕様 | 装置コンセプトモデル | ISO15745 装置プロファイル提 案、まとめ |
| ツール | D1,D2,D3 アウトプットベ ースツール仕様 | ツール実装 | 関連モデル間変換ルール、手法と して上記標準化提案に包含 |

(1) 全体方針および検討課題：D1

- FAシステムにおけるレイヤ間、レイヤ内サブシステムの情報連携を実現するためには、今後のFAにおける様々なビジネスモデル対応のコンセプト（概念データ機能）モデルをジェネリックに構築する。（ホワイトペーパー[別添]記載のメンバーの関与するビジネスモデルをベースとして）構築されたコンセプトモデルは、ビジネス対応サブシステム情報連携で実現を期待されるFAシステムの要求仕様となる。コンセプトモデルはサブシステム間コラボレーションモデルによりサブシステム間インタラクション仕様として補強される。
- 上記ジェネリックな情報連携の概念モデルをもとに、スペシフィックな分野対応のオブジェクトモデルに特化する。
- 分野特化されたオブジェクトモデルは同分野の特定のアプリケーション対応エンジニアリングモデルにマッピングされる。同エンジニアリングモデルではコンセプトモデルとコラボレーションモデルで示されたインタラクションの実際が分散実装環境、そのネットワーク、プロトコルとして示される。

- 上記エンジニアリングモデルには、既存のサブシステム、デファクトコンソーシアム仕様をサブシステムとして包含する利用モデルが何らかの変換ゲートウェイモデルなどとともに示される。
- D1 は MES エリアのサブシステム連携について、MESX の活動成果を活用するものとし、MESX 活動と D1 活動は一体で進める。具体的には現在 MESX でまとめられているレベル 3 を中心としたユースケース、及びそれより抽出した MESX オブジェクトクラスモデルを上記コンセプトモデルとして位置づけられる方向。

(2) 検討課題：D2

- FA におけるロボット搬送機、NC 機械、検査装置、各種 PLC などの生産設備を統一上位アプリケーションとして提供する目的で既存の様々なミドルウェア提案が種々あるが、これらのミドルウェア間の情報連携を如何に達成するかという問題がある。
- 上記ミドルウェアの提供する API は対象とする装置対応インタフェース意味定義ではなく、それぞれのサブシステムに使用されているロボットコントローラ、NC、PC などのコントローラとしてのインタフェース意味定義に過ぎない。このため、MES アプリケーションよりの具体的な装置対応意味づけされた情報連携要求に対し対応できていない。
- コントローラプロファイルから、装置・設備プロファイルとしての意味定義を規定し、API として提供できる方向で検討する。
- 上記内容を既存のテクノロジ+（ゲートウェイ機能）として提供できるエンジニアリングモデルの提供を検討する。
- 当面、業界別モデルの調査と検討、各デファクトミドルウェアモデルの基本アーキテクチャに沿った XML 表現、基本アーキテクチャの妥当性検討を実施する。

(3) 検討課題：D3

- ジェネリックなコンセプトモデル構築を念頭において、そのサブシステムに位置づけされる任意分野対応の装置・設備オブジェクトモデルをボトムアップアプローチで構築する。
- 対象装置としては NC ロケータ（数軸の位置決めよりなる簡単なもの）を対象例として取り組む。
- 目標とするモデルとしては装置・設備オブジェクトモデル（設計への要求仕様を表す装置のコンセプトモデル）と特定システムとしてのエンジニアリングモデル（設計されたパラメータインスタンス、オペレーション対応メソッドをインスタンスとして有する、設計後のモデル）
- また、フィールドネットワークプロファイルとして ISO 15745 対応 CC-Link プロファイルを例として、装置、設備プロファイルへのサブシステムプロファイルとしての応用、（統合組み込み手法）について検討してみたい。（上記 CC-Link プロファイルは日本より ISO

/TC184/SC5/WG5へ新提案される予定である)同CC-Linkプロファイルには、設備診断など、本プロジェクトで議論されている実装設備アドレス情報インスタンスの格納領域のアサインなど先取りの工夫がなされている。

(4) 統合課題：ツール

- 検討課題D1～D3の活動に平行してワークし、各課題へ適用可能なツールを提案し、同要求仕様を策定する。ツールの実装に関してはD1～D3のエンジニアリングモデル検討、実装フェーズにあわせて来年度以降の活動として別途検討する。
- ホワイトペーパーに記載の表1「要求されるテクニカルタームに対応する技術内容」機能をツールに展開する。優先順位や、ツールとしての一体化、フレームワークとしてのまとめなども考慮する。

2.2 情報連携モデル案

2.2.1 概要

FAOP-XML 情報連携モデルの概要を図 2 - 1 に示す。FAOP-XML では製造システムを計画系、MES、装置の3階層に分け、各階層のエンジニアリングモデルと標準実行モデルを規定する。エンジニアリングモデルによって、エンジニアリング情報の交換を実現し、標準実行モデルによってプラットフォームに依存しない実行モデルを規定し、異なるプラットフォームを含んだ製造システムのエンジニアリング情報交換を実現する。

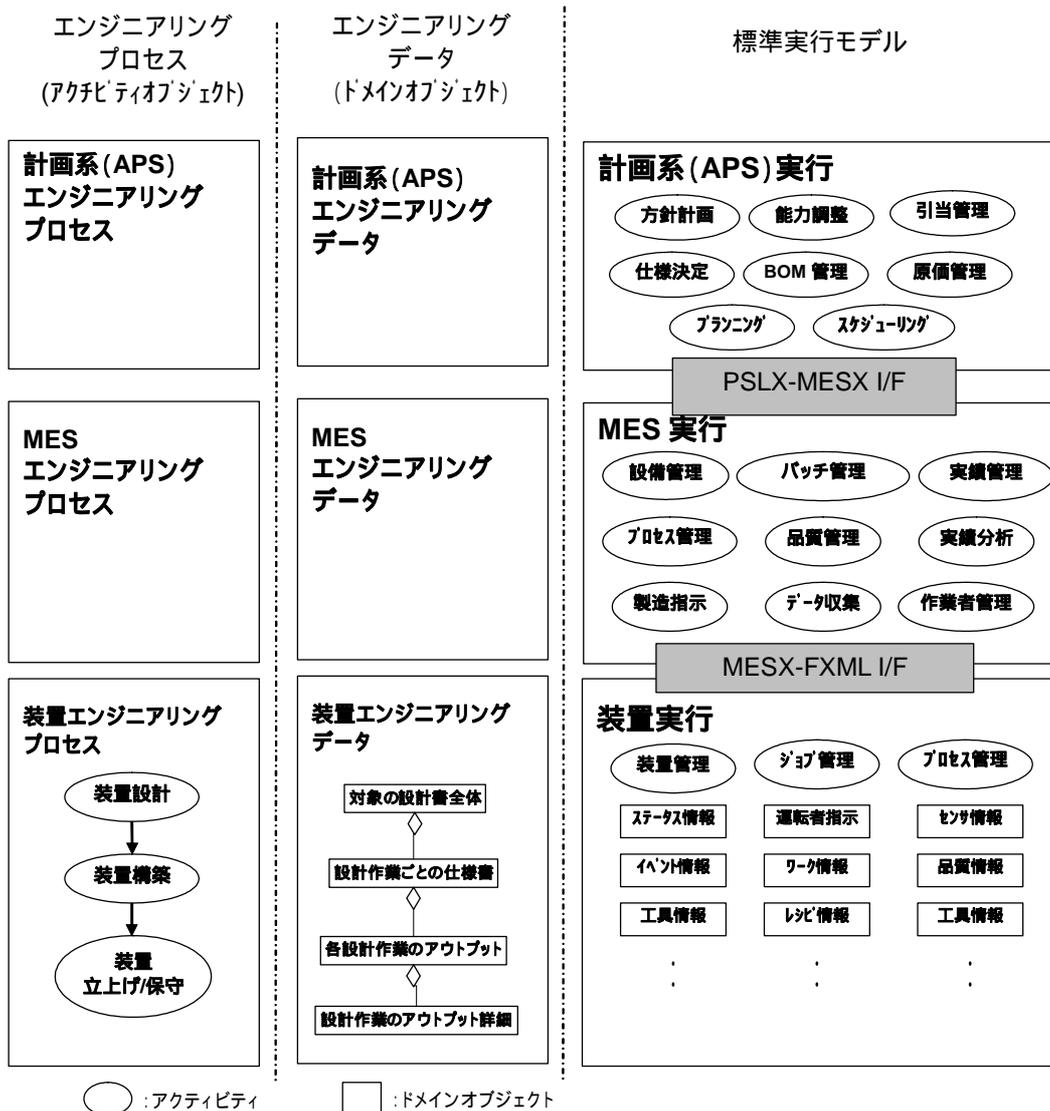


図 2 - 1 FAOP-XML モデル概要

(1) エンジニアリングモデル

エンジニアリングプロセスと各エンジニアリング工程で利用/作成するエンジニアリングデータのモデルを規定する。エンジニアリングモデルを規定することで、エンジニアリング情報の交換、検索、流用を実現する。

(2) 標準実行モデル

プラットフォームに依存しない実行モデルを規定する。各プラットフォームの固有モデルを標準実行モデルにマッピングすることで、異なるプラットフォーム間の情報連携を記述する。また、プラットフォームに依存しない論理的な連携モデルを構築する。

(3) ジェネリックモデル

FAOP-XML 情報連携モデルでは、連携対象をアクティビティとして定義し、連携データをドメインオブジェクトに定義する。情報連携は、アクティビティ間およびアクティビティとドメインオブジェクト間のコラボレーションで記述される。情報連携を直接アクティビティ間のメッセージだけで定義しないでドメインオブジェクトを導入した理由は、実装に依存しない連携データ定義を行うためである。一般にメッセージの転送サイズはプラットフォームや要求性能を実現する実装方式に依存するため、メッセージ定義はプラットフォーム依存になる。プラットフォームに依存しない情報連携を規定するためにアクティビティ-ドメインオブジェクトモデルを導入する。

ドメインオブジェクトの実際の実装は、単一あるいは複数のメッセージやファイル転送でもよいし、共有ファイルアクセスやデータベースを用いてもよい。結果的にドメインオブジェクトに相当する連携データが交換できれば、アクティビティ間は連携することができる。このモデルによって異なるプラットフォーム間においても論理的な連携モデルを構築できる。

本節ではアクティビティ-ドメインオブジェクトを用いた連携モデルの共通スキームを定義するジェネリックモデルについて説明する。

(4) アクティビティの説明

アクティビティとは、情報連携する対象を表す。エンジニアリングの情報連携では、各種エンジニアリングソフトウェアやそれを実行する作業者が対応する。実行時の情報連携では、MESアプリケーションや装置機能の実行対象に当る。

(a) 装置エンジニアリングモデルの例

- 装置設計（設計ソフトウェア）
- 装置構築（構築ソフトウェア）
- 装置立上げ（インストールソフトウェア）
- 装置保守（保守ソフトウェア）

(b) MES 実行モデルの例

- 設備管理
- バッチ管理
- 実績管理
- プロセス管理
- 品質管理
- 実績分析
- 製造指示

- データ収集
- 作業管理

(c) 装置実行モデルの例

- 装置管理
- ジョブ管理
- プロセス管理

2.2.2 ドメインオブジェクトの説明

ドメインオブジェクトは対象や分野ごとに定義された、ある程度まとまった目的で利用される受動的なデータの集まりを示す。設計プロセスでは仕様や設計データ、実行モデルでは交換データ等を示す。ドメインオブジェクト自体も階層的に構造化でき、個々の要素アクセスやブロックの一括転送、装置情報全部のバックアップなど粒度をフレキシブルに取れる。

ドメインオブジェクトによる情報連携はアクティビティのコラボレーションモデルで記述される。

(1) 装置エンジニアリングデータのドメインオブジェクトモデル

図 2 - 2 にエンジニアリングデータのドメインモデル構造を UML クラス図で示す。第 1 階層は対象の設計書全体に対応し、第 2 階層は設計作業ごとの仕様書に対応し、第 3 階層は各設計作業のアウトプットに対応し、第 4 階層以下は各アウトプットの詳細に対応するように、装置の設計情報を共通データモデルで記述する。図 2 - 3 に設計情報のドメインモデル例を示す。

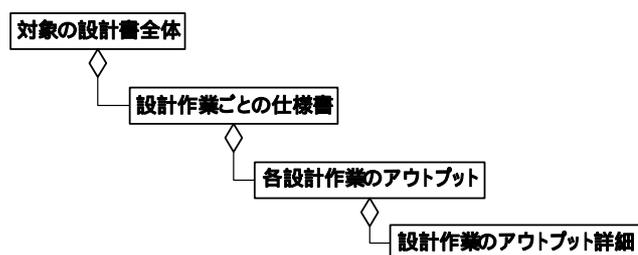


図 2 - 2 エンジニアリングデータのドメインオブジェクト構造例

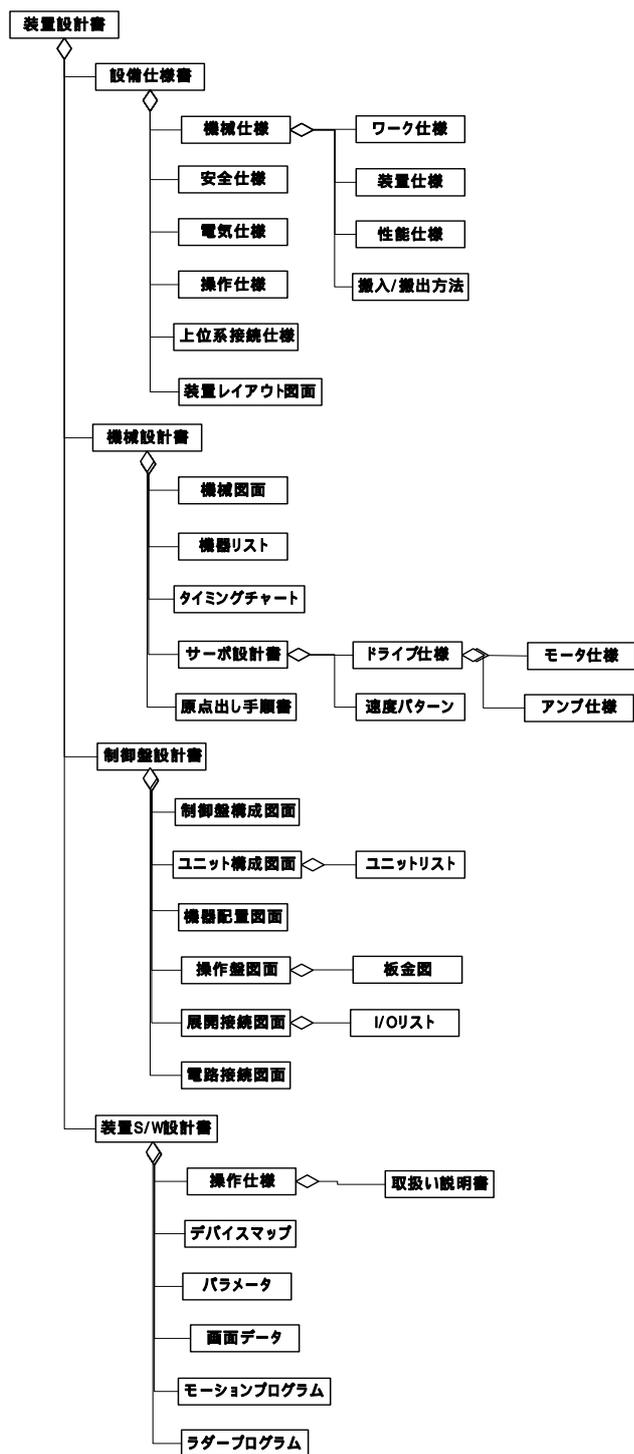


図 2 - 3 装置エンジニアリングデータのドメインオブジェクトモデル例

(2) コラボレーションモデル

情報連携を記述するために、ドメインを扱うアクティビティのコラボレーションモデルを記述する。図 2 - 4はアクティビティ 2 に関連するクラス図の例を図 2 - 5にはアクティビティ 2

のコラボレーションを示す。この例では、アクティビティは設計工程に対応し、ドメインは設計データ等に対応する。

アクティビティ1が装置XのドメインAデータを生成し、アクティビティ2にイベントを送る。イベントを受けたアクティビティ2はドメインAデータを読み出し、ドメインBとドメインCのテンプレートを読み出し、ドメインAデータを基に設計した情報をドメインBとドメインCに書き出す。設計完了後にアクティビティ3にイベントを送る。イベントを受けたアクティビティ3は装置XのドメインCデータを読み出し、処理を実行する。

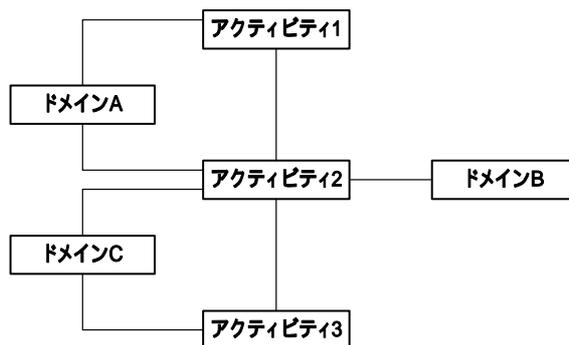


図 2 - 4 アクティビティ2に関連するオブジェクトのクラス図例

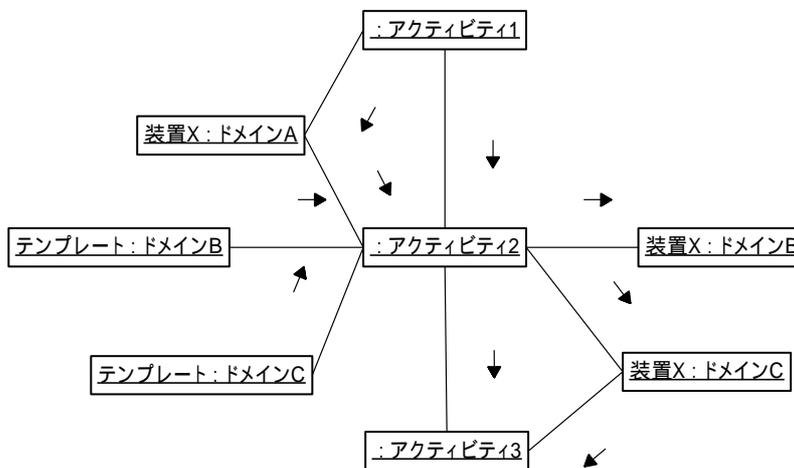


図 2 - 5 アクティビティ2のコラボレーションモデル例

2.2.3 MES アプリケーション連携モデル

MES アプリケーション連携モデルとしては、図 2 - 6 の Interface-B にあたる計画系と MES (レベル4 とレベル3) との間での連携、Interface-C にあたる MES (レベル3) 内での連携、Interface-A にあたる MES と装置 (レベル3 とレベル0-2) との間での連携を考慮する必要がある

る。

Interface-A に関しては、次項で記述するため、ここでは計画系-MES 間の連携モデル、および、MES-MES 間の連携モデルについての方法論について述べる。

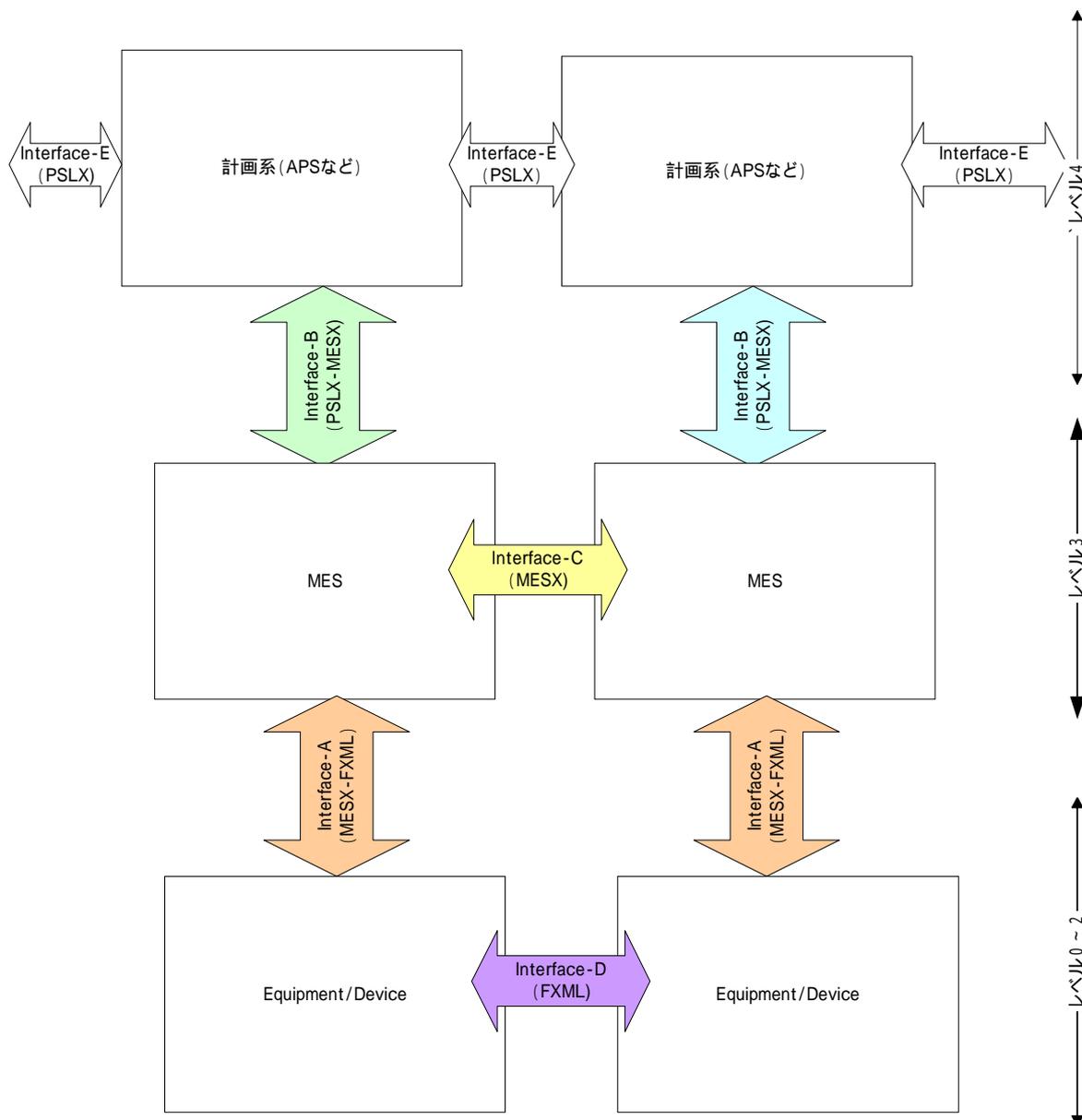


図 2 - 6 FAOP-XML 基本システム

計画系 - MES 間の連携モデルについては、MESX 活動において標準化策定が検討されている。また、MES アプリケーション間の連携モデルについては、FA 分野の国際標準化団体である ISO / TC184 によって、検討されている ISO 16100 (製造ソフトウェア・ケイパビリティ・プロファイリング) で規定されている方法論を分野別に適用する方向で検討されている。

(1) 計画系-MES 間情報連携モデルと方法論

(a) 標準化としての方針

MES インタフェースを検討しているグループ (MESX) では、MESX 標準化の方向付けとして、

- SP95 の階層モデルの採用
- 階層ごとのオブジェクトモデルの記述
- XML によるインタフェースの記述
- 機能モデル

という4つの項目を標準化の方針としている。

(b) インタフェース設計の方針

インタフェース設計のためのモデルとしては、以下のような方針で行っている。

- コミットメントネットワークモデル

レベル間のコミュニケーションを Medina-Mora たち[1]のコミットメントネットワーク (CN) モデルでとらえる。CN モデルは、図 2 - 7 に示すように、Customer (依頼者) と Performer (実行者) の間で交わす作業依頼のモデルである。依頼者は実行者に作業を依頼する前に、それが可能かどうかを打診する。可能であればその依頼を受ける。これは一種の契約である。実行者はその作業を実行し、結果を依頼者に報告する。この間の打診、契約、実行、報告はそれぞれ代理人や下請けの実行者に再依頼することができるが、最終的には、最初の依頼者に報告されなければならない。

レベル内のコミュニケーションは、これよりやや緩い契約関係に基づくと思われるが、インタフェース設計の対象ではないため検討しない。

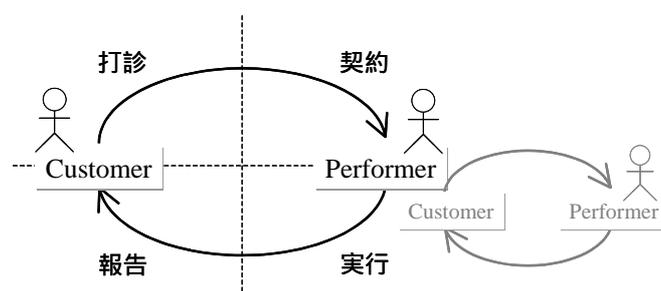


図 2 - 7 コミットメントネットワーク

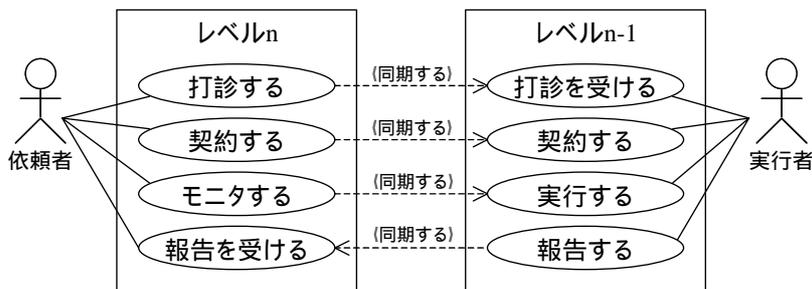


図 2 - 8 コミットメントネットワークのユースケース表現

CNモデルに基づく基本的なユースケースのパターンは図 2 - 8 のようになる。すなわち、依頼側のユースケースとして「打診する」、「契約する」、「(実行を)モニタする」、「報告を受ける」があり、実行側のユースケースとして「打診を受ける」、「契約する」、「実行する」、「報告する」がある。このような委託要求は上位レベルから下位レベルへという方向だけでなく、その逆方向にも発行される。報告の流れは、委託要求の流れと逆になる。

実行者の候補が複数ある場合は、RFP (Request for Proposal) を公開し、入札して決めるプロトコルが知られている (契約ネット) が、分散型の生産では落札基準の設定が難しいので、ここでは考えない。生産管理システムでいうと、実行者、たとえば工程の選定はレベル 4 ですで行われているか、どの工程で実施するかを選定をレベル 4 に依頼するものとする。

● 9 状態のプロトコル

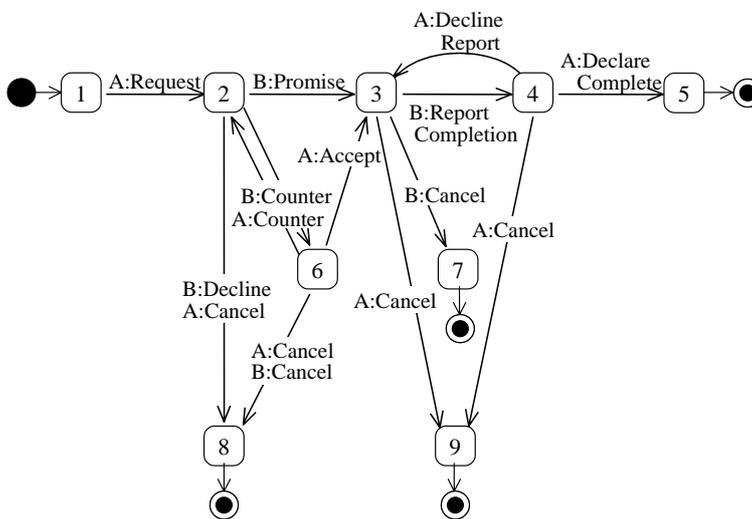


図 2 - 9 コミットメントを伴うインタラクションの 9 状態

こうした委託関係には、実際には打診や契約が不調に終わったり、依頼者が実行途中で依頼をキャンセルする、実行者が実行を中断したり放棄するなどの異常状態もありうる。したがって、委託関係がとりうる状態は図 2 - 9 のようになる[2]。このプロトコルを適用する場合には、作

業の識別子をつけて対話を行う。

詳細のプロトコルを検討する段階では、状態遷移を引き起こすすべてのプロトコルについて規定する必要がある。

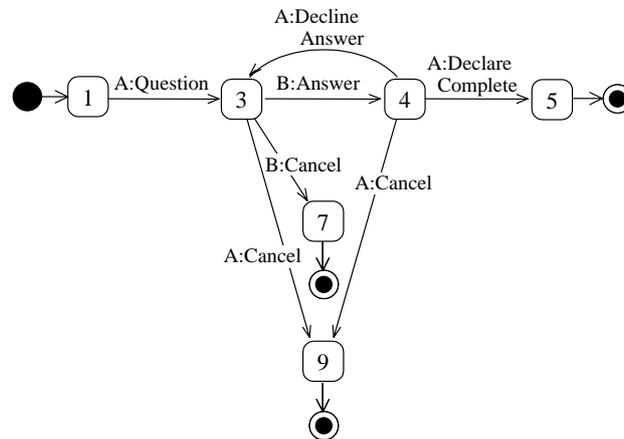


図 2 - 10 コミットメントを伴わないインタラクションの6状態

● 6状態のプロトコル

一方、委託関係でないコミュニケーションもある。それは、情報を問い合わせる場合であって、問い合わせ先が何らかのコミットをすることはない。この状態モデルは、上の9状態の一部を取り出した形になる。

このような問い合わせは、特定の作業識別子をつけて行うものでない点で、上の9状態のプロトコルと区別される。もし、特定の作業について情報を問い合わせるとすれば、それは9状態プロトコル中の「実行をモニタする」ユースケースである。

(c) 具体的な設計方法

上記の方針に基づきデータモデル、および、通信プロトコルの定義は、次にあげる手順を反復的に行って改善を図りながら進めている。

● ユースケースの獲得

MESXに参加しているメンバーが典型的と考える生産管理システムを想定して、計画系(レベル4)、実行系(レベル3)および装置系(レベル0-2)が持つべき機能を列挙し、ユースケースとして記述する。ユースケースとは、システムが備えるべき機能に1つひとつ命名したもので、要求の管理単位である。

ユースケースの記述内容は、ユースケース名、アクタ、目的、この機能の実行のより変化する内容を示す事前状態と事後状態の記述、アクタとシステムの対話、例外条件での振る舞いなどを記述するのが一般的である。この記述を「ユースケース記述」と呼ぶ。列挙したユースケースす

べてについてユースケース記述を作成すると作業量が膨大になるので、一部の主要なものについてのみ作成する。

図 2 - 1 1はその様子を模式化したものである。計画系のユースケース A-2 は実行系のユースケース B-2 との間で、コミットメントネットワークのユースケースの関係にある。たとえば A-2 が「工程ごとに作業実施を指示する」という打診の働きかけであるとする、B-2 は「工程ごとに作業実施を受ける」という打診を受けるユースケースで対応するという具合である。インタフェースの規定に必要なのはこうしたユースケースであり、これについてはユースケース記述を念入りに行う。レベル間での対話を伴わないユースケース、たとえば A-1 はレベルを越えないため、インタフェースを規定する必要がない。

任意に列挙されたユースケースが、生産管理システムのあらゆる機能要求を網羅していることはありえないので、ここから検討するインタフェースは、この列挙された機能の範囲を越えて適用することはできない。規定されたインタフェースに対して一般化し改良を繰り返すことで、適用領域を拡大していく。

● タイプ図の作成

獲得されたユースケースとすでにある知識を基に、実行系、装置系のタイプ図(UMLのクラス図の表記法で記述する)を作成する。ここでいう「タイプ」とは、ISA/SP95でいう「抽象データ」に相当し、実装方法を規定しないクラスであり、データ属性と操作(抽象的なシグニチャ)を持つ。

タイプ図を記述する目的は、問題領域に現れる語彙と概念構造を記述し、それらの名称と意味を規定することにある。レベルごとに表れる語彙とその意味は異なっているのが普通であり、意味論上のギャップがある。そのギャップを正しく埋めるように通信メッセージを設計する。

タイプ図を作成する過程で発見された新しいタイプやユースケースがあれば、これらを付加することを繰り返して、妥当性を高めていく。

計画系のタイプ図は PSLX ですでに規定されているので、これを尊重する。

● インタフェースの設計

列挙されたユースケースのうち、計画系と実行系の間で通信する機能を必要とするもの、実行系と制御系(レベル0-2)との間で通信する機能を取り上げて、その間の通信メッセージを設計する。インタフェースは、層間の通信回数と通信量が最小となるように設計する。

● メッセージの設計

メッセージ形式は PSLX の形式に準拠する(2.3 節を参照のこと)。メッセージの形式は、KQML (Knowledge Query and Manipulation Language) [3]のメッセージ形式を参考にして、命令部(performative:)とデータ部(content:)、制御部(sender、receiver、ontologyなどの指定)からなるものとする。これらを XML でエンコードする。

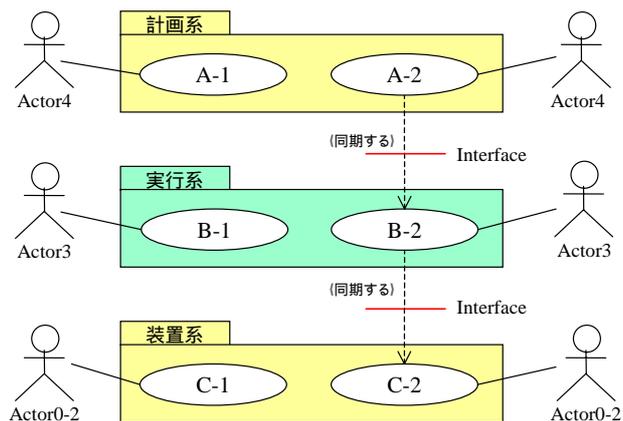


図 2 - 1 1 ユースケース図とインタフェース

上記の手順 において、得られたユースケース一覧を以下に示す。但し、これまでに MESX のメンバーが取り上げたユースケースは 20 数件であり、これで十分とは考えられなかったため、網羅性を高めるために、会合で議論しながら、想定されるユースケースを追加した。追加したユースケースの内容は記述せず、ユースケース名だけを挙げるにとどめた。こうしてユースケースは、最終的に 111 件を得た。

表 2 - 2 ユースケース一覧

| レベル | 番号 | ユースケース名 | ユースケース記述の有無 | |
|-----|--------------|----------------|-------------------|--|
| | | | 有無 | ユースケース名 |
| 5 | 1 | 利益計画を立てる | | |
| | 2 | 長期需給計画を立てる | | |
| | 3 | 資産を管理する | | |
| | 4 | 新製品を企画する | | |
| | 5 | 品質計画を立てる | | |
| | 6 | 品質統計を取る | | |
| | 7 | 製品を設計する | | |
| | 8 | 製品の設計を変更する | | |
| | 9 | 製品情報を管理する | | |
| | 10 | 標準仕様の原価を予測する | | |
| | 11 | 製品品質の基準を決める | | |
| | 12 | 工程を設計する | | |
| | 13 | 作業品質の基準を決める | | |
| | 14 | 設備品質の基準を決める | | |
| | 15 | 最適運転条件を求める | | |
| 4 | 16 | 注文を受ける | | |
| | 17 | 内示注文を受ける | | |
| | 18 | 納期を回答する | | |
| | 19 | 注文を生産計画に引き当てる | | |
| | 20 | 出荷を指示する | | |
| | 21 | 調達要求を出す | | |
| | 22 | 定期保守を計画する | | |
| | 23 | 生産計画を立てる | | |
| | 24 | 注文品の生産計画を立てる | | |
| | 25 | 試作品の生産計画を立てる | | 試験用製品の制作依頼をかける |
| | 26 | 個別仕様の原価を予測する | | |
| | 27 | 計画に部品、資源を割り当てる | | |
| | 28 | 端数在庫を優先する | | |
| | 29 | 仕掛在庫を優先する | | |
| | 30 | 順序計画を立てる | | 納期状況に対応したスケジューリングを行う 製造現場トラブルに対応したスケジューリングを行う |
| | 31 | 生産能力を把握する | | 作業者の生産能力を更新する |
| | 32 | 実在庫を把握する | | 在庫データを渡す |
| 33 | 製造を指示する | | 生産計画をMESに渡す | |
| 34 | 製造実績報告を受け取る | | | |
| 35 | 品質を追跡する | | 顧客からのクレームの原因を調査する | |
| 36 | 異物混入の製品を特定する | | 異物混入の出荷品を特定する | |
| 37 | 品質問題の原因を追及する | | | |

| レベル | 番号 | ユースケース名 | ユースケース記述の有無 | |
|-----|------|-------------------|-------------|-------------------------|
| | | | 有無 | ユースケース名 |
| | 38 | 調達を実施する | | |
| | 39 | 実績原価を把握する | | |
| | 40 | 監査に対応する | | 監査機関による工程の抜き打ち検査に対応する |
| | 41 | 稼働率、収率を把握する | | 生産システムの稼働履歴情報を取得する |
| 3 | 42 | 製造指示を受ける | | |
| | 42-1 | 製造指示の変更を受ける | | |
| | 43 | 作業指示を配布する | | |
| | 44 | 製造指示を制約内で変更する | | バリエーション「生産システムの順序を調整する」 |
| | 44-1 | 当日作業分のロットサイズを変える | | |
| | 44-2 | 段取り最小になるよう順序を変える | | |
| | 44-3 | 仕掛在庫を消化するよう順序を変える | | |
| | 45 | 作業実績報告を受ける | | |
| | 46 | 作業結果を評価して警告する | | |
| | 47 | 製造実績を報告する | | |
| | 49 | 予実を比較する | | |
| | 50 | 基準収率とチェックする | | 投入量と出来高に矛盾がないか確認する |
| | 51 | 進度を把握し、作業順を微調整する | | |
| | 52 | 仕掛かりの在庫を把握する | | |
| | 53 | 現品を追跡する | | |
| | 54 | 製造原価の基礎データを得る | | |
| | 55 | 小日程計画を立てる | | |
| | 56 | 作業文書を作成する | | |
| | 57 | 改善のための原データを記録する | | |
| | 58 | 工程内の改善を行う | | |
| | 59 | <生産性を上げる> | | |
| | 60 | 原データを分析する | | |
| | 61 | 資源の能力を計測する | | |
| | 62 | 作業者の能力を計測する | | |
| | 63 | 作業者を識別する | | 作業者を認証する |
| | 64 | 作業指示を知る | | |
| | 65 | 作業方法を知る | | |
| | 66 | 代替部品が利用可能か知る | | |
| | 67 | 作業文書を検索する | | |
| | 68 | 前段取りをする | | |
| | 69 | 原料を小分けする | | 原料の小分け指示を表示する |
| | 70 | キittingする | | |
| | 71 | 作業実施を指示する | | |
| | 72 | 作業実績を報告する | | 生産実績の受け渡し |
| | 73 | 作業中の異常を報告する | | |
| | 74 | 代替部品の使用を報告する | | 加工情報の変更 |

| レベル | 番号 | ユースケース名 | ユースケース記述の有無 | |
|-----|-----|-------------------|-------------|-------------------|
| | | | 有無 | ユースケース名 |
| | 75 | 材料の使用状況を報告する | | |
| | 76 | 中間品の生産状況を記録する | | |
| | 77 | 工程内検査を行う | | |
| | 78 | 後段取りする | | |
| | 79 | 担当装置の異常に対処する | | 異常復旧 |
| | 80 | 製品品質を検査する | | |
| | 81 | 品質情報を測定する | | |
| | 82 | 品質情報を記録する | | 品質情報を通知するタイミングを得る |
| | 83 | 故障状況を記録する | | 不具合データを収集する |
| | 84 | 故障状況を分析する | | |
| | 85 | 保守実績を記録する | | |
| | 86 | 治工具のメンテを指示する | | |
| | 87 | 治工具のメンテを予測する | | |
| | 88 | 定期保守を計画する | | |
| | 89 | 定期保守を実施する | | 定期的な設備保守を実施する |
| | 90 | 設備診断を行う | | |
| | 91 | 入庫する | | |
| | 92 | 出庫する | | |
| | 93 | 棚卸する | | |
| | 94 | 出荷指示を受ける | | |
| | 95 | 調達品を受け入れる | | |
| 0-2 | 96 | 運転状況を記録する | | |
| | 97 | 運転状況を報告する | | |
| | 98 | 作業開始を記録する | | 作業開始 |
| | 99 | インライン計測データを記録する | | |
| | 100 | 不具合情報を記録する | | 不具合データを収集する |
| | 101 | 作業終了を記録する | | |
| | 102 | 治工具の寿命を管理する | | |
| | 103 | 工具の使用実績を記録する | | |
| | 104 | 装置にプログラムをダウンロードする | | |
| | 105 | レシピをダウンロードする | | 作業開始 |
| | 106 | 作業実施の指示を受ける | | |
| | 107 | 最適運転条件を設定する | | |
| | 108 | 故障状況を報告する | | |
| | 109 | 設備、治工具の段取りの指示を受ける | | |
| | 110 | 設備、治工具を準備する | | 工具の払い出し、取り付け |
| | 111 | 設備、治工具を片づける | | 工具の取り外し、返却 |

上記の手順において、前述のユースケースを参考にして、実行系のシステムに表れる概念とそれらの関係をタイプ図として表している。このタイプ図がインタフェースを規定する際のオント

ロジとなると考えている。また、インタフェースのプロトコルで受け渡されるデータは、ここで完成したモデルで規定されたものとする。

以下に、これまでに上がったユースケース、および、ユースケース記述から作成したタイプ図を示す。

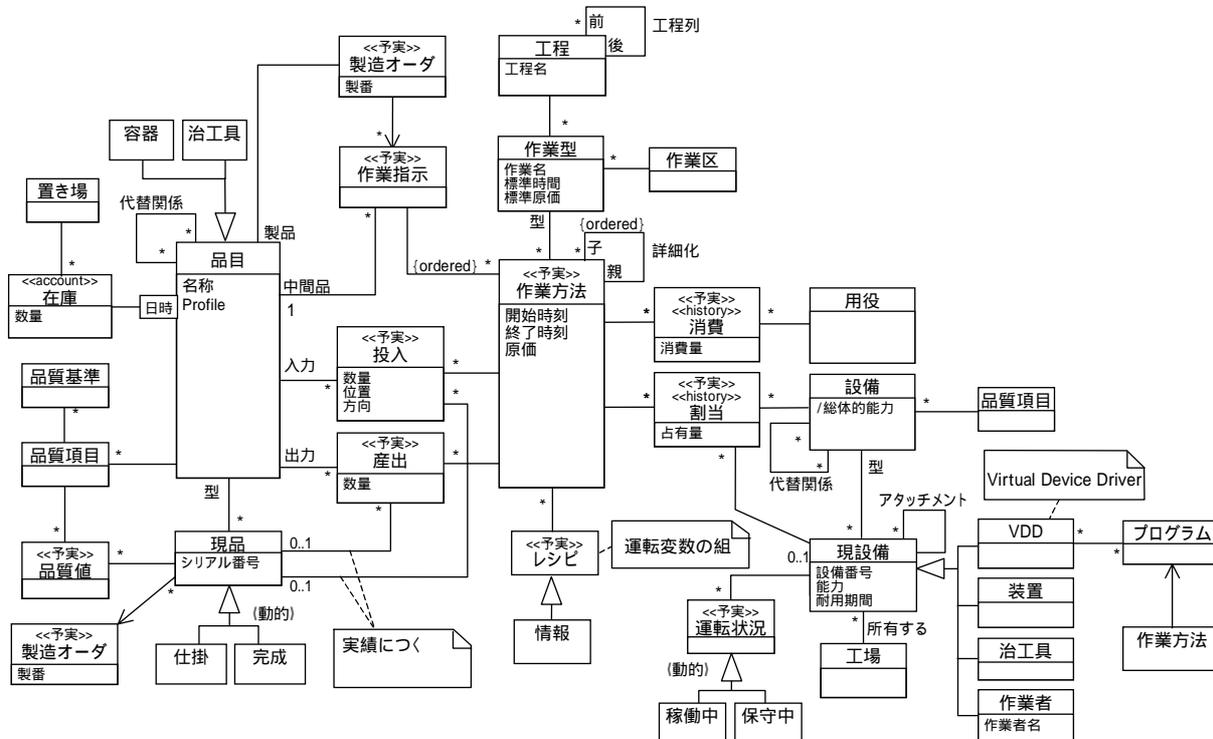


図 2 - 1 2 実行系システムのタイプ図 (例)

タイプ図で使うステレオタイプは《予実》と《history》である。

《予実》の意味は、予定に関わるサブタイプと実績に関わるサブタイプを持ち、双方の関連の多重度が「0.1」となっている構造である。多重度の「0」は、それぞれ予定が履行されない場合および予定がなくても実施される場合を表している。

(2) MES アプリケーション間情報連携モデルと方法論

MES アプリケーション間インタフェースはMESの諸機能を実現するため機能サブシステム間 (MES - MES 間) を情報連携するしくみである。

前述のとおりMESアプリケーション間情報連携モデル構築の方法としては、ISO 16100(製造ソフトウェア・ケイパビリティ・プロファイリング)で規定されている方法論を分野別に適用する方向で検討されている。

ISO 16100とは、製造ソフトウェアの相互接続性、互換性を実現するために、製造ソフトウェアのケイパビリティ(能力)を記述するための規格である。実際には、製造ソフトウェアのプロセスや製造ソフトウェアが使うリソース、製造情報などの相互運用性に関する情報をUMLでモ

デル化し、その情報を XML でプロファイルとして記述し、一般に共有可能なデータベーススキーマを開発する。

具体的な方法としては、ISO 16100 のケイパビリティプロファイリング構築手法に基づき、製造用アプリケーションソフトウェアユニットに関する各分野別の関連モデル記述（プロセスモデル、情報モデルなど）と、情報連携モデルとしてはソフトウェアユニットのケイパビリティテンプレートの構築を行う。

ISO 16100 で規定されるソフトウェアインタフェースの要求により、以下の開発が促進される。

- 相互運用可能な設計と製造ソフトウェアツール（これにより、製品開発時間が短縮できる）
- 現状技術の統合化による新しいソフトウェアツール
- レガシーシステムの置き換え支出を抑える、新しいアプリケーションソフトウェア
- プログラミングインタフェースとデータベーススキーマ（ソフトウェア・インテグレーションのために専有のインタフェースを開発しなくてもよいことにより、コスト節約が可能）

その結果、生産コストと製造情報管理コストを低減することができる。

また、ISO 16100 では、次に示す内容を提供することで、製造ソフトウェアのインテグレーションを可能にする。

- 異なるベンダーに開発された産業オートメーションシステムのソフトウェアユニットの間で情報交換を許す標準インタフェース仕様
- ユーザに要求機能を満たすソフトウェアユニットを選択することができるようにするためにソフトウェア・ケイパビリティ（能力）・プロファイリングの標準的な方法
- ソフトウェア・インテグレーションの完全性を保証する適合テスト

(a) ISO 16100 Part 構成と概要

現在標準化作業として進められている ISO 16100 の Part 構成と概要について示す。

| PartNo. | 内容 |
|---------|---------------------|
| Part1 | フレームワーク |
| Part2 | プロファイル方法論 |
| Part3 | インタフェースプロトコルとテンプレート |
| Part4 | 適合テスト方法、基準と報告 |

図 2 - 1 3 ISO 16100 の Part 構成

Part1 では、製造ドメイン内で使用されるソフトウェア製品と製造アプリケーションへその統合を促進するための相互運用性フレームワークを定義する。フレームワークは、情報交換モデル、

ソフトウェア・オブジェクト・モデル、サービス、インタフェースおよびプロトコル、ケイパビリティ（能力）・プロファイリング、そして、適合テスト方法を定義する。

Part2 では、ソフトウェア・ケイパビリティ（能力）の相互運用を可能にするための情報共有および交換モデル、ソフトウェア・オブジェクト・モデルおよびコンテキストを定義する。このパートでは、ソフトウェア・オブジェクトのクラス、共有インタフェースを識別するクラスの関係、およびインタフェースを介して交換される項目を含んでいる。この部分は、共有されている情報のシンタックスおよびセマンティクスをカバーし、関連するモデルを表わすために UML を使用している。

Part3 では、製造ソフトウェアユニットの相互運用を可能にするインタフェースのサービスとプロトコルを定義する。インタフェースは、IDL など適切な言語を用いて記述される。また、ソフトウェア・オブジェクト間の情報交換を可能にするサービス定義およびプロトコル仕様書は、この部分の中で定義される。

(b) フレームワークと対象ドメイン

製造ソフトウェアのためのフレームワークは製造アプリケーションの相互運用フレームワークに基づくものとする。このようなアプリケーション相互運用フレームワーク（ISO 15745-1 で詳細に説明されている）は、製造アプリケーションアーキテクチャ内にオートメーションとコントロールシステムのアーキテクチャを統合する根拠を提供するとしている。

図 2 - 14 で示されるように、統合製造アプリケーションは、製造プロセス、リソース、およびリソース間で共有、交換されるの情報構造のコンビネーションとしてモデル化されるものとします。

例えば、リソースとはアプリケーションによって要求されたプロセスと情報の交換を支援するのに必要なコミュニケーション・ネットワーク、デバイス、ソフトウェア、設備、資源、および、人である。

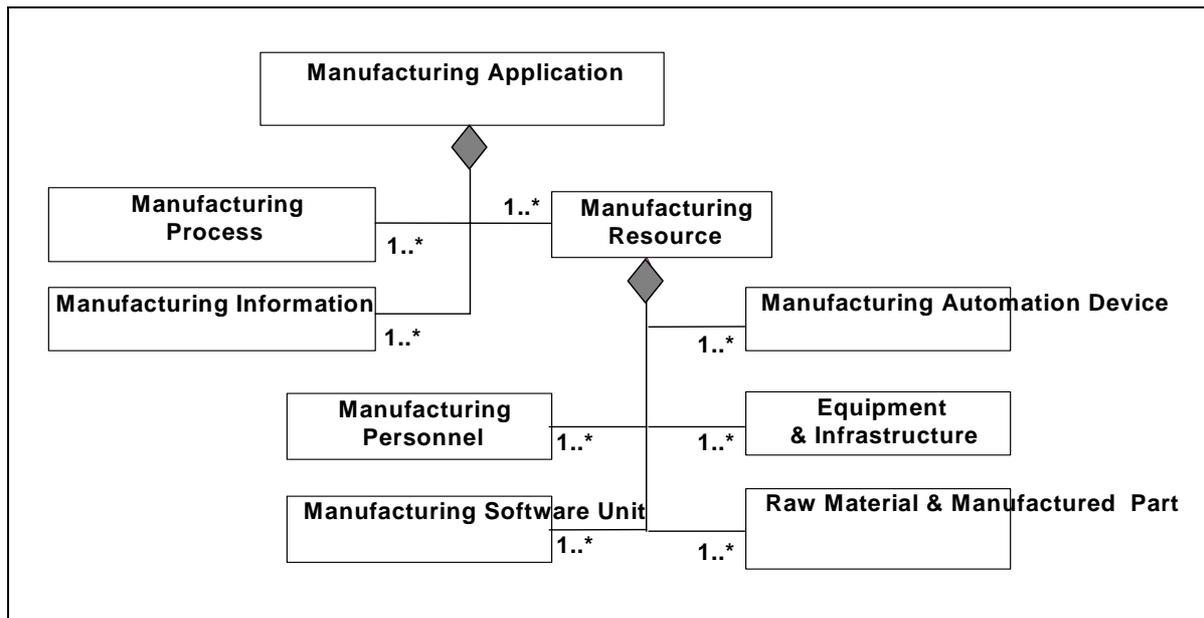


図 2 - 1 4 製造アプリケーションのクラス図

ISO 16100 がターゲットとしている製造ドメイン(対象)は、図 2 - 1 5 に示すとおりディスクリート系、バッチ系、連続制御系を含み、あらゆる産業を包含する。

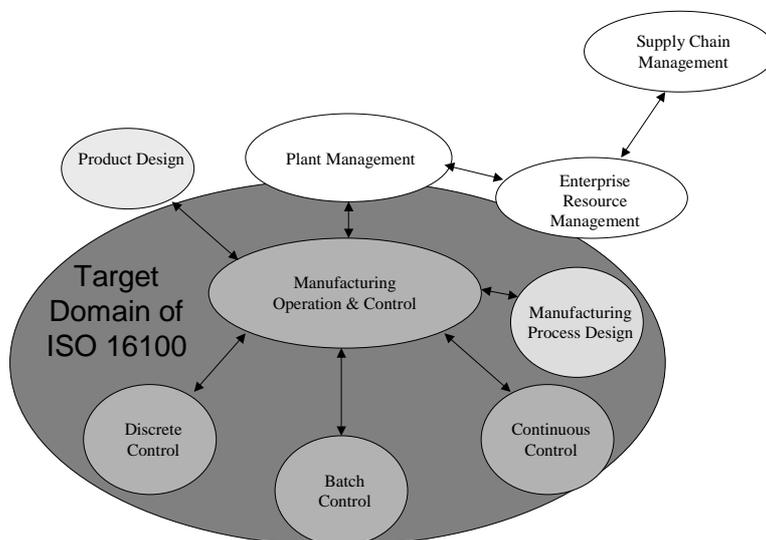


図 2 - 1 5 ISO 16100 のターゲットドメイン

(c) ケイパビリティ(能力)・プロファイリングの方法

相互運用可能なソフトウェアの統合するケイパビリティ(能力)・プロファイリングのコンセプト

トを以下に示す。ソフトウェアユニットの相互運用性は、機能、インタフェース、構造に関連するケイパビリティ（能力）で記述されることができる。

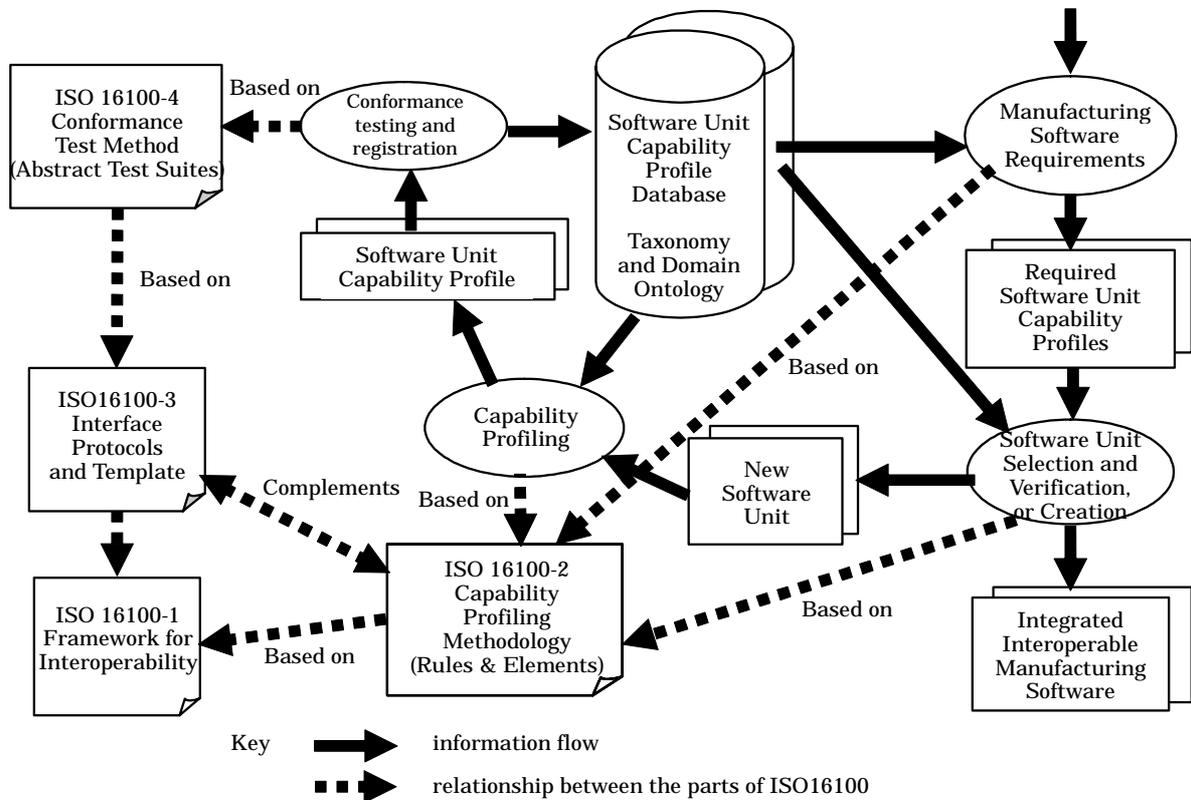


図 2 - 16 ケイパビリティ（能力）・プロファイリングのコンセプト

製造プロセスには、ネスト・階層の双方の構造があり、それぞれのレベルにおける製造ソフトウェアの必要条件とは、類似の構造で組織されたケイパビリティクラスのセットとしてモデル化できることである。

ソフトウェアユニットのプロファイリングは、ソフトウェアユニット、提供されるインタフェース、および、ターゲットのケイパビリティの必要条件としてサポートされるプロトコルをりよめることによって、実現可能な製造ケイパビリティについての簡潔なステートメントの生成を含んでいる。

このケイパビリティプロファイリングのメソロジーには、ユニット名、機能、その他必要なクラス属性についてケイパビリティプロファイルを記述する各々のソフトウェアユニットに関係するドメイン特有の特性と方法を使用する。

また、製造ソフトウェアの必要条件は、いくつかの製造ソフトウェアユニットをインテグレーションすることによって満たされる。具体的には、既存の製造ソフトウェアユニットの再利用、あるいは、新しい製造ソフトウェアユニットの開発によって、それぞれの製造ソフトウェアユニットを提供できる。

必要とされるプロファイルはデータベース内で既存のプロファイルと比較される。一致するものが見つかった場合は、ソフトウェアユニットプロファイルはインテグレーションのために準備ができていると考えられ、一致するものが見つからなかった場合は、必要とされるケイパビリティを持っているソフトウェアユニットは、新たに開発され、プロファイリングされ、そしてプロファイルデータベースに登録されるべきとなっている。

ISO 16100 が規定する製造用ソフトウェアユニットのケイパビリティプロファイルの内容には、以下の項目を含むことになっている。但し、全項目が必須ではない。

- type of manufacturing domain;
- type of manufacturing activity as differentiated by the process it is part of, the resources involved in conducting the activity, and the information types exchanged during the activity;
- type of computing system as differentiated by the operating environment, the software architecture, and the design pattern used;
- type of services, protocol, and data types used in running the software unit;
- supplier name, software version, and change history;
- performance benchmarks;
- reliability indices;
- service and support policy;
- pricing terms and conditions of use.

(d) 具体的な方向性

今後、今までに述べた ISO 16100 の Part1、および、Part2 で定義された方法論を適用し、分野別に対応テクノロジーを開発していく。また、それらの成果は、ISO 16100 の Part3 へ反映させていくこととなる。

但し、ISO 16100 には、以下にあげたような懸念事項があることを認識しており、同時にこれらの解決も行っていく必要がある。

● 規約の策定レベル

ワーキングドラフトであり、モデルやプロファイルの分類や作成方法の規定が不十分である。また、今後の審議で変更される可能性がある。

● システム全体のケイパビリティ（能力）記述

製造ソフトウェアの分類や個々のソフトウェアユニットの記述はできるが、個別システムのソフトウェア構成の記述規定が不明。

- システムの実リソースへの割付

システムの実リソース（装置、コントローラ、デバイス）への割付方法や記述規定が不明。

- UML-XML マッピング

UML にてモデルを作成し、XML にてプロファイルを記述することになっているが、UML 図から XML データを生成するマッピング規定が明確でない。そのため、独自に変換方法を決める必要がある。（標準の UML-XML マッピング規約としては OMG が XMI 仕様を策定している。）

2.2.4 MES-装置間情報連携のモデル

(1) MES - 装置間の情報連携

図 2 - 17 に示すように、ネットワーク上の SCADA にエンジニアリングデータベース上の装置情報と製造実行システム（MES）の設計情報からそれぞれの共通データモデルとの変換情報を用いてデータ変換を持つことで複数のデータ変換部を用意する必要がなく、接続対象を増やす毎に変換部を作成する必要がなくなる。接続対象の追加は、新たに追加された設備の情報をエンジニアリングデータベースから取得することで可能となる。エンジニアリングデータベースからの変換情報の取得は、ネットワークを介して実行時に行っても、事前に SCADA がエンジニアリングデータベースから変換情報を取得しても可能である。

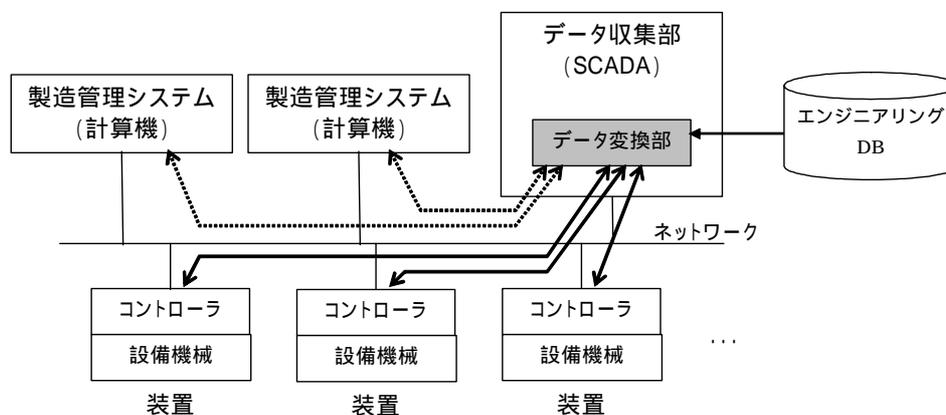


図 2 - 17 エンジニアリングデータベースの変換情報を用いたデータ変換事例

図 2 - 18 に示すように、エンジニアリングデータベースに格納された設計情報、例えば MES アプリケーションの設備接続仕様と設備装置の上位系接続仕様と設備の構成を示す設備構成仕様を用いてデータ変換手段のデータ変換を行う。MES アプリケーションプログラムは、設備接続仕様をもとにネットワークインタフェースを介してデータ変換部と通信を行い、設備装置にデータアクセスする。設備装置のプログラム/パラメータは上位系接続仕様をもとにネットワークイン

タフェースを介してデータ変換部と通信を行い、製造管理システムにデータを提供する。データ変換部は、設備接続仕様と上位系接続仕様と製造管理システムのMESアプリケーションプログラムや各設備装置の構成を示す設備構成仕様をもとに製造管理システムと設備装置間、あるいはMESアプリケーションプログラム間や設備装置間のデータ変換を行う。

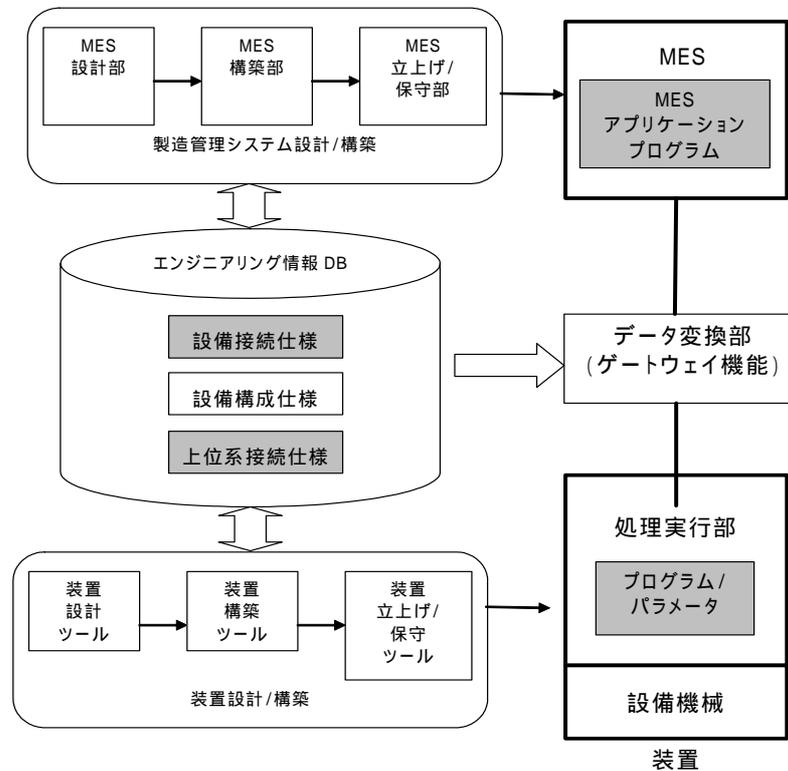


図 2 - 18 製造システム開発支援装置を用いたデータ変換

図 2 - 19 のデータ変換部の論理構成例を示す。予め定められた FAOP の MES 共通スキーマと装置共通スキーマを基に、設備接続仕様は MES 共通データに対応する MES アプリケーションプログラムの MES オリジナルデータを記述し、上位系接続仕様は装置共通データに対応する装置プログラム/パラメータの装置オリジナルデータを記述し、設備構成仕様は、実際の各 MES と各設備装置の構成とそれらが使う MES 共通データと装置共通データの関係性を記述する。データ変換部はこれらの情報を用いてデータ変換を行う。

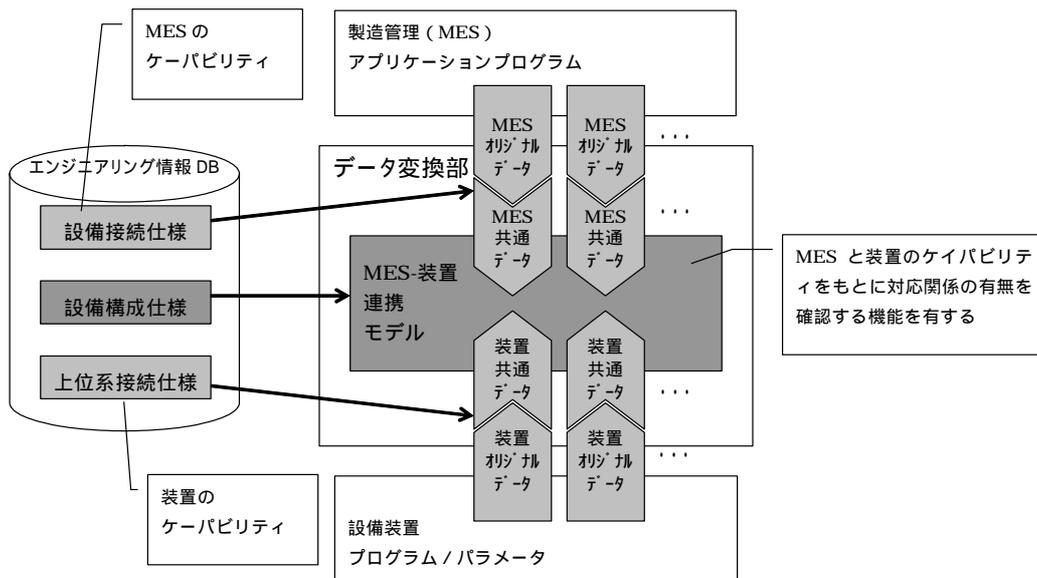


図 2 - 19 データ変換部の論理構成例

(a) ゲートウェイを MES 側の実装するデータ変換方式

図 2 - 20 に示すように、製造実行システム上にデータ収集部が存在し、データ収集部がエンジニアリングデータベース上の装置情報の設計情報から共通データモデルとの変換情報を用いてデータ変換を行う。エンジニアリングデータベースからの変換情報取得は、ネットワークを介して実行時に行っても、事前に製造実行システムがエンジニアリングデータベースから変換情報を取得しても可能である。

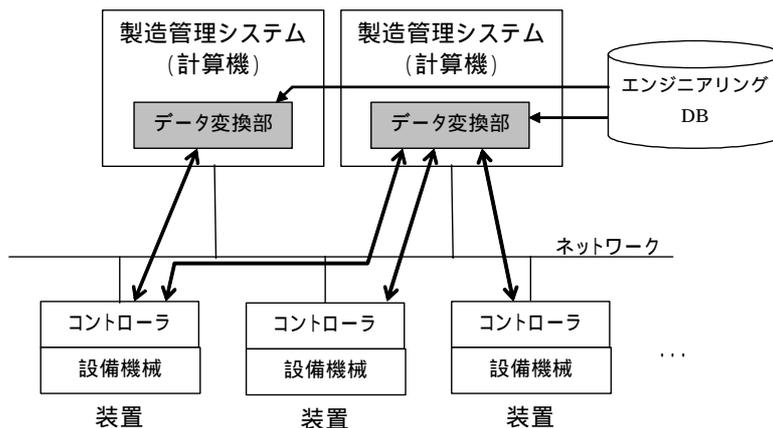


図 2 - 20 製造実行システムがデータ変換を行う例

(b) ゲートウェイを装置側に実装するデータ変換方式

図 2 - 2 1 に示すように、装置上にデータ収集部が存在し、データ収集部がエンジニアリングデータベース上の製造実行システム (MES) の設計情報からそれぞれの共通データモデルとの変換情報を用いてデータ変換を行う。エンジニアリングデータベースからの変換情報取得は、ネットワークを介して実行時に行っても、事前に製造実行システムがエンジニアリングデータベースから変換情報を取得し、各装置に設定しても可能である。

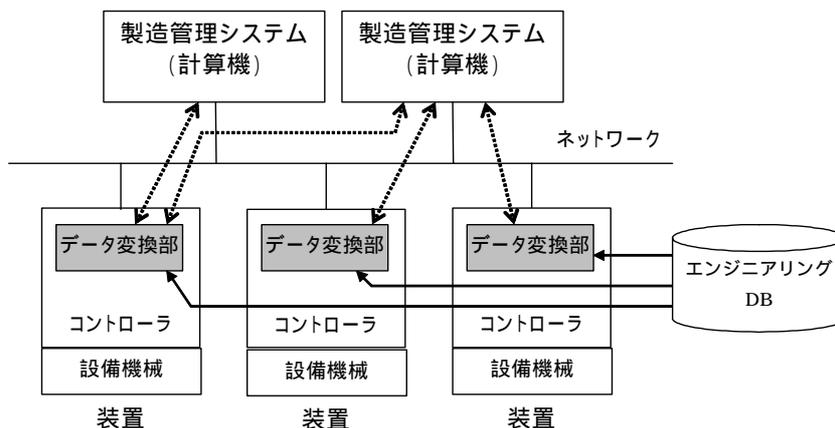


図 2 - 2 1 装置がデータ変換を行う例

(2) MES - 装置間情報連携の実装例

(a) MES アプリケーションの設備接続仕様

図 2 - 2 2 に MES アプリケーションの設備接続仕様の MES 共通データモデル構造を示す。MES アプリケーションが扱うデータは MES 機能ごとに MES 共通データを定める。そして、予め定められた MES 共通データに対して MES アプリケーションの実装ごとの MES オリジナルデータを記述する。

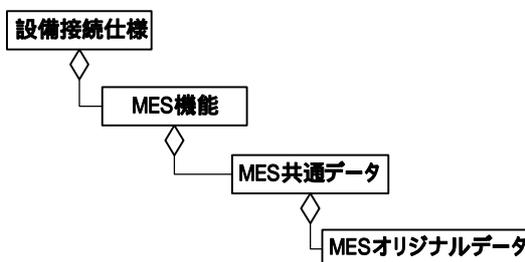


図 2 - 2 2 MES アプリケーションの設備接続仕様の共通データモデル構造

図2-23に図2-22に準じたMESアプリケーション仕様書のMES共通データモデル例を示す。

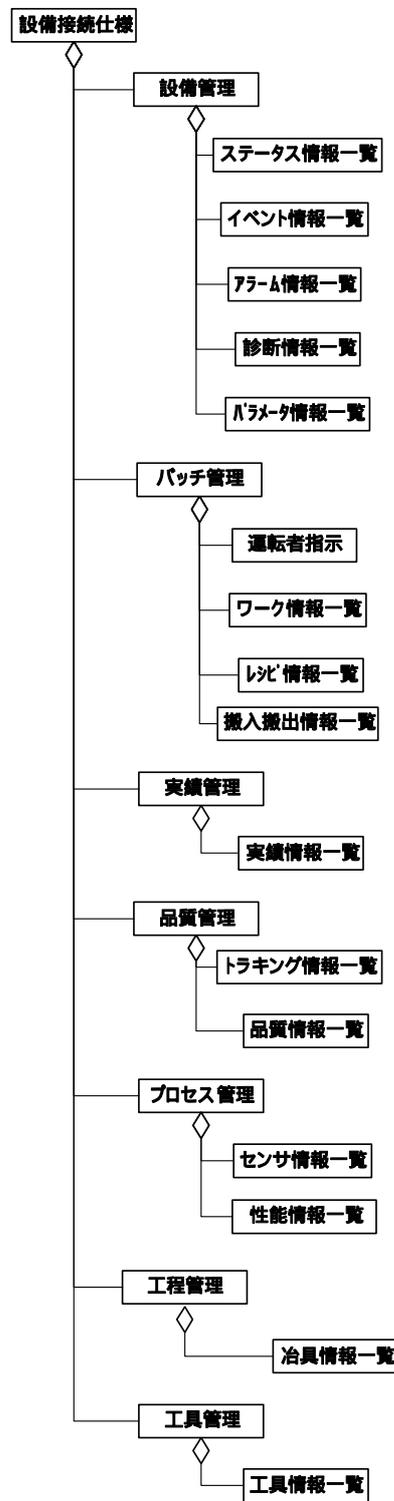


図2-23 MESアプリケーションの設備接続仕様のMES共通データモデル例

図 2 - 2 4 には、図 2 - 2 3 に対する MES アプリケーションの設備接続仕様の共通データ XML 例を示す。ただし、便宜上 XML タグは説明用語で記述する。実装時はアルファベット文字や数字など処理部に合わせた XML タグ名に対応する。MES オリジナルデータの記述内容は MES アプリケーションプログラムの実装方法に依存する。MES オリジナルデータの記述方法としては、SOAP (Simple Object Access Protocol) 等の XML メッセージ通信による Web サービスの場合は、WSDL (Web Service Description Language) で記述される。CORBA や DCOM 等のバイナリ通信による RPC (Remote Procedure Call) では IDL (Interface Description Language) で記述され、FTP やファイル共通プロトコルを用いた XML データのファイル転送では XSD (XML Schema Description) で記述される。他に、MES オリジナルデータは MES アプリケーションベンダー独自のデータフォーマットの場合もある。この場合、MES オリジナルデータの記述と解釈は MES アプリケーションベンダー依存になり、専用の解釈 / 変換機能が必要となる。

```

<設備接続仕様 name="設備装置 A">
  <MES 機能 name="設備管理">
    <MES 共通データ name="ステータス情報一覧">
      <MES オリジナルデータ 記述言語="WSDL"> . . . . . </MES オリジナルデータ>
    </MES 共通データ>
    <MES 共通データ name="イベント情報一覧">
      <MES オリジナルデータ 記述言語="WSDL"> . . . . . </MES オリジナルデータ>
    </MES 共通データ>
    . . . . .
  </MES 機能>
  <MES 機能 name="バッチ管理">
    <MES 共通データ name="運転者指示">
      <MES オリジナルデータ 記述言語="WSDL"> . . . . . </MES オリジナルデータ>
    </MES 共通データ>
    <MES 共通データ name="ワーク情報一覧">
      <MES オリジナルデータ 記述言語="XSD"> . . . . . </MES オリジナルデータ>
    </MES 共通データ>
    . . . . .
  </MES 機能>
  . . . . .
</設備接続仕様>

```

図 2 - 2 4 MES アプリケーションの設備接続仕様の MES 共通データ XML 例

(b) 装置の上位系接続仕様

図 2 - 2 5 に装置仕様書の上位系接続仕様の共通データモデル構造を示す。装置が MES アプリケーション等の上位系に提供するデータは図 2 - 2 5 示すように、装置機能ごとに共通データを定める。そして、予め定められた共通データに対して装置ごとの装置オリジナルデータを記述する。

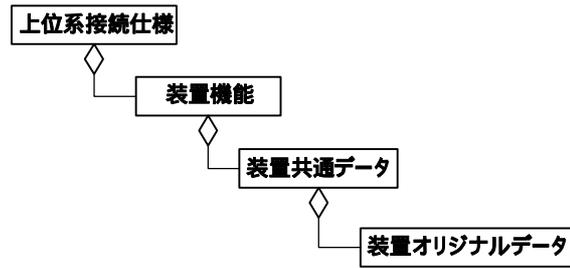


図 2 - 2 5 装置仕様書の上位系接続仕様の共通データモデル構造

図 2 - 2 6 に図 2 - 2 5 に準じた装置仕様書の上位系接続仕様の共通データモデル例を示す。

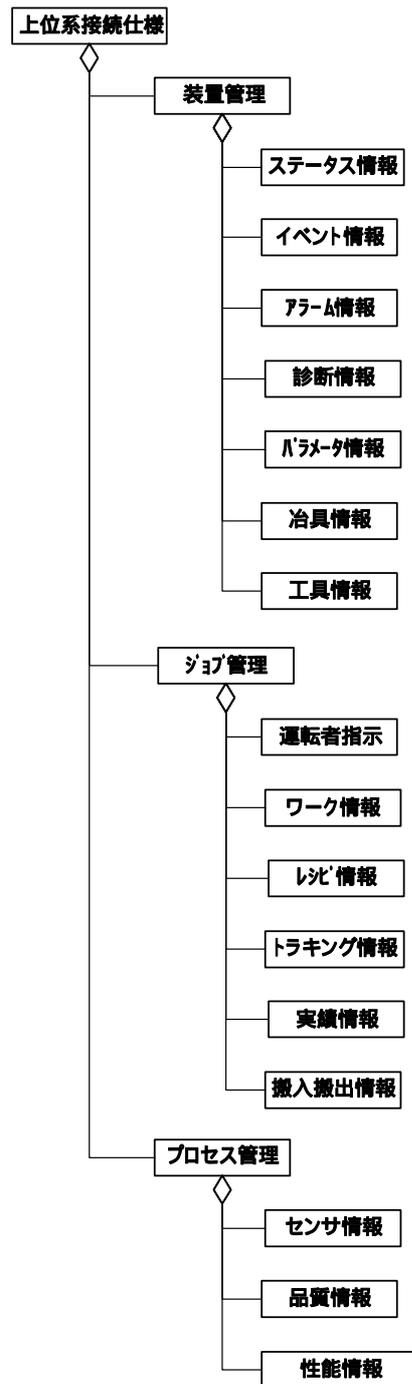


図 2 - 2 6 装置仕様書の上位系接続仕様の共通データモデル構造例

図 2 - 27 には、図 2 - 26 に対する MES アプリケーションの設備接続仕様の共通データ XML 例を示す。ただし、便宜上 XML タグは説明用語で記述する。実装時はアルファベット文字や数字など処理部に合わせた XML タグ名に対応する。MES オリジナルデータの記述内容は MES アプリケーションプログラムの実装方法に依存する。MES オリジナルデータの記述方法としては、SOAP 等の XML メッセージ通信による Web サービスの場合は、WSDL で記述される。CORBA や DCOM 等のバイナリ通信による RPC (Remote Procedure Call) では IDL (Interface Description Language) で記述され、FTP やファイル共通プロトコルを用いた XML データのファイル転送では XSD (XML Schema Description) で記述される。他に、装置オリジナルデータは装置メーカー独自のデータフォーマットの場合もある。この場合、装置オリジナルデータの記述と解釈は装置メーカー依存になり、専用の解釈 / 変換機能が必要となる。

```

<上位系接続仕様 name="設備装置 1">
  <装置機能 name="装置管理">
    <装置共通データ name="ステータス情報">
      <装置オリジナルデータ 記述言語="IDL">.....</装置オリジナルデータ>
    </装置共通データ>
    <装置共通データ name="イベント情報">
      <装置オリジナルデータ 記述言語="IDL">.....</装置オリジナルデータ>
    </装置共通データ>
    .....
  </装置機能>
  <装置機能 name="ジョブ管理">
    <装置共通データ name="運転者指示">
      <装置オリジナルデータ 記述言語="メーカ独自">.....</装置オリジナルデータ>
    </装置共通データ>
    <装置共通データ name="ワーク情報">
      <装置オリジナルデータ 記述言語="メーカ独自">.....</装置オリジナルデータ>
    </装置共通データ>
    .....
  </装置機能>
  .....
</上位系接続仕様>
    
```

図 2 - 2 7 装置仕様書の上位系接続仕様の共通データ XML 例

(c) 製造システムの設備構成仕様

図 2 - 2 8 に設備構成仕様のモデルを示す。設備内の製造実行システムの MES 機能の実体と設備内の設備装置の装置機能の自体の接続関係を設備構成関連で記述する。また、関連を記述した MES 機能と装置機能のそれぞれの MES 共通データと装置共通データのデータ構成関連を記述する。設備構成関連の情報によってデータ変換を行う対象の組み合わせを管理し、データ交換先がわかる。また、データ構成関連の情報によってデータ構造やデータタイプの変換、複数データの一括収集、時系列データの一括収集等のデータら変換を行う。

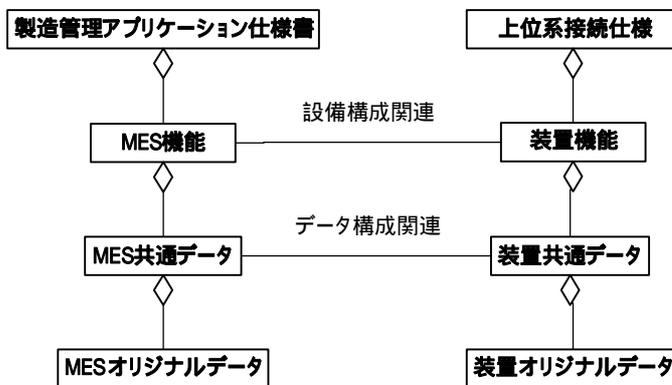


図 2 - 2 8 設備構成仕様モデル

図 2 - 29 に設備構成仕様の設備構成関連とデータ構成関連の XML 例を示す。

```

<設備構成仕様 name="設備 A">
  <設備構成関連 name="設備管理：MES1-設備装置 1/2">
    <MES 機能 name="設備管理" address="http://設備 A/MES1/設備管理">
      <装置機能 name="装置管理 1" address="http://設備 A/設備装置 1/装置管理"/>
      <装置機能 name="装置管理 2" address="http://設備 A/設備装置 2/装置管理"/>
    </MES 機能>
    <データ構成関連 name="ステータス情報一覧">
      <MES 共通データ name="ステータス情報一覧" 機能="設備管理">
        <装置共通データ name="イベント情報" 機能="装置管理 1"/>
        <装置共通データ name="イベント情報" 機能="装置管理 2"/>
      </MES 共通データ>
    </データ構成関連>
    . . . . .
  </設備構成関連>
  . . . . .
  <設備構成関連 name="バッチ管理 1">
    <MES 機能 name="バッチ管理 1" address="http://設備 A/MES1/バッチ管理/1">
      <装置機能 name="ジョブ管理 1" address="http://設備 A/設備装置 1/ジョブ管理"/>
    </MES 機能>
    <データ構成関連 name="運転者指示 1">
      <MES 共通データ name="運転者指示" 機能="バッチ管理 1">
        <装置共通データ name="運転者指示" 機能="ジョブ管理 1"/>
      </MES 共通データ>
    </データ構成関連>
    . . . . .
  </設備構成関連>
  <設備構成関連 name="バッチ管理 2">
    <MES 機能 name="バッチ管理 2" address="http://設備 A/MES1/バッチ管理/2">
      <装置機能 name="ジョブ管理 2" address="http://設備 A/設備装置 2/ジョブ管理"/>
    </MES 機能>
    <データ構成関連 name="運転者指示 2">
      <MES 共通データ name="運転者指示" 機能="バッチ管理 2">
        <装置共通データ name="運転者指示" 機能="ジョブ管理 2"/>
      </MES 共通データ>
    </データ構成関連>
    . . . . .
  </設備構成関連>
  . . . . .
</設備構成仕様>

```

図 2 - 29 設備構成仕様の設備構成関連とデータ構成関連の XML 例

2.2.5 装置エンジニアリング情報連携モデル

(1) 装置エンジニアリングの情報連携

図 2 - 30 に示すように、MES や装置は設計ツールによって設計され、従来および新たに設計された設計情報は、電子データ化されてエンジニアリングデータベースに格納され、エンジニアリングデータベースに格納された設計情報を用いて新たな設計を行い、構築ツールによって

MES や装置の構築/立上げ/保守に利用することができる。

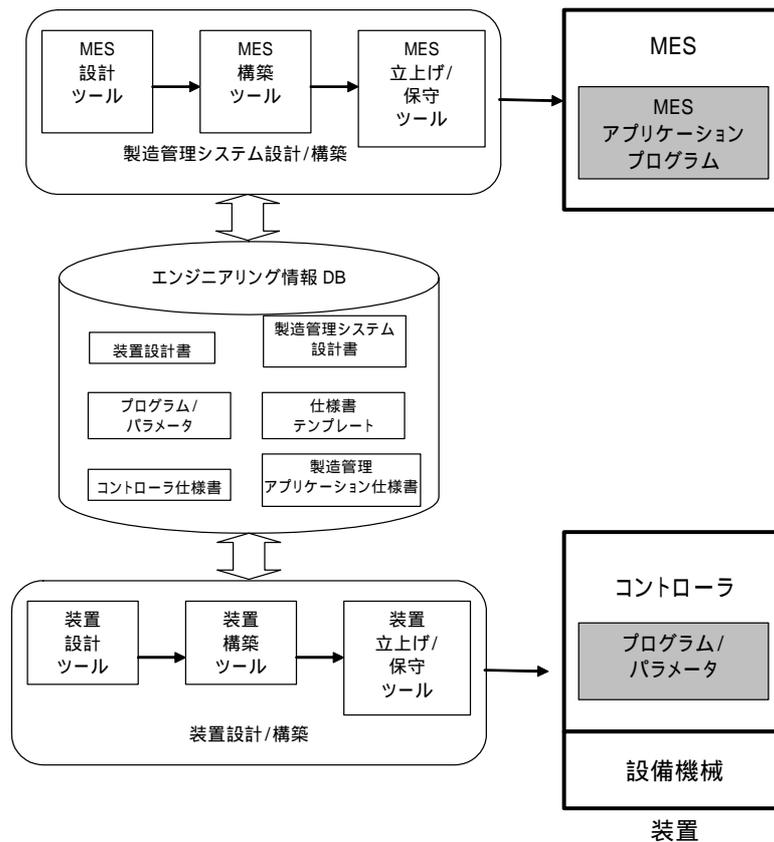


図 2 - 3 0 製造システム開発支援装置

装置設計を例に設計支援を説明すると、図 2 - 3 1 に示すように設計書のテンプレートをエンジニアリングデータベースから取出し、設計部においてテンプレートに設計情報を埋めてゆく。設計完了後は、エンジニアリングデータベースに装置設計書を格納する。設計部は、CADソフトウェアやプログラミングソフトウェアなどの装置設計ソフトウェアである。図面などのCADデータやプログラムなどのバイナリデータはエンジニアリングデータベースにそのファイルを格納し、テンプレートにはそのファイルパスを記入する。

参考にする既存の設計書が存在する場合は、テンプレートの代わりに既存の設計書を取り出し、必要部分を追加・変更して設計を行い、設計完了後、別バージョンの装置設計書として、エンジニアリングデータベースに格納する。

共通の電子データ管理方式を使うことで、テンプレート利用や既存設計書の流用が容易になり、設計の効率化が図れる。

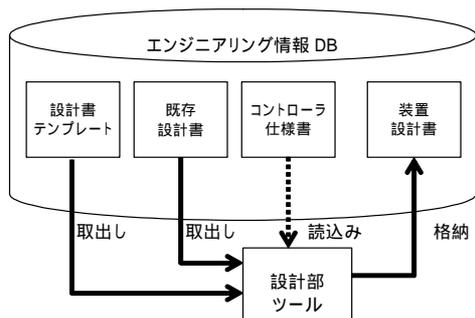


図 2 - 3 1 装置設計支援

装置構築を例に構築支援を説明すると、図 2 - 3 2 に示すように設計が完了した装置設計書をエンジニアリングデータベースから取出し、装置を構成する設備機械を製作し、コントローラを構成するユニットを制御盤に組み付け、コントローラの入出力デバイスを設備機械に接続し、装置を工場に設置して、ネットワークインタフェースをネットワークに接続する。これらの作業は通常いくつもの業者に分割され作成されるため、従来の紙による図面や仕様書では伝達や管理が大変であったが、共通の電子データ管理方式を使うことで、効率化することが可能になる。

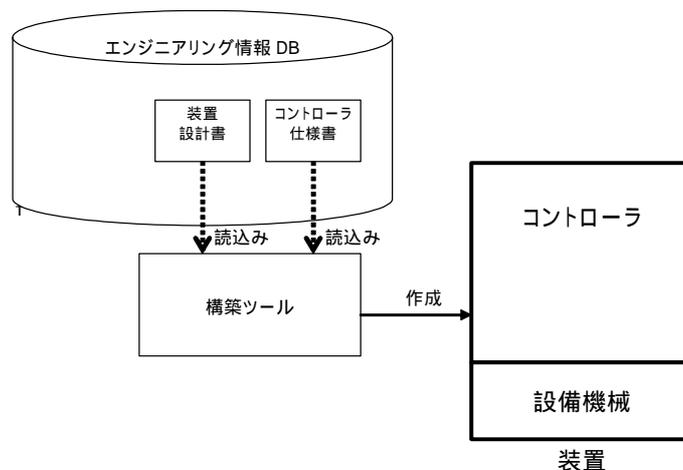


図 2 - 3 2 装置構築支援

装置立上げ/保守を例に立上げ/保守支援を説明すると、図 2 - 3 3 に示すように設計が完了した装置設計書をエンジニアリングデータベースから取出し、装置を構成する設備機械を製作し、コントローラを構成するユニットを制御盤に組み付け、コントローラの入出力デバイスを設備機械に接続し、装置を工場に設置して、ネットワークインタフェースをネットワークに接続する。これらの作業は通常いくつもの業者に分割され作成されるため、従来の紙による図面や仕様書では伝達や管理が大変であったが、共通の電子データ管理方式を使うことで、効率化することが可

能になる。

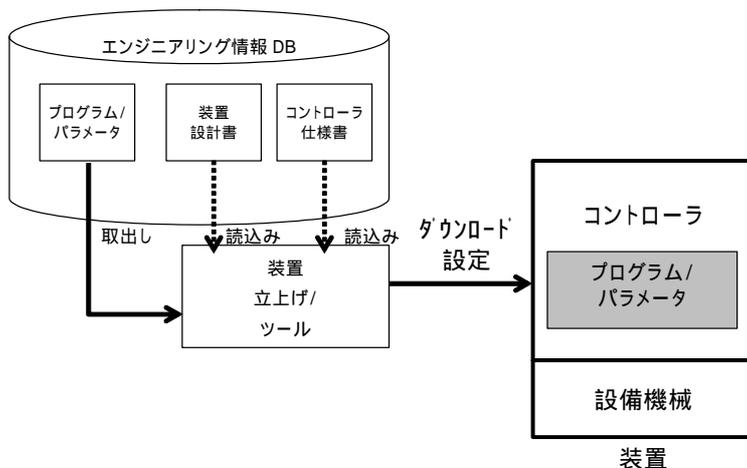


図 2 - 3 3 装置立上げ/保守支援

(2) 装置エンジニアリング情報連携の実装例

(a) 装置設計アクティビティ例

- 設備仕様設計
- 機械設計
- 制御盤設計
- 装置ソフトウェア設計

(b) 設計・構築ドメインオブジェクトモデル

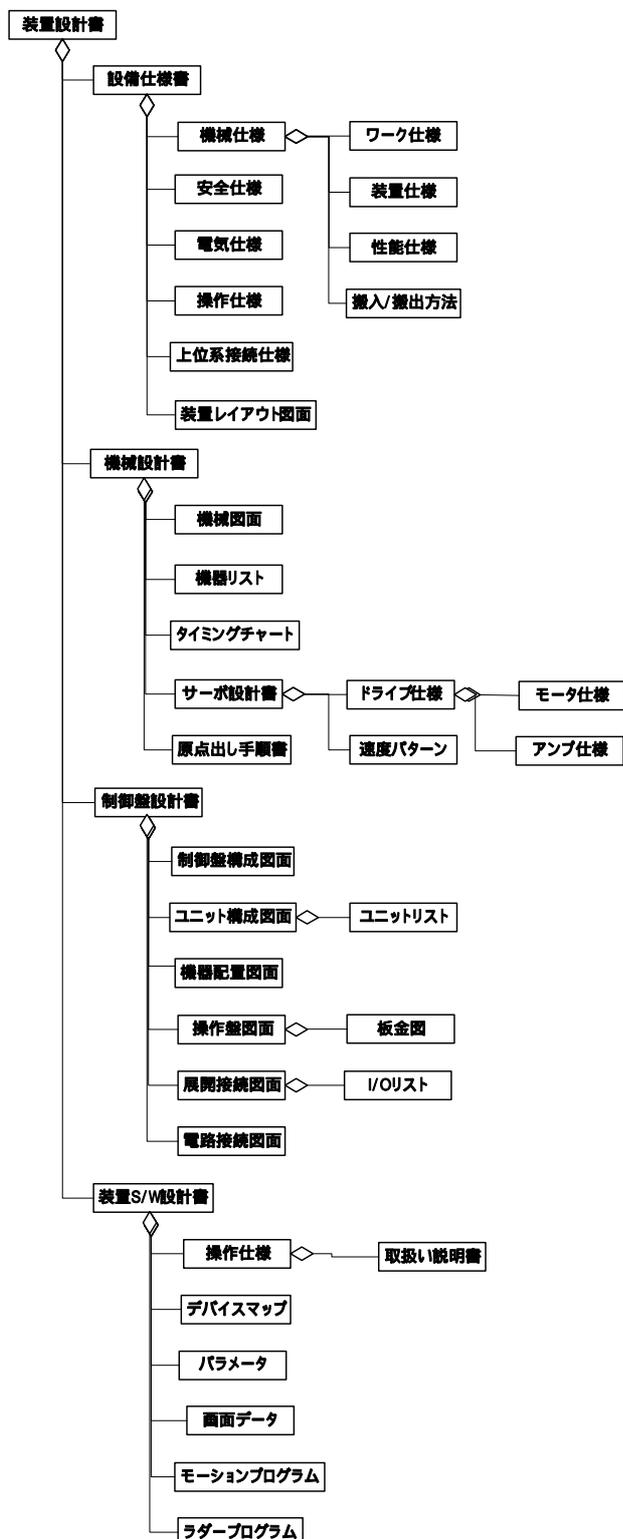


図 2 - 3 4 装置設計ドメインオブジェクト例

図 2 - 3 4 の UML のクラス図で記述された構造を XML 構造にした例を図 2 - 3 5 に示す。XML 構造のテンプレートの中に設計情報を埋め込み、エンジニアリングデータベースに格納する。設計情報を共通データモデルで記述された XML 構造を使って登録、削除、複製、移動、検索等の管理をおこなう。

```

<?xml version="1.0" encoding="UTF-16"?>
<装置設計書>
  <設備仕様書>
    <機械仕様>
      <ワーク仕様> . . . . . </ワーク仕様>
      <装置仕様> . . . . . </装置仕様>
      <性能仕様> . . . . . </性能仕様>
      <搬入搬出方法> . . . . . </搬入搬出方法>
    </機械仕様>
    <安全仕様> . . . . . </安全仕様>
    <電気仕様> . . . . . </電気仕様>
    <操作仕様> . . . . . </操作仕様>
    <上位系接続仕様> . . . . . </上位系接続仕様>
    <装置レイアウト図面> . . . . . </装置レイアウト図面>
  </設備仕様書>
  <機械設計書>
    <機械図面 filePath=" . . . . . "/>
    <機器リスト> . . . . . </機器リスト>
    <タイミングチャート> . . . . . </タイミングチャート>
    <サーボ設計書> . . . . . </サーボ設計書>
    <原点出し手順書 href=" . . . . . "/>
  </機械設計書>
  <制御盤設計書>
    <制御盤構成図面 filePath=" . . . . . "/>
    <ユニット構成図面 filePath=" . . . . . "/>
      <ユニットリスト> . . . . . </ユニットリスト>
    </ユニット構成図面>
    <機器配置図面 filePath=" . . . . . "/>
    <操作盤図面 filePath=" . . . . . "/>
      <板金図 filePath=" . . . . . "/>
    </操作盤図面>
    <展開接続図面 filePath=" . . . . . "/>
      <IO リスト> . . . . . </IO リスト>
    <電路接続図面 filePath=" . . . . . "/>
  </制御盤設計書>
  <装置 SW 設計書>
    <操作仕様>
      <取扱い説明書 href=" . . . . . "/>
    </操作仕様>
    <デバイスマップ> . . . . . </デバイスマップ>
    <パラメータ> . . . . . </パラメータ>
    <画面データ filePath=" . . . . . "/>
    <モーションプログラム filePath=" . . . . . "/>
    <ラダープログラム filePath=" . . . . . "/>
  </装置 SW 設計書>
</装置設計書>

```

図 2 - 3 5 装置設計ドメインオブジェクトの XML テンプレート例

2.3 分野別モデリング対象調査

2.3.1 加工ラインシステム

シャフト加工ラインシステムは、生産管理システム（上位システム）、ラインコントローラ（兼RDBサーバ）、作業端末1（兼加工予約端末）、作業端末2（実績計上、工程既存端末）、CAM端末（兼稼働管理、マスタ情報入力端末）、NC旋盤、マシニングセンタ、NC研削盤、インコンベア、アウトコンベア、およびコンベア制御用シーケンサで構成され、LANで接続されている（図2-36）。

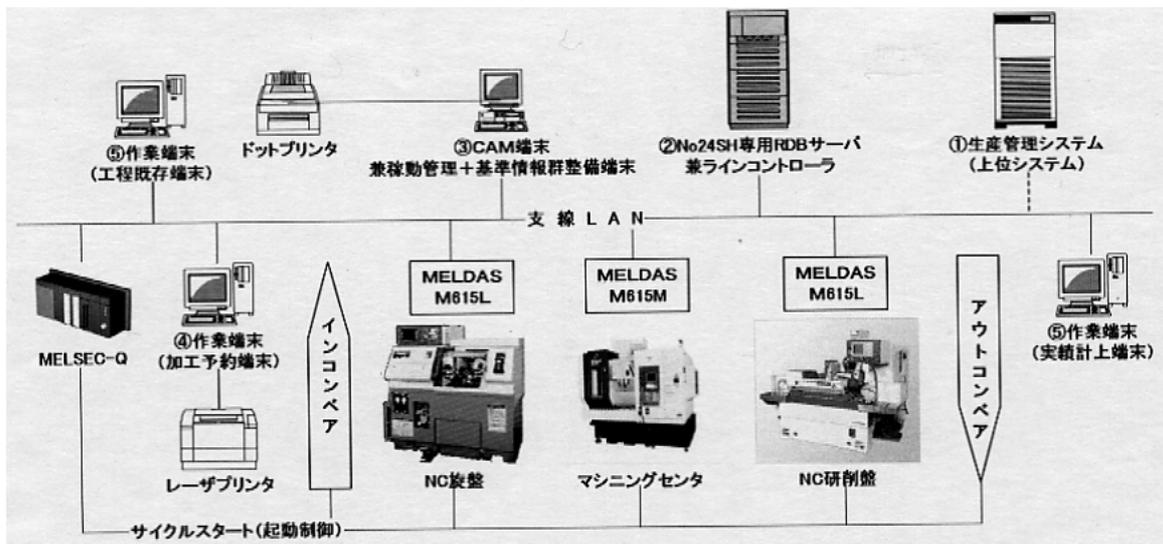


図 2 - 3 6 シャフト加工ラインシステムのハードウェア構成

生産管理システム（上位システム）では、顧客から注文を受けたモータを構成部品に展開し、部品毎にオーダ情報ファイル（部品番号、個数、納期など）を作成する。そのうち、シャフトに関するオーダ情報ファイルは本システムのラインコントローラが定期的（1回/日）に読み出し、データベース（Oracle）へ登録する。加工予約端末では登録された情報に基づき、加工効率（納期と段取り時間の関係を最適化する）を考慮し、仮の加工スケジュールを自動的に組み立てる。作業者はそのスケジュールに合わせ、CAM端末でNCプログラムを作成し、加工素材の準備を行う。加工に必要な作業がすべて完了すると、作業者は加工予約端末で加工スケジュールを確定し、作業指示表と現品票を印刷する。

次に作業者は確定したスケジュールに従い、インコンベアに素材を準備する。素材が準備できたことをシステムに伝えると、素材を旋盤にセットし加工を開始する。

シャフトは、旋盤、マシニングセンタ、研削盤の順に加工され、加工を完了するとアウトコンベアで搬出される。加工を完了したシャフトは、実績データとしてラインコントローラのデータベースへ登録される。最終的な実績データは、検査後作業者が実データに変更する。

(1) ラインコントローラの実行モデル

ラインコントローラには以下の3つのソフトウェアで構成される。

(a) 工作機械状態監視ソフトウェア

イーサネット経由でシーケンサから NC 装置の稼動状況を取得し、データベースに記録する。取得した稼動状況は作業端末の予約画面に表示し、作業者に異常を伝える。

(b) 加工時間管理ソフトウェア

シーケンサからシャフト加工時間を取得し、データベースに記録する。シャフト加工時間は作業端末の工程スケジュール機能で利用する。

(c) 工具寿命管理ソフトウェア

イーサネット経由で NC 装置が管理している工具情報を取得し、その情報をデータベースに記録する。その情報も作業端末の予約画面に表示し、作業者に異常を伝える。

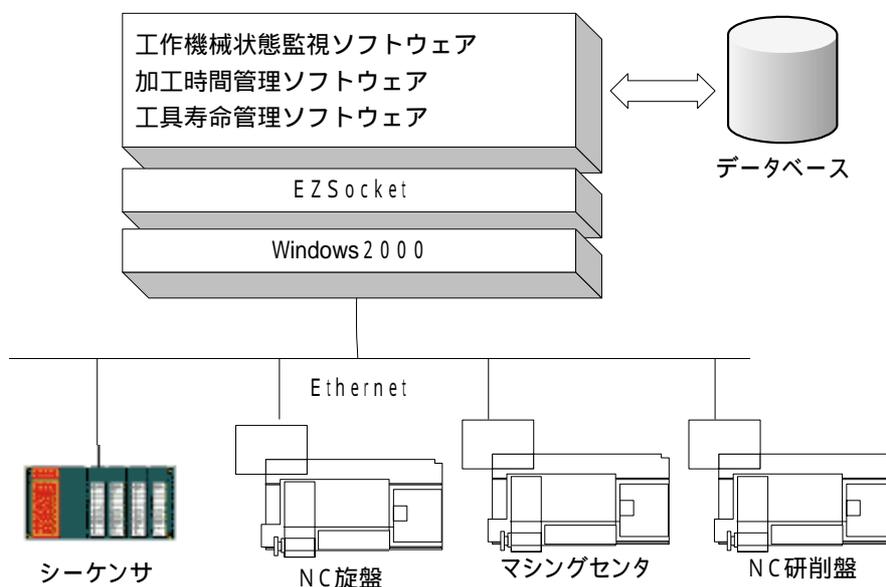


図 2 - 3 7 ラインコントローラのソフトウェア構成

図 2 - 3 8 にラインコントローラの実行モデル例を UML のクラス図で示す。MES、アクティビティ、ドメインを UML のステレオタイプ “<< >>” を使って示している。UML の集約によってアクティビティが管理するドメインオブジェクトを示している。

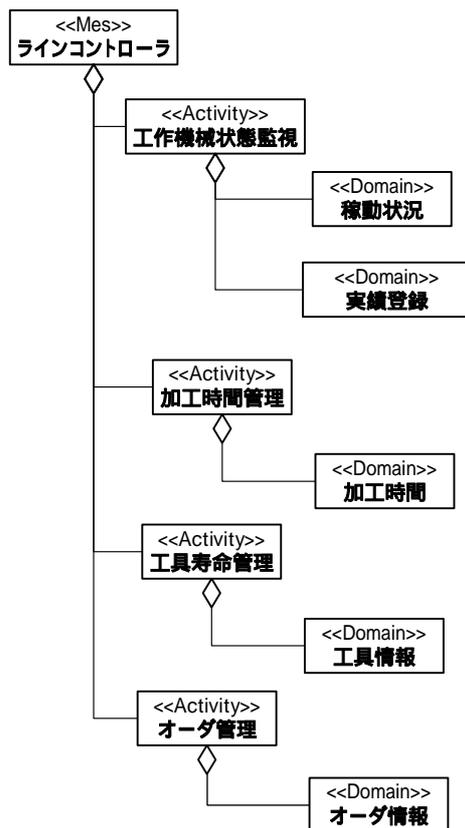


図 2 - 3 8 ラインコントローラの実行モデル例

(2) 作業端末の実行モデル

(a) 工程スケジュール機能

データベースに取り込まれたオーダ情報を、加工する順番に仮スケジュールする。

(b) 加工予約画面

作業者は作業端末の加工予約画面で、仮スケジュールを実際に加工する順序に並べる。

(c) NC プログラム管理

加工スケジュールを基に各 NC 装置に対して、NC プログラムのダウンロードを実施する。

図 2 - 3 9 に作業端末の実行モデル例を UML のクラス図で示す。MES、アクティビティ、ドメインを UML のステレオタイプ“<< >>”を使って示している。アクティビティが管理する度面員オブジェクトは集約で示し、他のアクティビティが管理するドメインを参照するときには関連で示している。

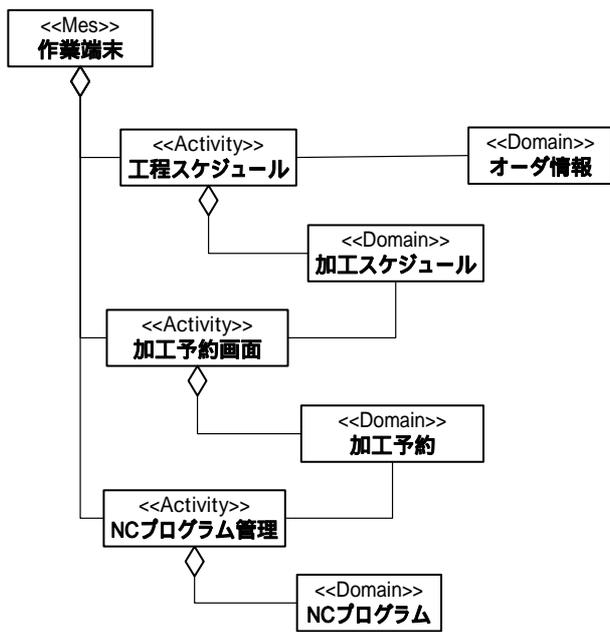


図 2 - 3 9 作業端末の実行モデル

(3) NC装置の実行モデル

NC装置（NC旋盤、マシニングセンタ、NC研削盤）の実行モデルを示す。この例では、ドメインオブジェクトは加工プログラムの例しか示していないが、各アクティビティには管理するドメインオブジェクトが存在する。

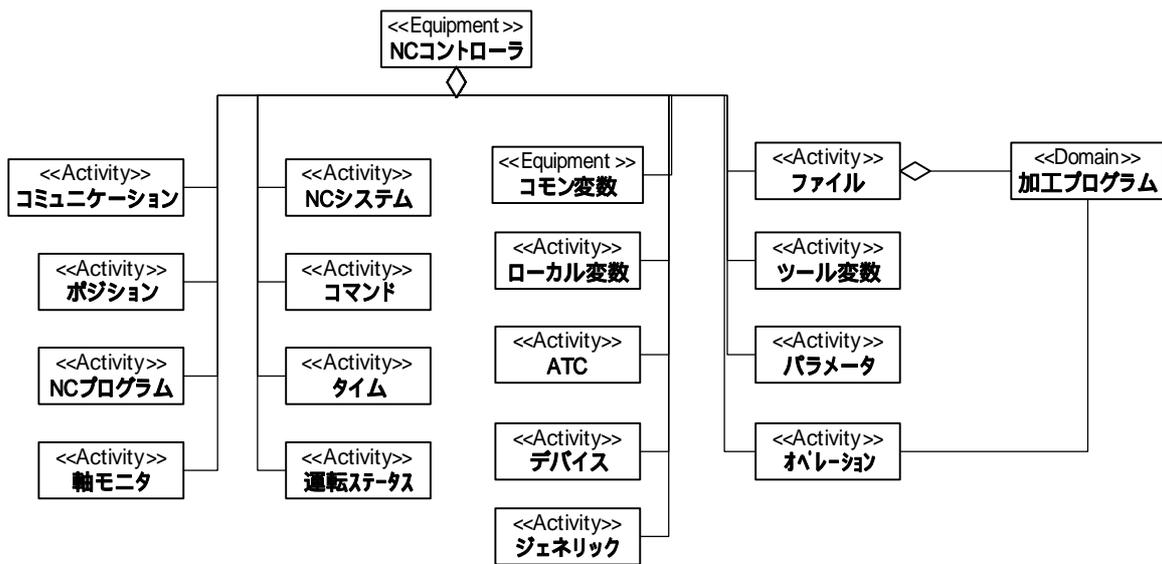


図 2 - 4 0 NC装置の実行モデル例

(4) ラインコントローラ - 作業端末 - 装置間のコラボレーション

オーダから加工スケジュールを立て、NC装置に加工プログラムをダウンロードし、実行させるまでのコラボレーション例を示す。

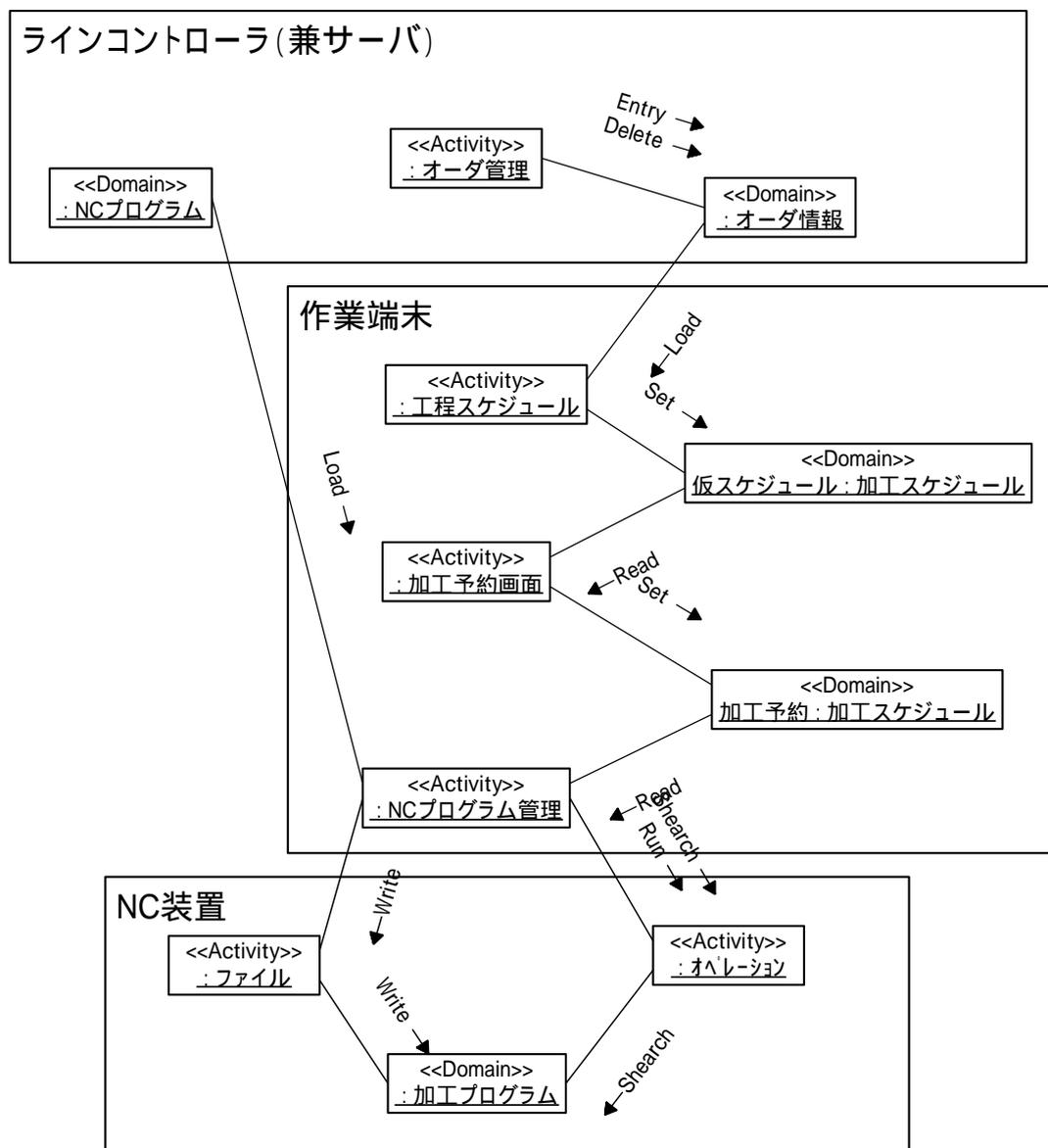


図 2 - 4 1 ラインコントローラ - 作業端末 - 装置間のコラボレーション例

2.3.2 機械加工

(1) MOII

MOII(Manufacturing Operation Information Interface)について、そのプロトコル(SNMP)と MIB の構造定義、および機器と上位情報システムとの接続について調査を行った。

(a) MOII 仕様

各種のFA機器がネットワークで相互接続されたFAのシステムは、管理コンピュータなどの「管理する側の機器」と、工作機械や搬送装置などの「管理される側の機器」で構成される(場合によって単一の機器が、管理する側と管理される側を兼ねることもある)。

管理する側の機器は、ある管理される側の機器を指定した上で、これに対して必要なおおじてステータス情報や、動作制御を要求し、管理される側の機器は要求に応じた情報を要求元に返したり、要求に応じた動作を実施したりする。

また、管理される側の機器は、管理する側の機器からの逐次の要求がなくても、事前の設定により、警報情報などを通知したりする。

ネットワークを介したこれら要求・応答あるいは警報メカニズムを達成するためには、管理する側とされる側において、主として

- 通信手続き(=プロトコル)を統一すること。
- プロトコルを会して送受信される情報が共通理解されること。

が必要となる。

MOII仕様では、機器の管理情報の収集、動作制御、警報通知などのために、イントラネットないし、インターネットで広く知られているSNMP(Simple Network Management Protocol)をプロトコルとして採用している。

また、SNMPでは、一般的に上記の情報共通理解をMIB(Management Information Base)という概念を用いて合理的に達成している。MOII仕様では、FA機器管理、制御のためのMIBを規定している。

(b) MOIIの構造

MOIIにおけるMIB定義はツリー構造になっている。具体的には、図2-42に示すように、iso(1).org(3).dod(6).internet(1).private(4).enterprise(4).mstc(9607)の下にMOIIのMIBが定義される。

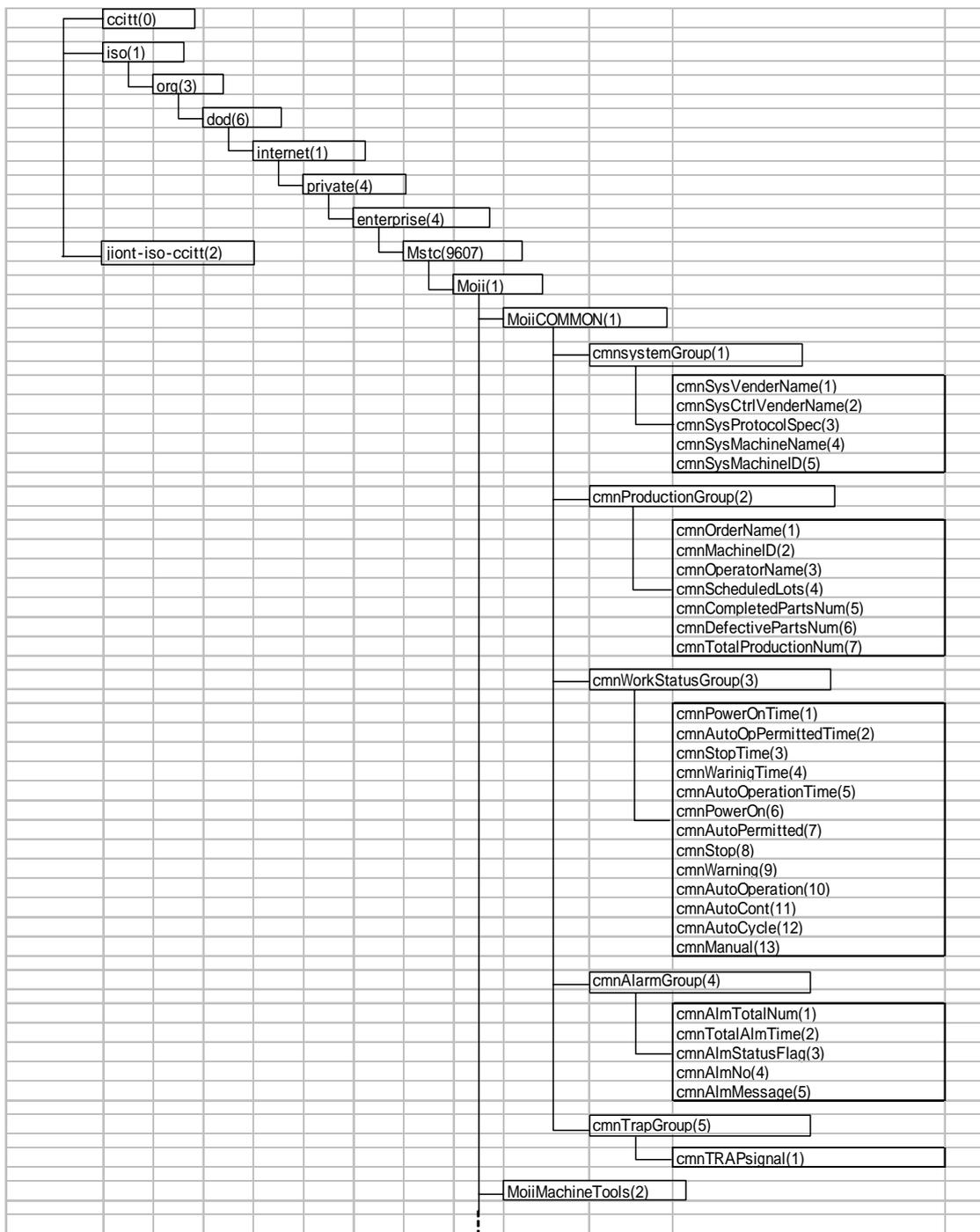


図 2 - 4 2 MOII 構成テーブル

MoiiCOMMON(1)は FA 機器共通管理データとなっている。

- ベンダー情報 [cmnsystemGroup(1)]

FA 機器の名称や、ベンダー名称など FA 機器固有の情報

- 生産情報〔cmnProductionGroup(2)〕
オーダ名や生産予定数、生産数など生産管理に必要な情報
- 稼動情報〔cmnWorkStatusGroup(3)〕
通電時間や停止時間などFA機器の稼動管理に必要な情報
- アラーム情報〔cmnAlarmGroup(4)〕
アラーム番号やアラームメッセージなどのアラーム情報
- トラップ情報〔cmnTrapGroup(5)〕
自動運転状態や警報などFA機器の状態変化を通知する情報

FA機器共通管理データは、通常上位システムからの要求により情報伝送が行われるが、トラップ情報は管理されているFA機器側からの通知で情報伝達を行う事が出来る。この機能は非常事態が発生した時の情報伝達手段として有用であり、きわめて特徴的である。

(c) 上位情報システムとの接続

図2-43は上位情報システムとFA機器の接続イメージである。業界標準のイーサネット上でSNMPプロトコルを用いて上位システム(SNMPマネージャ)から情報を要求し、FA機器側(SNMPエージェント)が共通管理データ(MIB)の形式で情報を送信する。

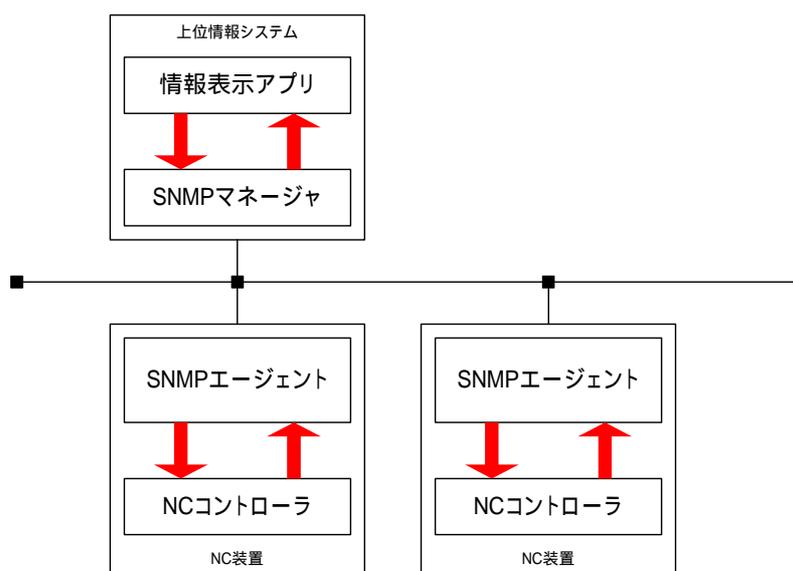


図2-43 上位システムとの接続イメージ

(2) 金型工場の無人スケジュール運転

「金型工場の無人スケジュール運転の事例」について調査し、そのモデル化に就いて検討を行った。

(a) 金型工場の無人スケジュール運転とは

- スケジュールは工場の管理パソコンより送付された複数の NC 加工プログラムデータに添えられる生産指示情報を元に、外段取りを実施するオペレータの判断で作成されている。
- 加工プログラムデータは、原点位置、取り付け姿勢、傾きなどを仮想して作られている。実際の加工はマシン上のワークの原点位置、傾きを測定し、測定データにより補正しながら DNC 運転により行われる。
- 使用工具が破損などにより利用できない場合などは、自動再スケジュールリングなどにより対応する
- レガシーな CNC 付き工作機械ではプロトコル A による DNC とシーケンサを介しての加工開始指令、加工終了、エラー情報などの受け渡しを行う。
- イーサネット対応 CNC 付き工作機械では、すべての情報がイーサネット経由で行われる。

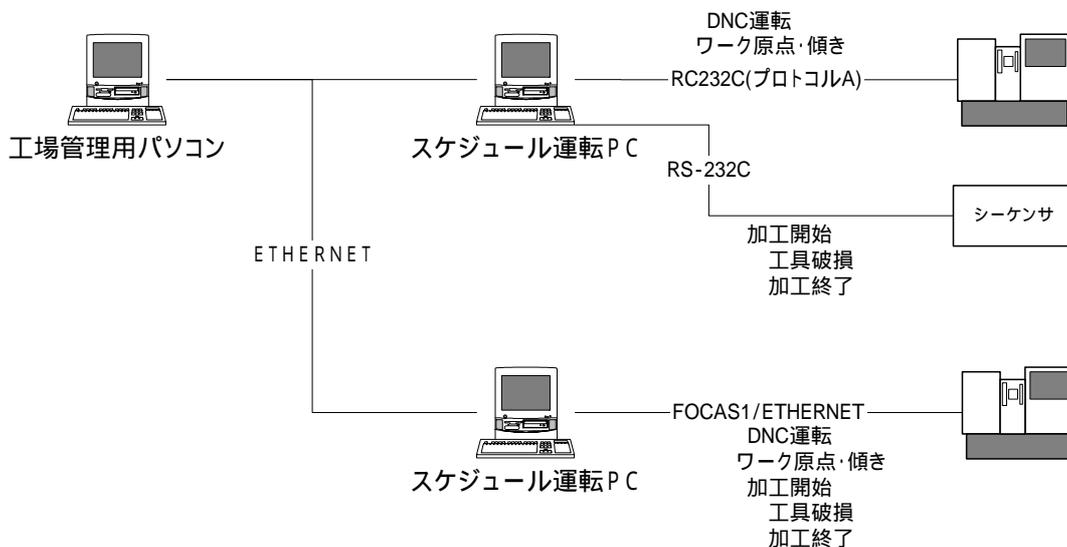


図 2 - 4 4 金型工場の無人スケジュール運転事例

- スケジュール運転 PC の機能
- スケジュールの入力
- 計測したワーク原点とワークの傾きから NC プログラムを補正
- DNC 運転
- 工具破損検知

- 工具破損時の自動再スケジューリング
- 工具寿命管理
- 治具管理

(b) モデル化

本システムにおけるシステム設計・構築フェーズ対応情報のモデル化により、システムのオペレーションマニュアル作成を設計・構築フェーズの情報と構造化するなどが可能となると考えられる。

設計・構築フェーズをモデル化することにより、システム設計プロセス、関連サブシステムに関するノウハウを含め、それらをプロファイルとして明示的に扱うことが出来るようになり、ランタイムモデルもこの情報から引き出せるのではと考えている

2.3.3 専用機

多軸サーボ・ロボット等で構成された専用装置に関し、「制御モデルに基づく構成の定義、順序定義によるわかりやすい制御の実現」に対するニーズと装置エンジニアリングモデルの現状を調査した。

(1) 対象システムの概要

ユーザの嗜好・ニーズの多様化に伴う市場環境の変化など、自動車産業を取り巻く環境は近年ますます激しく変化している。この様な変化に対し自動車製造各社において生産ラインのフレキシビリティを上げる取り組みがなされている。本項の調査対象はその方策の1つである大規模なFMS設備を取り上げ、設備の開発・維持における情報共有、連携の問題点を提起したい。

本装置の構成を(図2-45)に示す。本装置を制御するにあたり、多数のサーボモータ/アクチュエータを通常設備と同じマシンタクトで制御し、かつ柔軟性と保全性=わかりやすさを成立させる必要があるため、以下の特徴を備えている。

- プログラマブルなコントローラ、多軸サーボ、I/Oをネットワークで接続した構成
- メカ要素毎にモジュール化した制御システム構造
- 上記モジュールの動作の標準ブロック化とデータ定義によるインターロック制御の実現

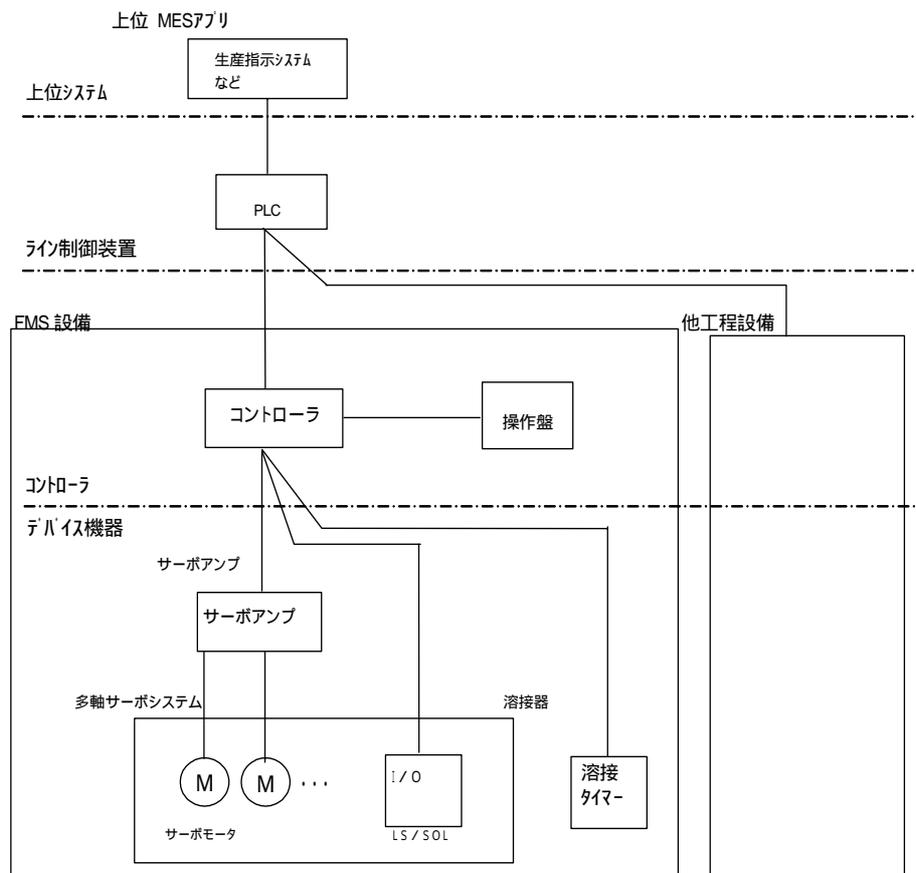


図 2 - 4 5 多軸サーボ・ロボット等で構成された専用装置

(2) 制御システムとその問題点

制御アプリケーションの設計時に設備トラブルの復旧作業を考慮すると、二次的な障害を起こさず、「障害周辺にある最低限の機器を安全に動かし」、「続きの動作を再開させる」ための制御アプリケーションを設計しなければならない。この「使いやすさ」の追求が制御システムの複雑さの一因となっている。仮に本装置のような多数のサーボモータ・ロボット等を組み合わせた設備の制御アプリケーションを一般的なラダー言語等で構築とした場合、あらゆる障害やその対処を予測して制御アプリケーションを設計すること、保全員がこの膨大なラダープログラムを読んで瞬時に理解して障害復旧すること等は不可能と考えた。

そこで、ロボットの集合体というメカ構成に着目し、ロボット群制御モデルをイメージした制御構成の定義と、それら各ロボットの動作順を定義することにより、車種特定の制御を実現するためのプログラム（ラダー）を一切書かずに設備を動かす制御システムを開発することと、それらに応じたビジュアルな設計・保全支援システムを供給することで問題を解決しようと試みた。

制御構成の定義とは、制御システムの制御対象及び、それらの構成要素の記述とその関連を記述したものの集まりであり、動作順の定義とは、各構成要素が各制御対象の動作順と、それらが

互いに干渉することを回避するための確認時期と確認事項を車種毎に順序だてて記述したものである。

具体例を挙げると、溶接前のサブパーツ部品を精度良く位置決めする「治具」の役目をするメカの塊を「位置決めマニピュレータ」と呼び、それらが固定したパネルを溶接するメカの塊を「溶接マニピュレータ」と呼んだ。これらはその機能や内部構成は異なるが、「マニピュレータ=ロボット」として定義することで全ての制御対象を「ロボット」単位で記述することができ、ロボットとロボットの相互動作を「ロボット群制御」として統一フォーマットで記述することができたのである。従って、これを操作したり保全したりする際に、種類の異なる機構を同じ操作要領で扱うことができるようになった。

当初「ロボットの集合体というメカ構成」が対象だったが、適用範囲が広がるにつれてセンサ、アクチュエータ、ロボットの集合という一般的なメカ構成に対しても、制御対象モデルを定義することによって小型の手動溶接用の治具制御、中型の自動溶接単体機、ロボットセル、メインラインと徐々に展開をしてゆき、制御のフレキシビリティに対して成果を収めてきた。

一方、こうした考え方で制御システムを構築してゆくことに課題が無いわけではない。概念的には「制御モデルに基づく構成定義、順序定義によるフレキシブルでわかりやすい制御の実現」とは言っても、現場の設計者や操作者・保全員にすればそれ自体が新たな概念であり、スキルの修得とレベルアップが必要である。もう一つは、制御システム（制御モデルを実装する実体=ハードとアプリケーション=ソフト）が完全でないうち（いわゆるバグがある状態）では、本装置も機能しないどころか、通常の制御装置と扱いが異なるために「誰にも使えない、直せない」という状態に陥りかねないリスクがある。

もちろんこれらは、モデル指向の制御システムを開発すること自体に起因する不都合ではなく、あらゆる新技術の導入による業務の改善・革新に際して起こり得ることである。

本件で扱いたい問題とは、こうした制御システムのコンセプトを、対象を広げて継続して改善・導入する際のリスクを少なくすることと、結果としてスピードアップを図ることである。

その課題は本事例のような比較的規模の大きい制御対象のモデル化を考えた場合、それに適した汎用デバイスや言語が存在しないことに起因する部分が多い。例えば PLC はその性能や点数や演算機能は大きく進歩したが、扱う対象は専ら I/O と単純変数（2 値 / 数値）に限られ、それらが集結してある機能を実現している対象や状態をモデル定義しようとしても、それらをデバイス上に表現する術がない。従って、それらは「仕様書」とか「変数割付書」、「プログラム説明書」に表し、共有せざるを得ず、それらのレベルや信頼性・共有度合い等は、設計者やユーザの技量に依存することとなる。設計者・ユーザの技量によるばらつきが発生した場合には、前述したリスクがその都度現実のものになってしまう可能性は極めて高い。

従って、コンセプトの共有にとどまらず、モデル定義やモデルの記述、実装までを共有・再利用することでリスクの顕在化と対策の先取り、スムーズな導入を果たして行きたいと考えている。この点に、UML (XML) 等モデル記述の為に標準言語やそれらに応じた実装手法（例えばオブジェクト指向設計）それらをサポートする標準制御装置の普及に期待するものである。

(3) 問題解決に対する方策案・ニーズ

- モジュール化(オブジェクト指向的アプローチ)の適用拡大
- 設備(制御システム)を表現するためのモデル化手法・記述言語の確立
- デバイス機器～上位システム間でやり取りされるデータを、「信号の変化」ではなく「意味のある情報」として扱うための定義・表現方法の確立
- モデルからの実装図面への展開方法の開発

(4) 装置エンジニアリングモデルの現状

(a) 設計・構築アクティビティオブジェクト間のコラボレーション

設計・構築フェーズのアクティビティオブジェクト間のコラボレーションを示す。

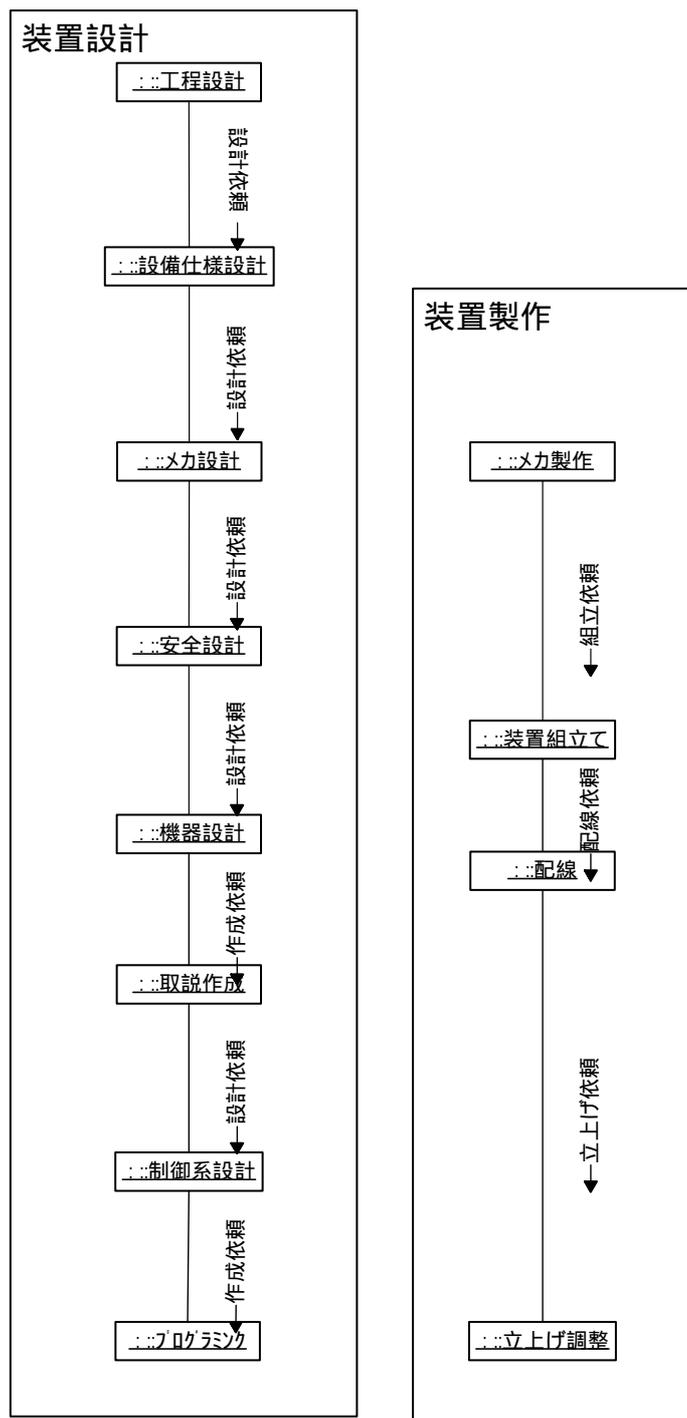


図 2 - 4 6 設計・構築フェーズのアクティブオブジェクトのコラボレーションモデル

(b) 設計・構築ドメインオブジェクトモデル

設計・構築ドメインオブジェクトを示す。

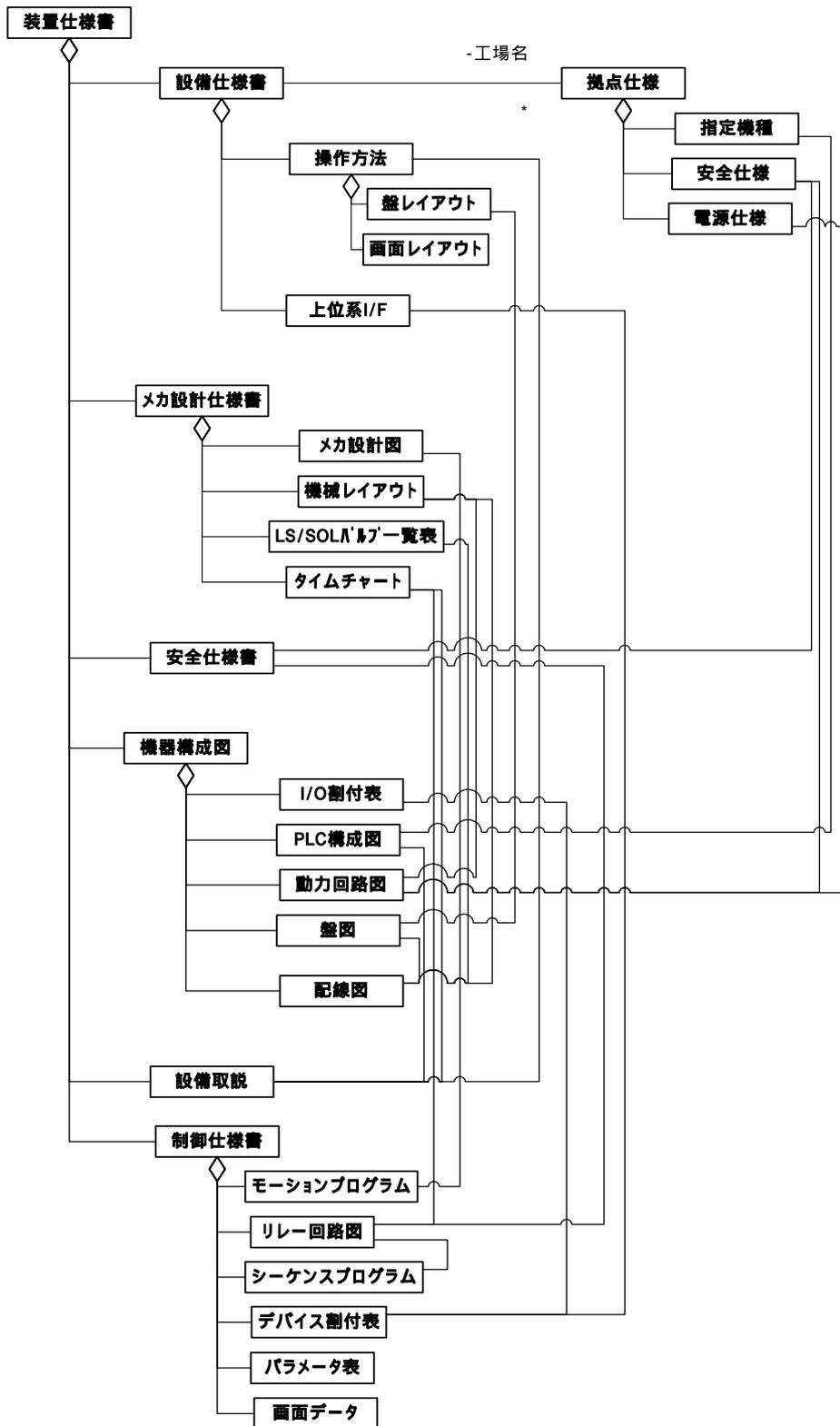


図 2 - 4 7 設計・構築ドメインオブジェクト

(c) 設計・構築アクティビティオブジェクトとドメインオブジェクト間のコラボレーション

● 装置設計

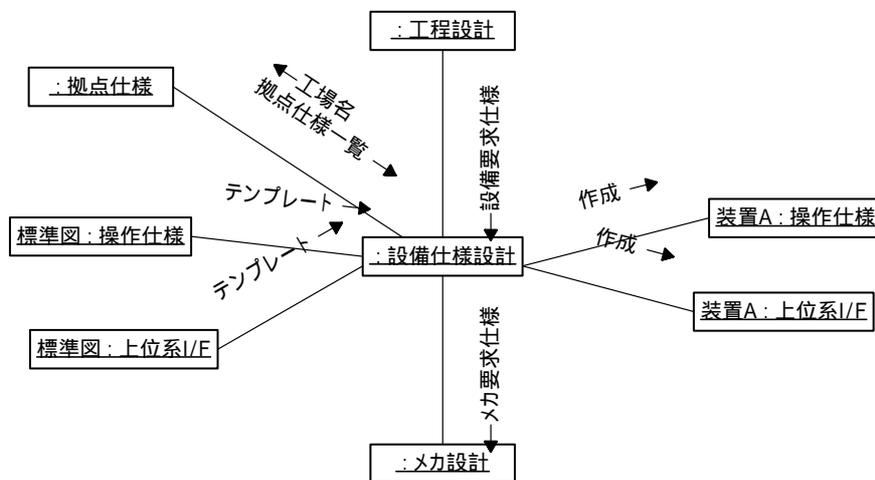


図 2 - 4 8 設備仕様設計コラボレーションモデル

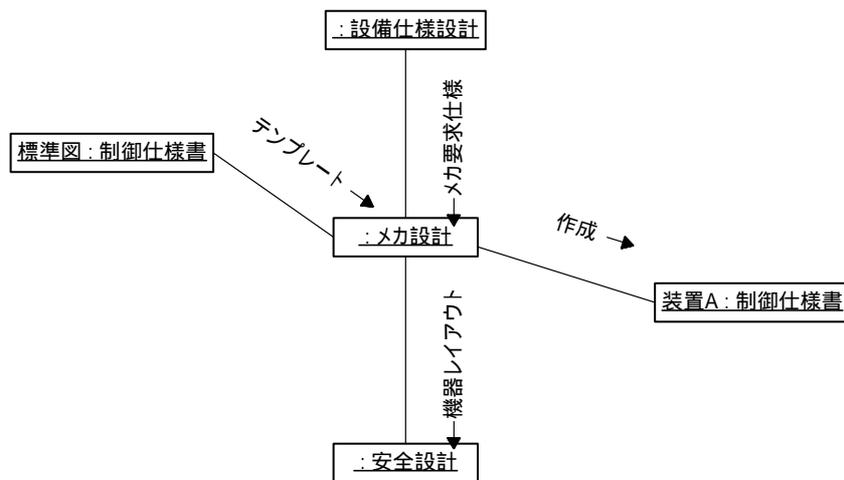


図 2 - 4 9 メカ設計コラボレーションモデル

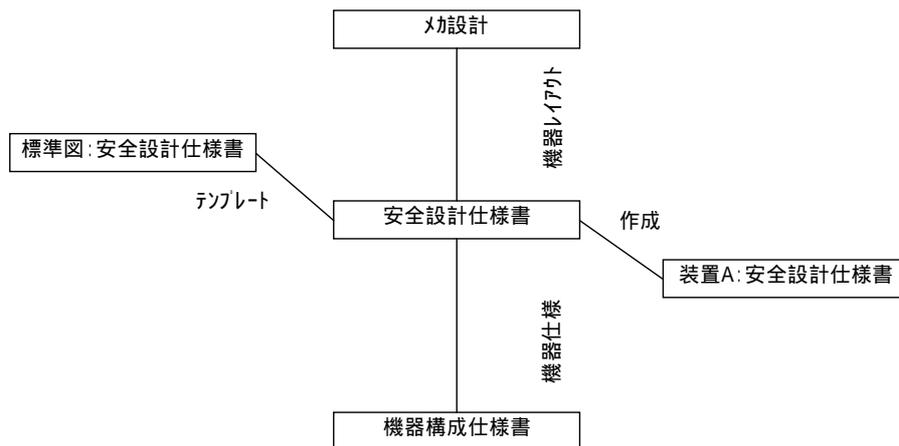


図 2 - 5 0 安全設計コラボレーションモデル

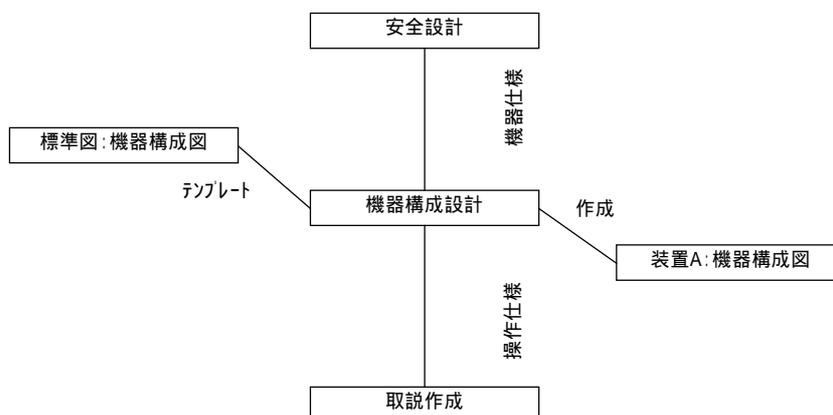


図 2 - 5 1 機器設計コラボレーションモデル

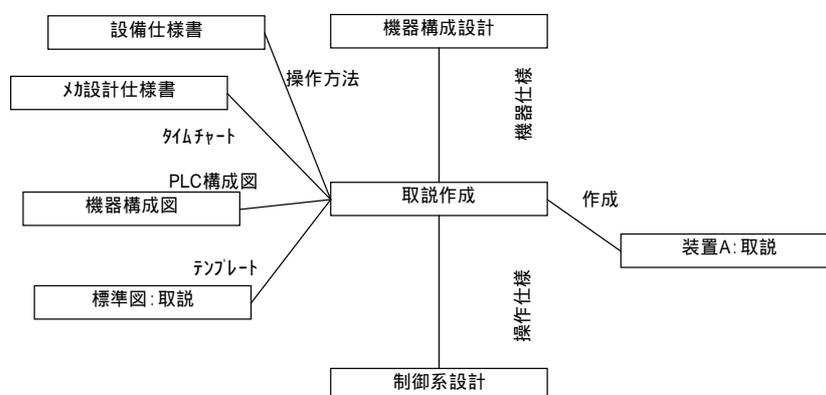


図 2 - 5 2 取説作成コラボレーションモデル

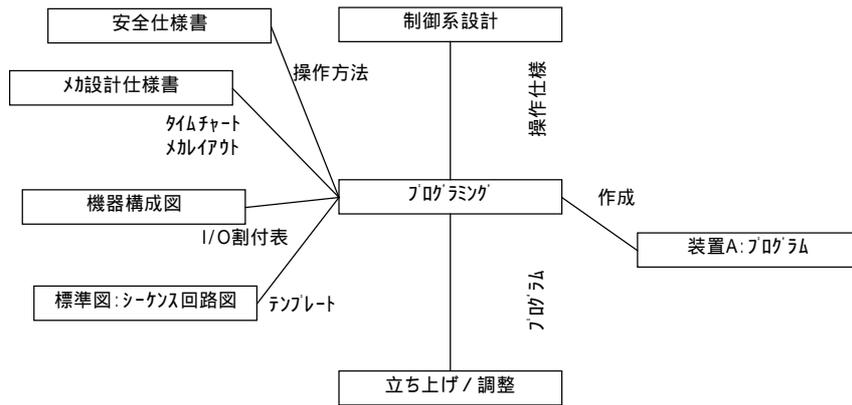


図 2 - 5 3 プログラミングコラボレーションモデル

● 装置製作

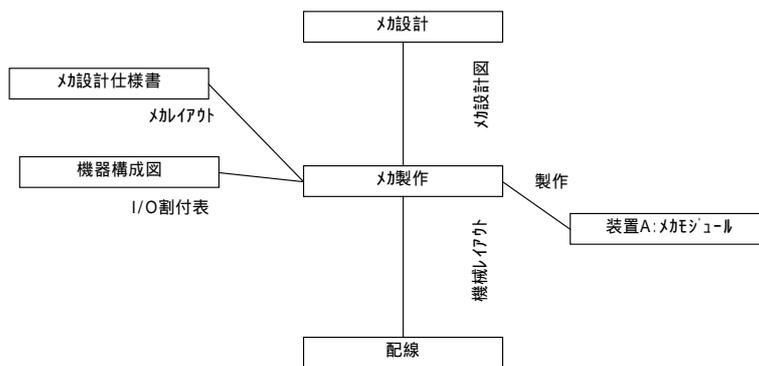


図 2 - 5 4 メカ製作コラボレーションモデル

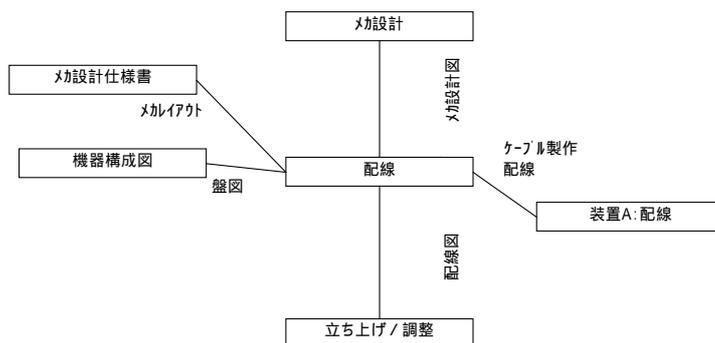


図 2 - 5 5 配線コラボレーションモデル

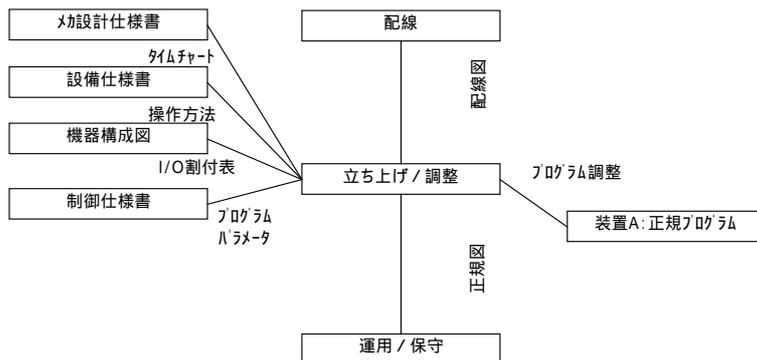
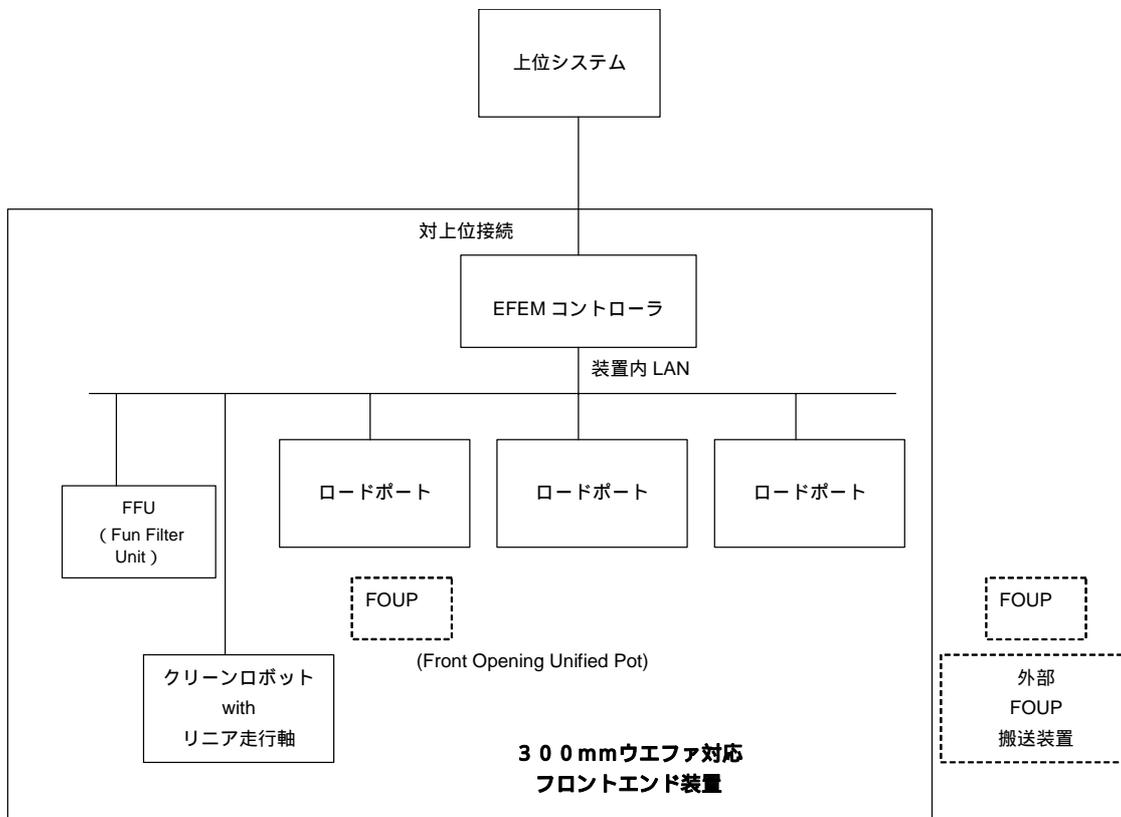


図 2 - 5 6 立上げ調整設計コラボレーションモデル

2.3.4 半導体製造

300mm ウエファ対応フロントエンド装置に関し装置モデル化の検討を行った。

本装置は下図に示す如く、300mm ウエファの 600 にもおよぶ加工プロセス対応、そのプロセスのフロントエンドとして機能する。ウエファを Front Opening Unified Pot (FOUP) 単位で、装置内複数 (2 から 4) のロードポートにて外部と受け渡しし、内蔵するリニア走行軸つきクリーンロボットにて、FOUP からプロセスへ、プロセスから FOUP へのウエファの搬送をクリーン環境で行うものである。



本装置は、上位システムより本装置がどのように見えるかの見え方、すなわち本装置のケイパビリティモデルに関しては、SEMIの装置モデルとしての汎用モデルの特化モデルとして位置づけられる。対上位インターフェースは、SEMIの装置インターフェース仕様(GEMなど)に準じて規定される装置内のデバイス・コントローラ間インターフェースはSEMIの通信仕様(SECS-、およびGEM)にて規定される。

なお、本装置の設計、製造、運用、保守(装置診断)に関するエンジニアリング対応のモデル化のニーズは今のところ顕在化していない。

2.3.5 ユーティリティ監視

環境計測(Utility監視)装置に関し、その監視の対象機器、データ取得のための通信システム、機器のモデル化などについて検討した。また、装置の要求仕様記述から、設計、製造、運用、装置診断、保守の統合エンジニアリングニーズについても現状を調査した。

(1) ニーズの高まり

省エネ法改正による、省エネ対策の必要性、エネルギー使用状況などの提出の義務が生じている。従来は製造業などの5業種に限定されていた「第一種エネルギー管理指定工場」の対象業種が全業種に拡大された。



国がエネルギー使用状況を的確に把握し、適切な対策を講じるための仕組みづくりとして、「第二種エネルギー管理指定工場」にもエネルギー使用状況などの報告書提出が義務づけられた。



特定建築物(2,000m²以上の住宅以外の建築物)の建築主に省エネルギー措置の提出が義務づけられ、また国土交通大臣から所管行政庁(建築基準法に基づく建築主事を置く市町村長など)に建築物の指導、助言に関する権限が委譲されることになった。

(http://www.mitsubishielectric.co.jp/shoene/revision/index02_b.html)

(2) 環境計測装置

本装置は生産設備（製造工場など）の環境（温度、湿度、防塵フィルター警報）使用電力量などのユーティリティ監視、および生産財としての工作機械、産業用ロボットなどの故障・稼働監視、油温、推量、水質監視を行うシステムである。

次図にシステム構成を示す。

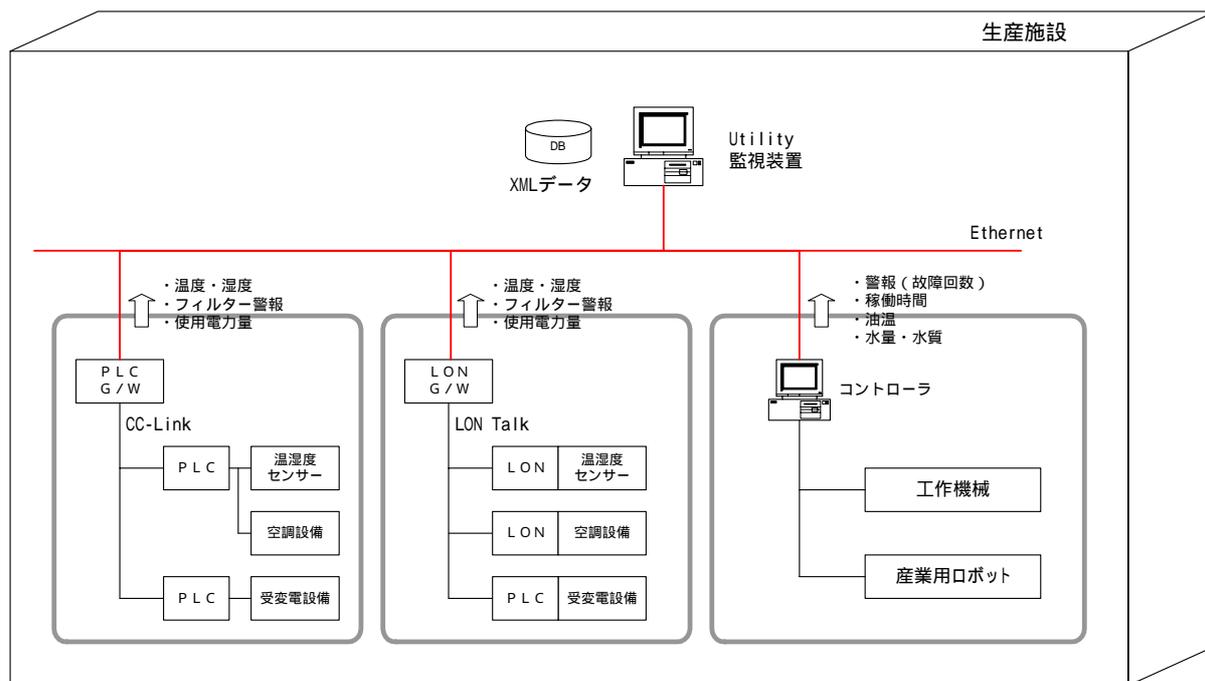


図 2 - 5 7 システム構成

(3) 本装置に関するモデル仕様、使用 LAN、通信プロファイル、関連事項

環境監視用の温湿度センサ、空調設備(温調計) 受配電設備への接続は、LON ゲートウェイ経由 LON Talk 接続か、または PLC ゲートウェイ経由 PLC 接続かのいずれかである。

表 2 - 3 LonTalk のプロトコル

| ISO レイヤ(層) | LonTalk Protocol Service | 利点 |
|---------------|--|---|
| 7 アプリケーション層 | 標準ネットワーク変数タイプ(SNVT) | ・互換性を保証 |
| 6 プレゼンテーション層 | SNVT と外部フレーム転送 | ・ネットワーク間ゲートウェイが可能 |
| 5 セッション層 | 要求/応答サービス | ・所望の処理の実行を保証 |
| 4 トランスポート層 | ・確認/非確認ユニキャスト、 マルチキャスト ・認証 ・共通順序付け、複製検出 | ・信頼性と効率 ・ネットワーク・セキュリティ ・雑音と消失メッセージのエラー除去 |
| 3 ネットワーク層 | ・アドレッシング ・ラーニング・ルータ | ・マルチメディア・ネットワーク ・拡張と再構成が簡単 |
| 2 データリンク層 | フレーミング 16ビットCRC | ・データの完全性を保証 |
| 1.5 メディアアクセス層 | ・予測 CSMA ・衝突検出とプライオリティ | ・メディアの効率的な利用 ・ネットワーク負荷の影響を除去 応答時間の保証 ・必要時に瞬時にネットワークにアクセス |
| 1 物理層 | ・ツイストペア線、電力線、無線、 同軸、赤外線、光ファイバ ・複数のデータ通信速度 | ・複数のメディアで低コスト化 |

(SNVT : <http://www.viste.com/LON/HowToUseLon.htm>)

アプリケーションとしては SNVT(Standard Network Variable Type: 標準変数タイプ: 下表)により仕様記述される。

表 2 - 4 標準ネットワーク変数タイプ (SNVT : Standard Network Variable Type)

| 名称 | Quantity | Unit(単位) | Range(範囲) | 増分 | SNVT Number |
|--------------|----------|-------------|---------------------|-------------|-------------|
| SNV_temp_p | 温度 | 摂氏 | -273.17- to +327.66 | 0.01 | 105 |
| SNV_time_sec | 経過秒 | 秒 | 0 to 6553.4 | 0.1 | 107 |
| SNV_count | イベント数 | 回 | 0 to 65000 | 1 | 8 |
| SNV_power | 電力 | W | 0 to 6553.5 | 0.1W | 27 |
| SNV_flow | 流量 | lps(リットル/秒) | 0 to 65534 | 1 liter/sec | 15 |
| SNV_press | 圧力 | 0.1kPa | -3276.8 to +3276.7 | 0.1kPa | 30 |
| SNV_str_asc | 文字列 | ACCII-CODE | 0 to 31 文字 | - | 36 |

問題は、このようなモデル化機能は LON Talk 接続に限られたものであり、PLC ゲートウェイと PLC 接続による場合はサポートされて居ないことである。工作機械、産業用ロボット接続に関しては、加工ラインシステム、機械加工ラインシステムのいずれか、またはその双方がソリューションとして流用できる、と考える。

本装置の要求仕様記述、設計、製作、運用、診断、保守関連エンジニアリングモデルの開発導入に関しては現状白紙であるがニーズはある。すなわち、工場内に導入される各種サブシステムは複数の業種のことなる業者により施工されるのが普通であり、業種間でも統一された仕様記述、アクセス手段が望ましい。

現状では LON Talk 仕様が行先しており、仕様記述面においても XML ファイル (XIF) による Device Interface 仕様記述、機能仕様についての FunctionBlock (FB) 記述を内容とする Resouce File 仕様記述が提案されている。ただし、設計、製作、運用、診断、保守に横断的に適用できる装置モデルのデファクトは未だ無い。

2.4 実装アーキテクチャの検討

ここでは実装アーキテクチャに関して述べる。ただし、実際にツールを利用する場合は、各ドメイン独自の機能内容に対応する必要がある。しかし、モデルに対する基本的な操作は全て同じであるため、ここでは共通なモデル操作をサポートするツールに注目する。

2.4.1 モデル共通操作

情報連携モデルでは、エンジニアリング時に登場するエンジニアリングモデルと標準実行モデルが存在する。これらのモデルに対して共通的な操作を以下に挙げる。

- モデル情報の作成
実体またはベース規約を基に、新規のモデル情報を作る。または、既存のモデルを組み合わせ、新たなモデルを作る場合もある。
- モデル情報の管理

作成したモデル情報を蓄え、必要な場合に検索し、情報を提供する機能。具体的には、エンジニアリング情報 DB がそれにあたる。

- インスタンス情報の作成

実際のものを表わすインスタンスをモデルから導出する機能。モデルに実体の諸情報等を付加する必要がある。

- インスタンス情報の管理

インスタンスおよびインスタンス情報で記述されたもののロケーション等の管理。

- インスタンスの利用

インスタンスを組み合わせて利用する。

2.4.2 モデル共通操作ツール

共通操作に対する想定されるツールを下表に示す。

表 2 - 5 共通操作に対する想定されるツール

| 区分 | 項目 | 内容・備考 |
|-------------|-----------------|-------------------------------------|
| エンジニアリングツール | モデル情報作成ツール | モデルを記述する XML を作成する |
| | モデル情報組み合わせツール | モデルを記述している XML を組み合わせて新たな XML を作成する |
| | インスタンスデータ作成ツール | モデルを継承し、インスタンスを表現する。 |
| | 接続構成設定ツール | インスタンスの組み合わせ構成を記述する |
| サーバ側サービス | リポジトリ管理サービス | モデル情報の管理を行う機能の性質を管理する機能 |
| | 名前・ロケーション管理サービス | オブジェクトの仮名やロケーションを管理する機能 |
| ランタイムソフトウェア | ゲートウェイ | MES アプリケーションや装置などの機能間を接続する機能 |

2.4.3 ツール開発における優先順位

表 2 - 5 に示したツール類のうちで、代替ソフトウェアに関しての可能性を検討した結果を表 2 - 6 に載せる。エンジニアリングツールに類するものは、XML で記述されたモデル情報を変形させる処理である。モデルを操作するツールがあれば便利ではあるが、いわゆる XML エディタでも実現可能である。そのため、ツールとしては、サーバ側サービスとランタイムソフトウェアに分類される、3 ツールが重要であると考えられる。さらに、サーバ側サービスのうちで、リポジトリ管理サービスはデータベースに、名前・ロケーション管理サービスは LDAP などのネームサービスにマッピングできると考えられる。そのため、ゲートウェイが最も重要度の高いツールであると考えられる。

表 2 - 6 代替ソフトウェアに関する可能性

| 区分 | 項目 | 代替案 |
|-----------------|-----------------|--------------------|
| エンジニアリング ツール | モデル情報作成ツール | XML エディタ |
| | モデル情報組み合わせツール | XML エディタ |
| | インスタンスデータ作成ツール | XML エディタ |
| | 接続構成設定ツール | XML エディタ |
| サーバ側サービス | リポジトリ管理サービス | データベースをベースとして開発可能 |
| | 名前・ロケーション管理サービス | ネームサービスをベースとして開発可能 |
| ランタイムソフト ウェア | ゲートウェイ | スクラッチからの開発が必要 |

2.4.4 ゲートウェイ

この節では、前節の検討においてツール類のうちで最も重要であると結論付けたゲートウェイに関して、その要求機能およびアーキテクチャに関して述べ、仕様として決定すべき項目を明確にする。

(1) ゲートウェイに対する要求

ゲートウェイに必要となる機能を、以下に挙げる。また、このような機能を実現したゲートウェイの例を図 2 - 5 8 に示す。

- 接続機器モデル非依存
ゲートウェイによって接続する対象モデル(MES アプリケーションのモデル、機器モデル)がどのようなものであっても接続可能な情報モデルを持つこと
- モデルの諸情報の管理
ゲートウェイミドルウェアの情報モデルの名前管理、データ型、情報ソースの管理ができること(例: ORiN の特定機器用 RAO モデルのある特定の変数の管理機能など)
- 情報の相互参照
ゲートウェイミドルウェアの情報モデル間で情報の参照が可能であること
- 時間同期機能
ゲートウェイミドルウェアの情報モデル内で MES アプリケーションと機器の情報を時間的に同期させることが可能であること
- 情報交換処理
ゲートウェイミドルウェアの情報モデル(変数個別)毎に、型変換、マスキング、BCD 変換、演算式等が可能
- 情報交換タイミング指定
ゲートウェイミドルウェアの情報モデル(変数個別)毎に、機器側から周期的に情報収集を行うかどうか、またその周期はなどを設定可能な機能があること

- DB 書き出し機能

ゲートウェイミドルウェアの情報モデル(変数個別)毎に、BD への書き込みを行うかどうか、またその条件やトリガ等設定可能

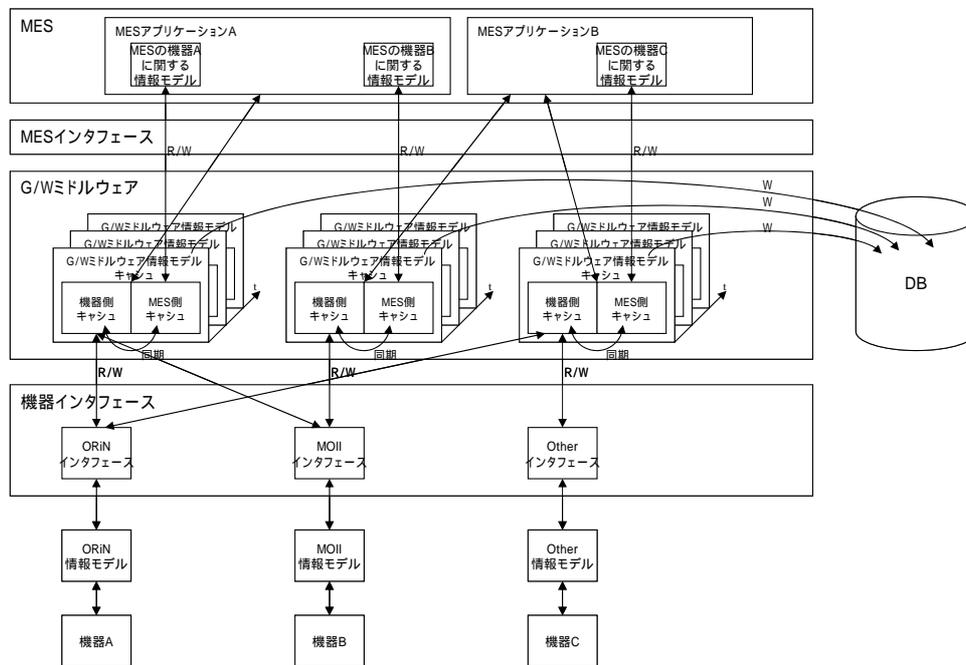


図 2 - 5 8 ゲートウェイ実装例

(2) ゲートウェイのアーキテクチャ概要

(a) 全体アーキテクチャ

図 2 - 5 9 にゲートウェイの実装イメージを示す。図 2 - 5 9 中の a) は、前述の MES と装置間のデータ変換部の構成を、変換機能も含めて示したものである。さらに、b) および c) に示すように、MES アプリケーション間や装置間でも同じ機構で接続できるようなアーキテクチャをとるべきである。これによりゲートウェイは、モデル連携におけるさまざまなインスタンスを、上下の区別なく接続する機能を持つことになる。

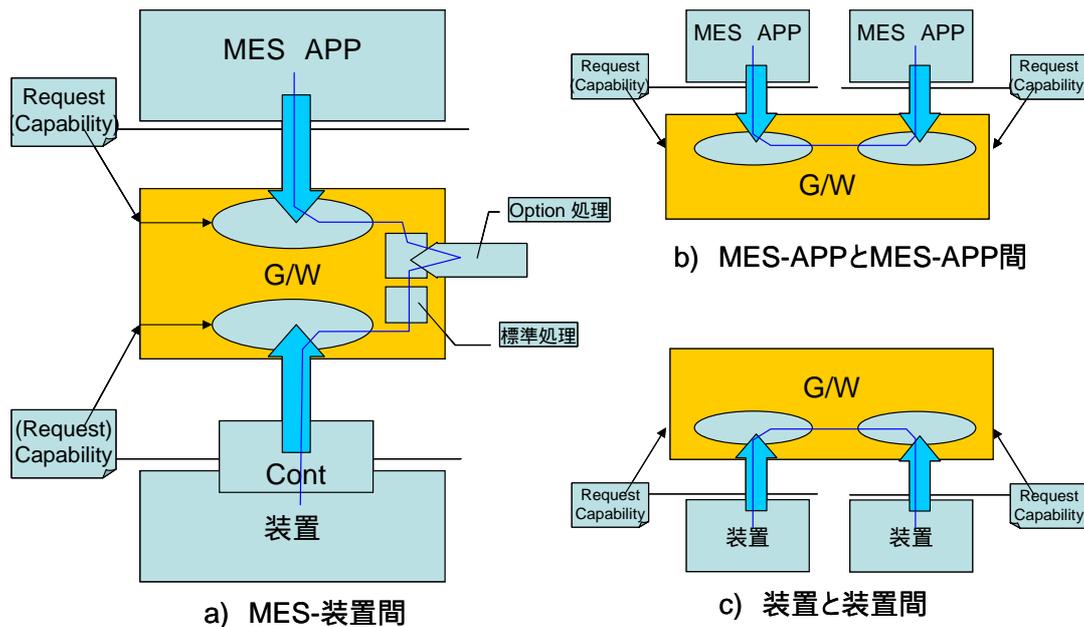


図 2 - 5 9 ゲートウェイのイメージ

(b) インタフェースアーキテクチャ

MES と装置間のような場合の実際のゲートウェイ周辺の接続方式概要に関して図 2 - 6 0 に載せる。装置や MES アプリケーションとゲートウェイの実際の接続には、何らかの通信ライブラリが使用される。ここでは、実際のソフトウェア構成上の実装内容に関して述べる。

● MES アプリケーションとゲートウェイ間のインタフェース

一般に、MES アプリケーションなどのソフトウェアから機能呼び出す場合、アプリケーションから目的の機能ライブラリを呼び出す方式で行う。そのため、基本的なゲートウェイのアクセス方式と同じアーキテクチャを実装時にも用いることができる。

● 装置とゲートウェイ間のインタフェース

一般に、さまざまなソフトウェアからコントローラをアクセスするため、インタフェース用のライブラリやドライバ類がコントローラベンダより配布（有償無償は問わず）されている。装置とゲートウェイ間の情報交換では、一般に、このようなソフトウェアを利用して構築される。しかし、これらのライブラリは、アプリケーションから装置をアクセスすることが主であるため、一般に装置側からアクセスをしていくインタフェース方式はとられていない。

そのため、ドライバのような何らかのソフトウェアをゲートウェイに搭載し、ソフトウェアベンダーの用いるコントローラアクセスライブラリを利用しつつ、ゲートウェイには MES アプリケーションとゲートウェイとのインタフェースと同様なインタフェース方式を実現できる仕様とする必要がある。

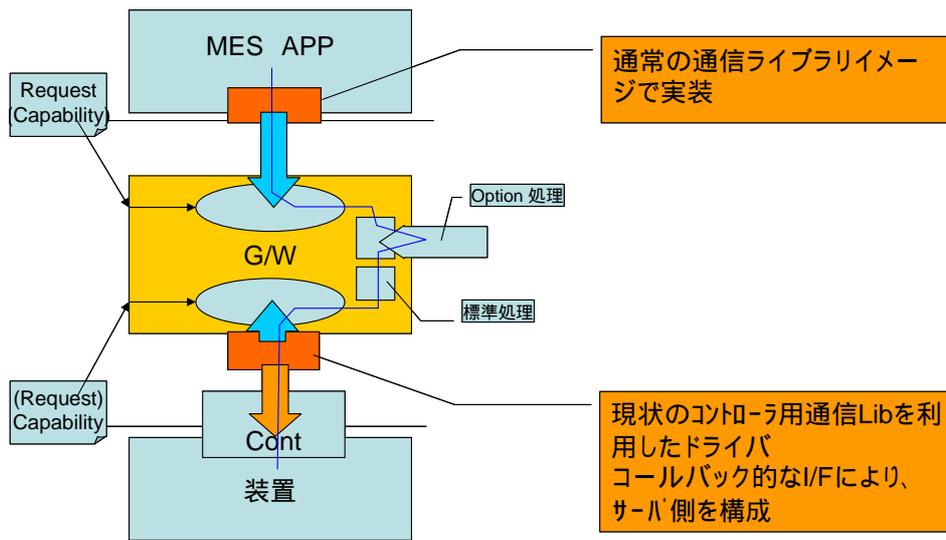


図 2 - 6 0 実装イメージ

(c) ゲートウェイと各ワーキンググループアウトプット

ゲートウェイと各 WG との関係を図 2 - 6 1 に示す。ゲートウェイおよび全体アーキテクチャ関連は D2 にて検討されている。しかし、D1 および D3 もアウトプットがゲートウェイに反映される必要がある。

● D1

MES アプリケーションが機器からの情報を入手あるいは指示を与える場合の要求事項をまとめ、それをデータ化する方法論を取り込む。また、個別・特定の MES アプリケーションの要求仕様を明確にし、それを要求仕様データとして作成する。ただし、MES アプリケーション間の情報交換の際は、MES アプリケーションも呼び出される側となることもあるため、場合によってはケイパビリティを持っている。

● D2

ゲートウェイの運用時およびそのエンジニアリング時の必要機能を明確にする。また、ゲートウェイが稼動するための全体アーキテクチャを明確にする。

● D3

装置のケイパビリティの表記方法を明確にし、また、個別・特定装置のケイパビリティをデータ化する。ただし、装置間通信などを考慮すると、装置によっては要求を持っているものもありうる。

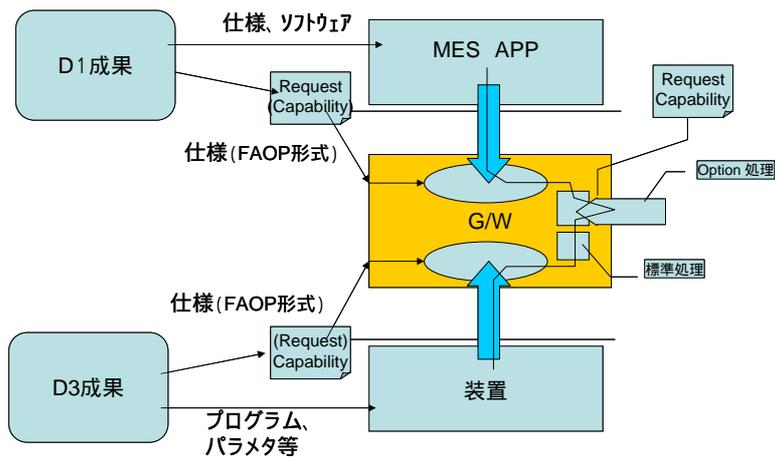


図 2 - 6 1 ゲートウェイと D2、D3 との関係

(3) ゲートウェイ仕様策定内容

図 2 - 6 2 に、ゲートウェイとして仕様策定の必要な箇所を示す。図中の実線の円は、抽象仕様の必要な箇所、点線の円は、実装仕様の必要な箇所である。また、各部分の概要に関して、表 2 - 7 に示す。

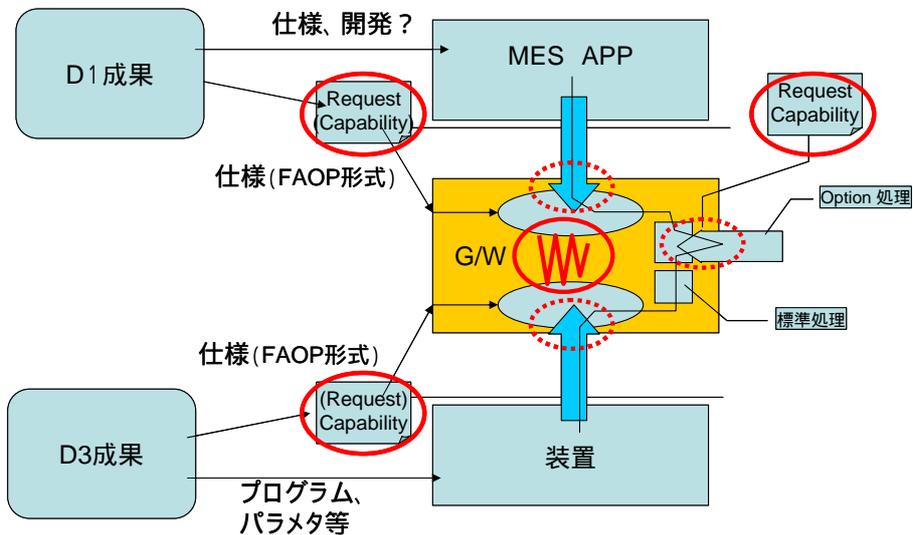


図 2 - 6 2 ゲートウェイとして仕様策定の必要な箇所

表 2 - 7 ゲートウェイの仕様策定項目

| 区分 | 項目 | 内容 |
|--------|-----------------------------------|---|
| 抽象仕様規約 | ゲートウェイアーキテクチャ | 通信パターンやプラグインなどのゲートウェイの全体アーキテクチャ。 |
| | 要求/ケイパビリティスキーマ | ゲートウェイによって接続するオブジェクトの要求やケイパビリティなどのインタフェース仕様。インスタンス仕様は分野別となる。 |
| | 設備構成仕様スキーマ | ゲートウェイを用いて MES アプリケーションや装置を接続するための構成仕様記述ルール |
| | 変換処理ケイパビリティスキーマ | 変換処理のインタフェース仕様。 |
| | 既存規約へのマッピング規約 | ORiN, MOII 等の既存規約のプロトコルからインタフェース仕様を生成する方法。 |
| 実装仕様規約 | ドライバ仕様 | ゲートウェイとオブジェクトを接続するソフトウェア仕様。管理仕様やインストール情報なども含む |
| | オプション処理機能 Lib 仕様、管理仕様 | オプションの変換処理のソフトウェア仕様およびゲートウェイへのプラグイン仕様。 |
| | 表示部品仕様 | ゲートウェイのエンジニアリング環境上でビジュアルイメージを利用する際のインタフェース仕様（ビジュアルなエンジニアリング環境が存在する場合のみ） |
| | ゲートウェイ用エンジニアリング環境のプロパティへのマッピングルール | エンジニアリング環境において要求/ケイパビリティスキーマから設定用のボックスなどにマッピングする際の仕様 |
| 実装仕様 | ゲートウェイツール実装仕様 | ソフトウェア独自の仕様。標準化する必要はない。 |

(4) ゲートウェイのエンジニアリング

ゲートウェイは、実行時のイメージが強調されやすいが、そのコンフィギュレーションは非常に複雑であるため、実際に運用するにはエンジニアリング環境が必須となる。ゲートウェイとそのエンジニアリング環境により設定される項目の位置づけを図 2 - 63 に示す。さらに、表 2 - 8 に概要を挙げる。実際のシステムではさらに多くなると予想される。

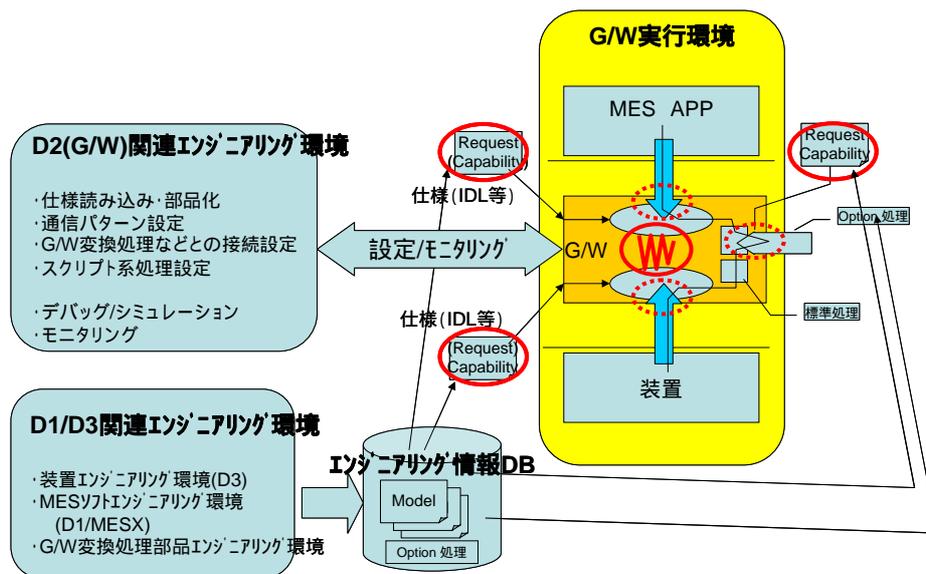


図 2 - 6 3 ゲートウェイとそのエンジニアリング環境により設定される項目の位置づけ

表 2 - 8 ゲートウェイのエンジニアリング環境設定項目例

| | 内容 |
|----------------|----------------------------|
| 接続するオブジェクトの指定 | 情報を交換するMESアプリケーションや装置などの指定 |
| 接続方法 | 情報交換の際に使用するドライバソフトウェアなど |
| 変換処理シーケンス | 変換処理および変換シーケンスの設定 |
| 通信パターン | 使用する通信パターンの指定 |
| オプション変換処理用機能設定 | 個別仕様として設定項目があった際の設定 |
| ライブラリ管理 | オブジェクト情報などのライブラリ化 |

(a) ビジュアル環境

図 2 - 6 4 に、ゲートウェイ用エンジニアリング環境のイメージを載せる。この例では、画面左にオブジェクト類のライブラリ、右上側にゲートウェイのコンフィギュレーション、右下側に時間イメージでの通信パターンを設定する領域があることを示している。実際のエンジニアリング環境はさらに複雑になるが、画面構成などの仕様は本委員会において検討・策定する範疇ではないため、ソフトウェアベンダーに任せられる。このエンジニアリング環境で構築されたコンフィギュレーション情報は、ネーム管理機構によって認識された実環境上のオブジェクトに設定または通信対象として認識される。また、これらの設定項目類は、リポジトリ管理機構によって管理される。

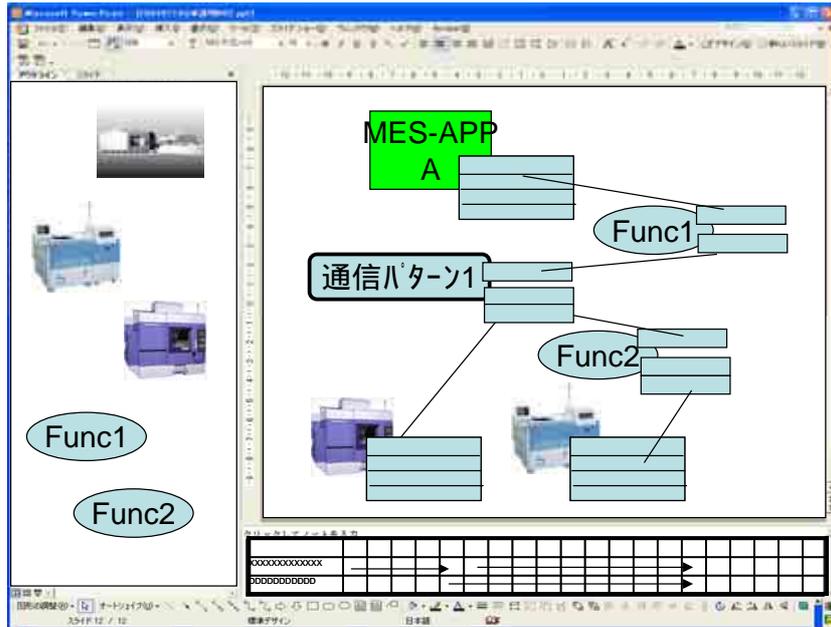


図 2 - 6 4 ゲートウェイ用エンジニアリング環境のイメージ

第3章 製造業における製造システム技術基盤構築に関する調査研究

本章は、FAオープン推進協議会（FAOP）/ネットワークを活用したものづくりサービス専門委員会（FAOP-RFM、[Remote Factory Management]）の協力の下、同委員会で進めている製造システムにおけるXML技術を活用したオープンな情報交換仕様の内容および取り組みについてまとめたものである。

3.1 CADコンテンツ流通のための共通基盤構築

商品リードタイムの短縮、効率的な調達、顧客情報の管理などを目的としたIT活用が行われているが、その実態は中小製造業においては大企業との間に情報格差が益々拡大している。その実態を東京都大田区の調査（平成13年度）を基に、平成12年度工業統計表から99人以下の企業について全国規模で算出すると以下ようになる。

なお、対象とした製造分野は、金属製品製造業、一般機械器具製造業、電気機械器具製造業、輸送用機械器具製造業、精密機械器具製造業とした。

| | | | |
|-------|-------|-------|----------|
| 設計担当者 | 選任 | 39.8% | 64,000社 |
| | 兼任 | 26.9% | 56,000社 |
| | いない | 42.3% | 90,000社 |
| CAD導入 | している | 50.8% | 107,000社 |
| | していない | 49.2% | 103,000社 |
| 引合、見積 | 電子データ | 24.1% | 51,000社 |
| | その他 | 75.9% | 159,000社 |

（その他は、手渡し、郵送、ファクシミリ）

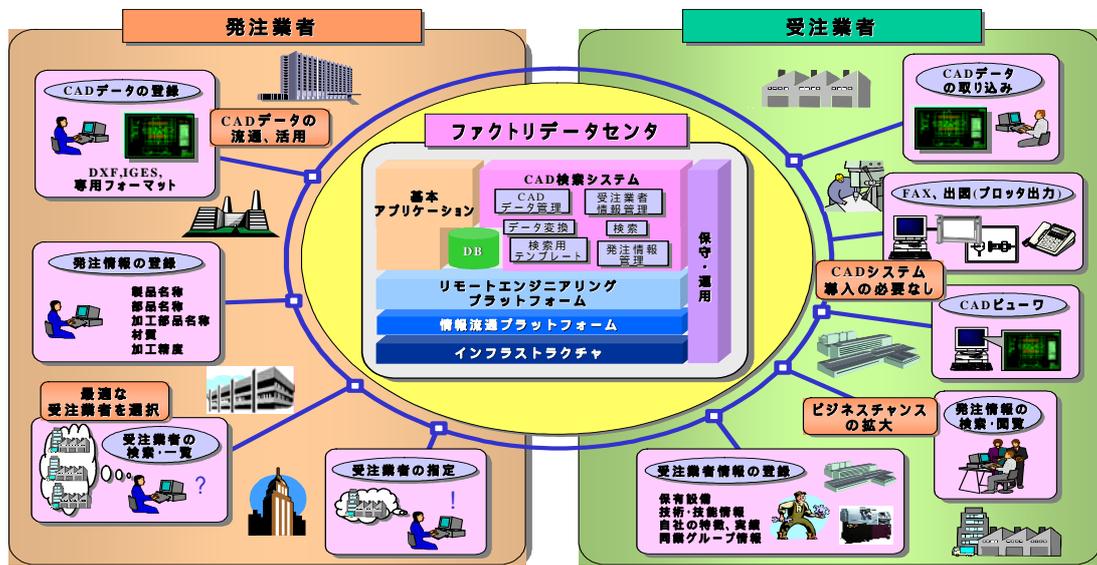
国内において多品種少量、高付加価値の生産体制を実現するには、製品開発、製品製造におけるCAD/CAM/CAE（ITを活用した製品設計、試作、製造）の導入は、設計や製造作業を効率化させリードタイムを短縮するものである。

一方、従来のインターネット活用の商取引には、「期待するものづくりが可能な製造業を検索する」という目的に対応した情報が登録されていないため、絞込みがむずかしく検索結果の信頼度が低いという問題があった。このため、ものづくりの受発注の特性である「技術」に着目し、受発注の相手企業を見つけるという検索方法を用いる事で、受発注者双方の負担を軽減し、リードタイムを大幅に短縮し、製造業の事業手法の革新を図るものである。

このため、従来の3次元CADデータに、技能、技術、製造機械と加工内容、材質、精度等の「技術」キーワードを追加した3次元CADメタデータをコンテンツとする、3次元CADデータの変換及び表示機能を有する検索システムを作成する。

また、この検索システムには、上記調査の引き合い、見積のために設計図を「手渡し、郵送、

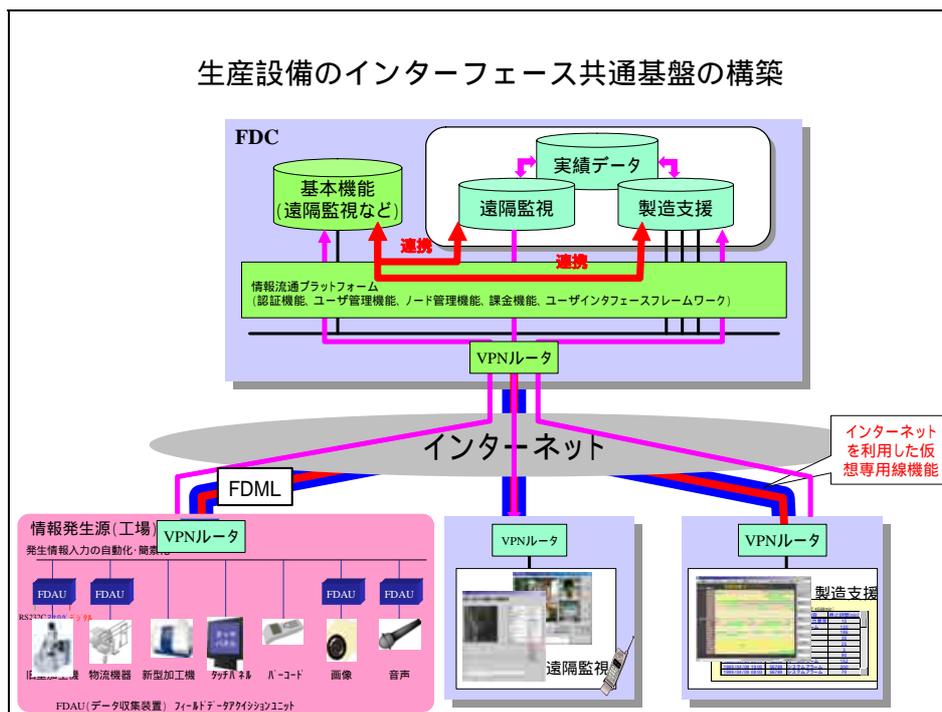
「ファクシミリ」で授受している実態は、知的財産が模倣される危険にあり、ネットワークの安全性を強化した高度な通信技術を装備する必要がある。



3.2 生産設備のインタフェース共通基盤の確立

これまでの FA 業界は機器提供メーカーがそれぞれに装置、制御方法、ネットワーク規約を作り出してきた。そのため LAN 等 IT 関係の設備設置者は投資を行う度に従来装置との運動性に気を使い、設置コストも増大した。体系化し共通なインタフェースを確立する事で、生産管理の効率化、多品種少量需要に対応する短納期生産に供する効率的な事業手法の確立、設備のリモートメンテナンス等設備総合効率化、CO2 等の削減と環境改善等につながる。

したがって、近年、情報システム間のインタフェース仕様として導入が進んでいる XML を活用し、リモート・ファクトリ・マネージメント(遠隔監視、遠隔制御、環境 3R 及び LCA 等)を実現できる製造業のユビキタス・マニュファクチャリングに対応したインタフェースの共通基盤を確立することである。

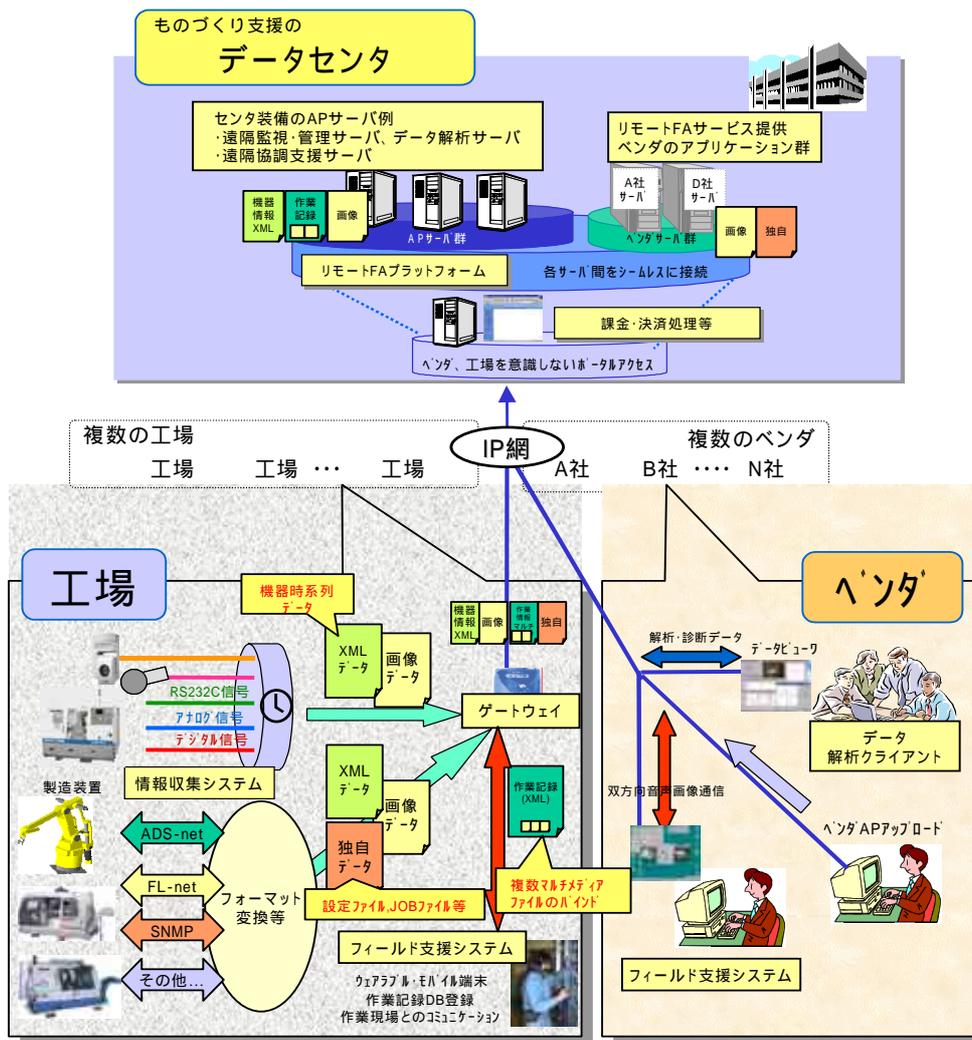


3.3 遠隔作業支援システムのための技術基盤構築

熟練者や技能者不足が深刻化していく産業界の現状から、効率の良い運用や保守を達成するために通信ネットワークとマルチメディアを活用した情報化が必要とされている。また、従来からの生産システムの完全自動化に向けた考え方から、人と機械が相互に補完し合う形での運用を実現する技術も必要である。

今後、情報インフラストラクチャの進展とともに、コスト削減、消費地域での生産への移行などにより、国外での生産等、企業のグローバル化も進展してきている。そこでは場所、時間を越えてのシステムの操作や管理、診断が必要不可欠となってきた。

このため、機械産業における生産技術に携わる人々に対して、マルチメディアに代表される新しい情報分野の技術を応用し、工場などの運用管理(機械の操作・監視、診断・保守、協調作業、技術者教育)を行う遠隔作業支援システムのための技術基盤を開発することである。

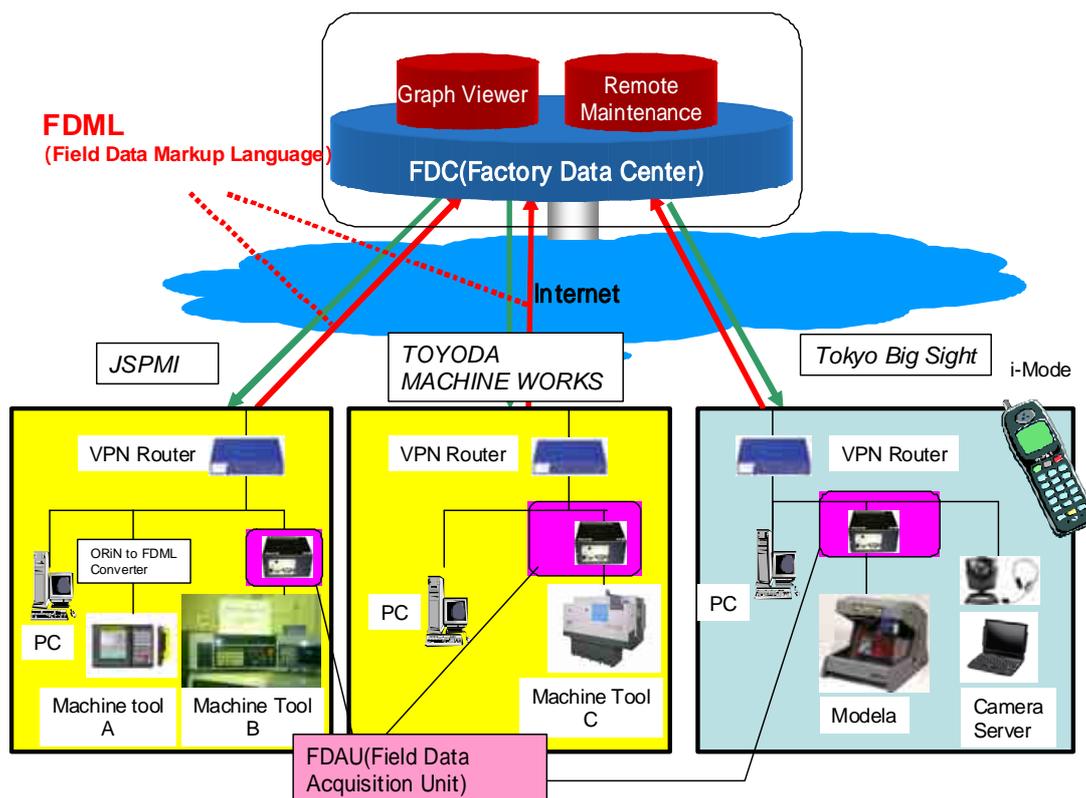


3.4 実証実験

製造現場および製造現場を模擬した製造環境とファクトリ・データセンタを接続する実験環境を構築した。実験環境の模擬図を以下に示す。

具体的には、機械振興協会技術研究所、豊田工機およびシステムコントロールフェア 2003 会場(2003/11/11~14、東京ビッグサイト)に工作機械を設置し、各工作機械より取得された情報をものづくり関連情報の連携に最適な XML ベースの FDML 形式に変換し、ファクトリ・データセンタへ連続的に送信する。蓄積した情報の閲覧は、システムコントロールフェア 2003 会場に設置された PC 上で一括的に行うものとし、送信された情報を基に定期的にグラフ表示を行うと共に、非定期的に発生する模擬的な異常の状況を確認した。

実験の結果、インターネットを經由し 7 時間×4 日間(1分~10分間隔)の送信情報を漏れなくファクトリ・データセンタで管理できること、定期的なグラフ表示と非定期的な異常発生検知が可能であること、異常発生時の静止画像(工作機械前面パネルを撮影)を同時に閲覧することで異常原因究明に要する時間を大幅に短縮することが可能となることを確認した。



この実証実験により、情報発生源のデータ収集での FDML の活用は、データ収集の一元化と、データセンタに対するデータベースの共通化等が実証された。

これらの事から情報発生源すなわち、生産設備機器のインタフェース共通基盤に FDML (XML) ならではの恩恵を享受するためには、それをういた開発や標準化活動に際し、技術的な特徴を熟知した上で安価でユーザの求め易い機器の開発市場への投入等の取り組みが重要となることが本調査から結論づけられた。

KEIRIN



この事業は、競輪の補助金を受けて実施したものです。

禁無断転載

平成15年度 製造業における情報技術
活用促進補助事業調査報告書

発行 平成16年3月

発行者 財団法人 製造科学技術センター
〒105-0002
東京都港区愛宕一丁目2番2号
電話 03-5472-2561