

博士論文

柔軟な異種システム連携を実現する  
ためのアーキテクチャに関する研究

山本 秀典



## 目次

1. 序論	1
1.1. 研究の動機	1
1.2. 対象とする領域	1
1.3. 本研究の目的	3
1.4. 本論文の構成	4
2. 従来研究・関連研究の動向	5
2.1. System of Systems (SoS)	5
2.1.1. SoS の概要	5
2.1.2. SoS の潮流	6
2.2. Cyber Physical System (CPS)	7
2.2.1. CPS の概要	7
2.2.2. CPS の潮流	9
2.3. 従来研究・関連研究との比較	10
2.3.1. 本研究の位置付け	10
2.3.2. 従来研究・関連研究での限界	11
2.4. 本研究のアプローチ	13
3. 異種システム連携技術	15
3.1. 異種システム連携のコンセプト	15
3.1.1. 概要	15
3.1.2. サービス	16
3.1.3. システムビュー	19
3.1.4. インフラシステム	20
3.2. 動的グループ形成・制御	22
3.2.1. グループの定義と動作原理	22
3.2.2. グループの制御・ライフサイクル管理	28
3.3. 異種ミドルウェア連携	34
3.3.1. 異種ミドルウェア連携の位置付け	34
3.3.2. システムの異種性	35
3.3.3. 規格間相互接続	36
3.3.4. 仲介	38
3.4. 必要な要素技術	41
3.4.1. 機能要件対応	42
3.4.2. 非機能要件対応	42
3.5. システム要件	45
3.6. 異種システム連携の境界条件	46

3.6.1.	境界条件の定義	46
3.6.2.	境界条件の判定例	48
3.7.	システムリファレンスアーキテクチャ	49
4.	システムの構築	51
4.1.	位置付け	51
4.2.	想定するソフトウェア構成	51
4.3.	想定するハードウェア構成	52
4.4.	想定する機能配置	53
4.5.	実現すべき機能一覧	53
4.6.	実現方式	56
4.6.1.	動的グループ形成・制御	56
4.6.2.	異種ミドルウェア連携	67
4.7.	システムの実現	77
4.7.1.	システム連携パターン	77
4.7.2.	システム構成パターン	78
4.8.	実現方式の評価	82
4.8.1.	実現方式の有用性の定量評価	82
4.8.2.	実現方式の有用性の定性評価	85
5.	実問題における適用例と評価	90
5.1.	社会インフラ(電力): 電力需給調整・スマートグリッド	90
5.1.1.	概要	90
5.1.2.	本研究の適用方法	95
5.1.3.	ユースケース適用	99
5.1.4.	評価	108
5.2.	航空宇宙: 惑星探査	109
5.2.1.	概要	109
5.2.2.	本研究の適用方法	111
5.2.3.	ユースケース適用	114
5.2.4.	換装・新旧入替	120
5.2.5.	評価	123
6.	考察	125
6.1.	本研究の提案技術の適用により期待できる効果	125
6.1.1.	社会インフラ分野における効果	125
6.1.2.	航空宇宙分野における効果	125
6.2.	本研究の提案技術の応用可能性	126
6.2.1.	社会インフラ分野における応用可能性	126

6.2.2.	航空宇宙分野における応用可能性 .....	126
6.2.3.	他分野への応用可能性 .....	127
7.	結論 .....	129
7.1.	研究のまとめ .....	129
7.2.	今後の課題 .....	131
	謝辞 .....	133
	参考文献 .....	134
	付録. 関連研究発表文献リスト .....	141

# 1. 序論

## 1.1. 研究の動機

近年の世界的な経済発展と環境保護の高まり、震災復興の経験により、社会インフラシステムには、安全、安定、さらなる利便性向上だけでなく、再生可能エネルギーの積極利用、有事であってもサービスが滞らないこと、災害発生時のインフラサービスの早期復旧等が強く要求されてきている。特に昨今、低炭素化社会の実現に向けた再生可能エネルギーや Electric Vehicle (EV) 等の大量導入を含め、地域で効率的にエネルギーの地産地消を行う地域エネルギーマネジメント等の、「地球環境の保全」と「安心・便利で豊かな都市生活」を両立させる新しい都市の在り方や都市づくりへの取り組みがグローバルで進められている<sup>38)39)</sup>。

また従来からの電力、交通、等の社会インフラの分野及び鉄鋼、一般産業等の分野においては、性能及び信頼性を維持したまま、低コストで、かつビジネス環境の急激な変化や顧客の要求変化に柔軟に即応できる情報制御システム<sup>83)84)</sup>が強く求められ続けている。

上述のようなこれからの社会インフラを支えるシステムの要件として、以下の3つが挙げられる。(1)様々な需要家やサービス事業者のインフラ供給への参入、新サービスの提供、供給と需要の関係の変動等の様々な変化に柔軟に対応できる、多種多様性、(2)従来からの既存設備だけでなく、様々な需要家やサービス事業者の大量の機器やアプリケーションが様々なネットワークを通じて社会インフラに加わった場合、災害発生等が原因でシステムの一部が動作不能となった場合でも安定的なインフラ稼働とサービス提供を実現する、信頼性、(3)社会インフラの成長や長期維持に対応した、拡張性、である。

上述のような要件を満たすためには、異種のシステム間で柔軟に連携することが必要となる。ここで異種システムとは、準拠規格やプロトコルの異なるシステム、業務の異なるシステム、制御系と情報系のような処理周期・タイミングの異なるシステム、業種や事業者の異なる異主体システムのことを指す。

なお社会インフラシステムは既に各々独立して構築されて稼働しているシステムであり、特に各インフラの制御系の性能、信頼性を維持することは不可欠である。

以上の事柄を踏まえて本論文では、社会インフラシステムの信頼性を保証し、どのような状況においてもサービス提供を維持させるために、目的、導入時期等の異なる異種のシステム同士を、各システム的设计当初の想定・前提に関わらず、状況に応じて、柔軟に連携させるためのシステムアーキテクチャ及び基盤技術としての異種システム連携技術について提案することを目的とする。

## 1.2. 対象とする領域

本研究が対象とする社会インフラシステムとして、まずは主として電力及びスマートグリッドの分野を扱う。また航空宇宙の分野についても扱う。なお本研究にて提案するシス

システムアーキテクチャ及び基盤技術は、他の社会インフラである鉄道、水道、ガス等についても同様に適用可能となるように拡張性を考慮する。

社会インフラシステムの概要例を図 1-1 示す。電力（配電）と鉄道（運行管理）の例を示す。図 1-1 に示すように、フィールドにある設備、設備を稼働させて社会インフラとしてのサービスを提供するための業務と、これらの業務と設備をつないで円滑に運用させるための Information and Communication Technology (ICT) システムから成る。

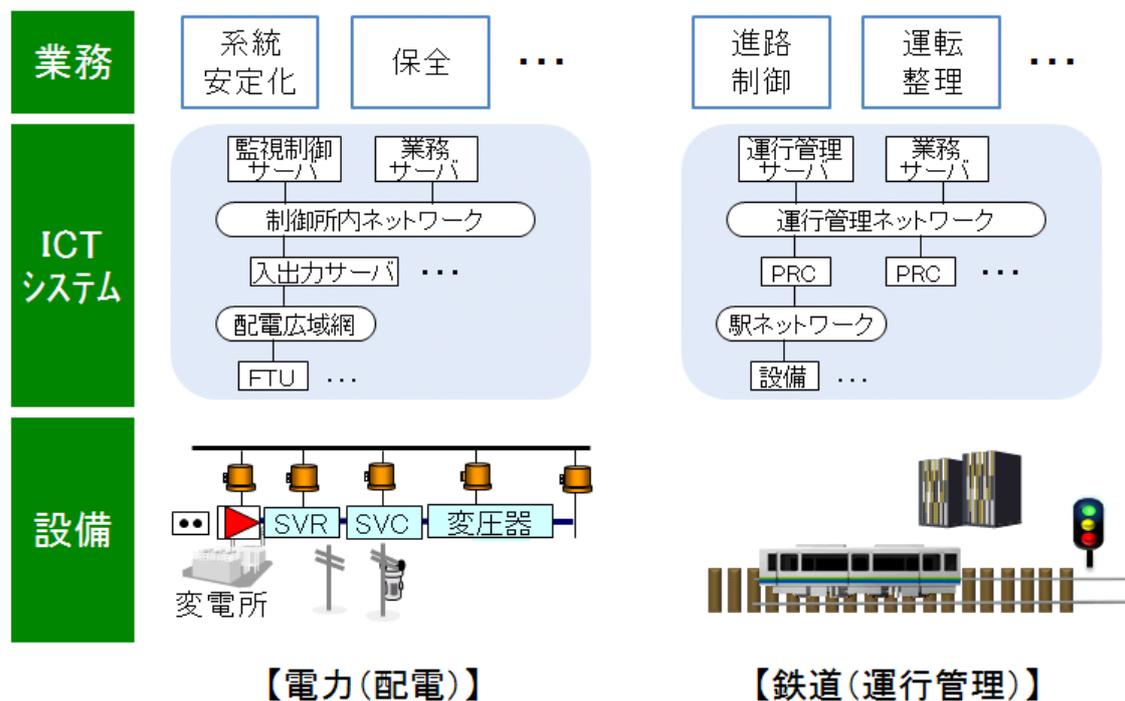


図 1-1 社会インフラシステムの概要例

図 1-1 に示した ICT システムのソフトウェア構成において、本研究の主な対象範囲を図 1-2 に示す。図 1-2 に示すように、ICT システムにおけるソフトウェアの構成として、システムコンセプト及びシステムアーキテクチャに基づいて、Operating System (OS)、汎用ミドルウェア (Program Product: PP)、分野ミドルウェア (Application Program Product: APP)、アプリケーション (Application Program: AP) がある。システムコンセプトは、システムの目的、特長、基本原理、概要等を表す。システムアーキテクチャは、システムコンセプトを具体化するために必要な、システムの構成要素及び構成要素間の関係、動作原理を表す。OS は主にハードウェアの入出力の基本機能を提供する。汎用ミドルウェア (PP) は、適用先のシステムの分野、アプリケーションの種別等に依存しない、汎用的な共通処理を提供する。Relational Data Base (RDB)、Web サーバ等が該当する。分野ミドルウェア (APP) は、適用先のシステムの分野毎の要件、準拠規格・プロトコル等に特化した仕様による共通処理 (通信、データ管理、アプリケーション実行管理、等) を提供する。分野ミ

ドルウェアは OS または汎用ミドルウェアの提供する機能を使用して実装する。アプリケーション (AP) は、適用先のシステムにおける個別の業務を実施するための処理を提供する。アプリケーションは OS、汎用ミドルウェアまたは分野ミドルウェアの提供する機能を使用して実装する。

本研究では、システムコンセプト及びシステムアーキテクチャと、これらを実現するためのソフトウェアプログラムの階層として、主に分野ミドルウェア (APP) を扱う。その他のソフトウェアの階層については必要に応じて検討するものとする。

上記の基盤技術とは、システムにおける業務処理を実現するアプリケーションプログラムに対して共通的な機能を提供する基盤に関する技術である。基盤としてはハードウェア及びソフトウェアが考えられるが、本研究で主に扱うのは、ソフトウェアであり、具体的には図 1-2 の分野ミドルウェア (APP) が該当する。

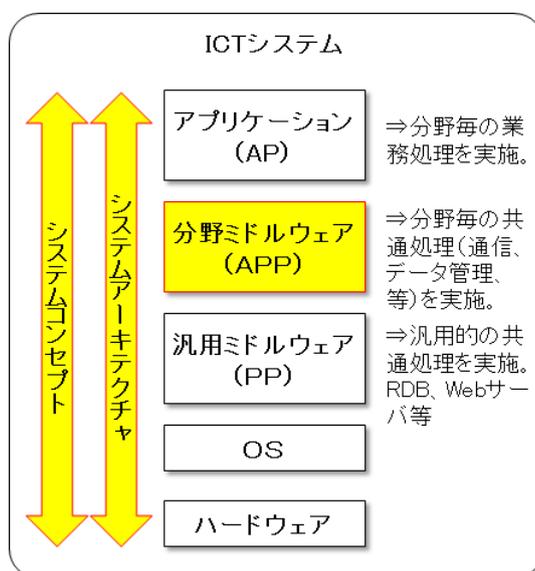


図 1-2 ソフトウェア構成における本研究の主な対象範囲

### 1.3. 本研究の目的

前節で述べた領域において、異種システム間の状況（サービスの実行状態、異常発生の有無、等）に応じた動的な連携による新サービス提供及び異常発生時のシステム動作維持、特に状況に応じたシステムの柔軟な拡張・範囲変更（状況への適応、非常時対応、等）を実現するための技術を提案することを本研究の目的とする。

上記の本研究の目的を達成する上では、目的・要件の異なる既存システムの組み合わせ及び新旧共存による System of Systems (SoS) システムの構築・更新・拡張が前提となる。またこれらの SoS システムに関してコスト低減、付加価値提供も求められる。さらに本研究の対象は、新規システムだけでなく、元々独立して各々の業務を遂行するために稼働している既存システムも含めるため、異種システム間の連携時にはこれらの既存システムの

保護、各システム間の整合性維持が制約事項となる。

#### 1.4. 本論文の構成

本論文の構成は以下の通りである。

第2章では、本研究に対する従来研究及び関連研究の動向について述べ、先に述べた本研究の目的を達成するために、これまで何ができなかったのかを明らかにしていく。

第3章では、第2章で述べた従来研究及び関連研究では対応できない問題を解決するために本研究にて提案する、異種システム連携技術について述べる。ここでは異種システム連携のためのシステムコンセプト及びシステム要件からシステムアーキテクチャを導出し、本アーキテクチャを実現するための特長機能として、動的グループ形成・制御、異種ミドルウェア連携について述べる。また本研究の提案技術を導入するシステムの運用、動作保証、既存システム保護等の、その他の必要な要素技術についても述べる。

第4章では、第3章にて述べた、本研究にて提案する異種システム連携技術を実装し、実際のシステムを構築するための方式について述べる。また提案技術の一部について、実現方式及び実施手段の観点からの評価結果について述べる。

第5章では、本研究にて提案する異種システム連携技術を適用すると効果が高いと見込まれる実問題における想定ユースケース例とそれらの想定ユースケース例を通しての評価結果について述べる。ここでは特に社会インフラ（電力）、航空宇宙の分野におけるユースケースを挙げる。

第6章では、本研究にて提案する異種システム連携技術の意義について、第5章で述べた評価結果に基づき、考察する。また他分野への応用の可能性についても検討する。

第7章では、本研究の結論と今後の課題について述べる。

## 2. 従来研究・関連研究の動向

本章では、本研究に対する従来研究及び関連研究として、これからの社会インフラを支える主要なシステム技術の1つである、System of Systems (SoS)やCyber Physical System(CPS)の概要、潮流について述べる。これらの技術潮流に対する本研究の位置付けを明確にするとともに、従来研究及び関連研究にて既に実施していること、本研究の目的を達成するために新たに必要となることについて述べていく。

### 2.1. System of Systems (SoS)

#### 2.1.1. SoS の概要

近年の環境やエネルギー等の社会課題の顕在化、人間や組織活動の地球規模での拡大、個人のライフサイクルの多様化等に対して、情報技術を活用した社会インフラの高度化が求められている<sup>1)</sup>。社会インフラの高度化を達成するためには、従来は個別にサービス提供してきた電力、交通、水道、等を個別に考えるのではなく、地域や都市の社会インフラ全体として見直すことが望ましい。上記を実現するためには、多様な企業や公共事業体を主体とする独立性の高い複数のシステムが統合されて、1つの社会システムを構成していくプロセスが必要となる<sup>1)</sup>。こうしたシステムは、図 2-1 に示すような System of Systems (SoS)あるいは超システムと呼ばれる考え方に合致する。

ここで SoS とは、複数の独立した異種のシステムを、全体目標のために合意形成の上で統合することによるシステムと定義される<sup>2)3)</sup>。Maier<sup>4)</sup>によると、システムとはある特定の機能を実行するために組織化された要素群の集まりであり、いかなる個々の要素にも還元できない振舞いや機能を生成する要素群の集まりとする。その上で SoS とは、上述のようなシステムの集まりであり、要素となるシステムは以下の2つの特徴を有するものとして<sup>5)</sup>。

- 運用的独立性 : SoS が要素となるシステムに分解された場合でも、要素となるシステムは個々に独立して動作する。
- 管理的独立性 : システムを動作させる／させないといったシステムの管理権限は、要素となる個々のシステムが有する。SoS として連携する場合も、要素となる個々のシステムは独立に運用可能である。

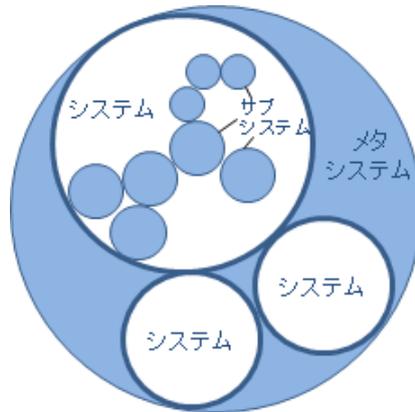


図 2-1 System of Systems (SoS)のコンセプト (参考文献5)より引用)

### 2.1.2. SoSの潮流

Boardman ら<sup>6)7)</sup> はそれまでの SoS の研究成果や動向を整理して、SoS の性質として、Connectivity (接続性)、Belongings (所属)、Autonomy (自律性)、Diversity (多様性)、Emergence (創発) の5つを挙げている。

Connectivity (接続性) とは、SoS の要素となるシステム間で相互運用するために接続を保つことである。Belongings (所属) とは、SoS の要素となるシステムは自システムの要件に従って SoS に属するか否かを選択・判断できることである。Autonomy (自律性) とは、SoS による目的を達成する上で要素となる個々のシステムは管理・運用上の独立性を保つことである。Diversity (多様性) とは、SoS の有する機能は要素となる個々のシステムにて元々設計されていた機能性を超えて多様であるべきであるという。Emergence (創発) とは、システムの発展または進化の結果として新しいシステム特性を形成することである。

多数の文献を参照することで見出した、上記の SoS の5つの性質の観点で捉えた、システム及び SoS に関する技術潮流を図 2-2 に示す。まず 2000 年頃以前には Connectivity (接続性) の観点に該当する主なものとして、分散システム等に関する技術開発に取り組まれていた。これらは広域でのシステム間連携が主な目的であった。2000 年頃以降は上記から発展して、技術潮流には大きく2つの方向がある。1つ目は主体間連携であり、2つ目は機器間連携である。本研究において、主体とは、独立した判断基準と上記判断基準に基づき自らを制御できるシステムまたは組織であると定義する。

1つ目の主体間連携の方向として、2010 年頃までに、Belongings (所属) の観点に該当する主なものとして、ソーシャルコンピューティング<sup>34)</sup>、クラウドコンピューティング<sup>35)</sup>等に関する技術開発に取り組まれていた。これらは人間系も含めて組織を跨いだ多主体間の連携が主な目的であった。その後、2010 年頃以降は、Diversity (多様性) の観点に該当する主なものとして、スマートグリッド、スマートシティ等の多主体多目的協調に関する技術開発に取り組まれている<sup>1)36)38)40)41)42)43)66)</sup>。これらは多主体間での異なる多数の目的のための連携を主な目的としている。

一方、2 つ目の機器間連携の方向として、2010 年頃までに、Autonomy（自律性）の観点に該当する主なものとして、組込みシステム<sup>8)9)</sup>、ユビキタスコンピューティング<sup>10)11)12)13)14)</sup>等に関する技術開発に取り組まれていた。これらは主に制御系を中心として多様な制御機器間の連携が主な目的であった。その後、2010 年頃以降は、Emergence（創発）の観点に該当する主なものとして、制御系及び情報系の統合に関する技術開発に取り組まれている<sup>1)5)16)</sup>。これらは多種多様な機器のより高度な制御を主な目的としている。

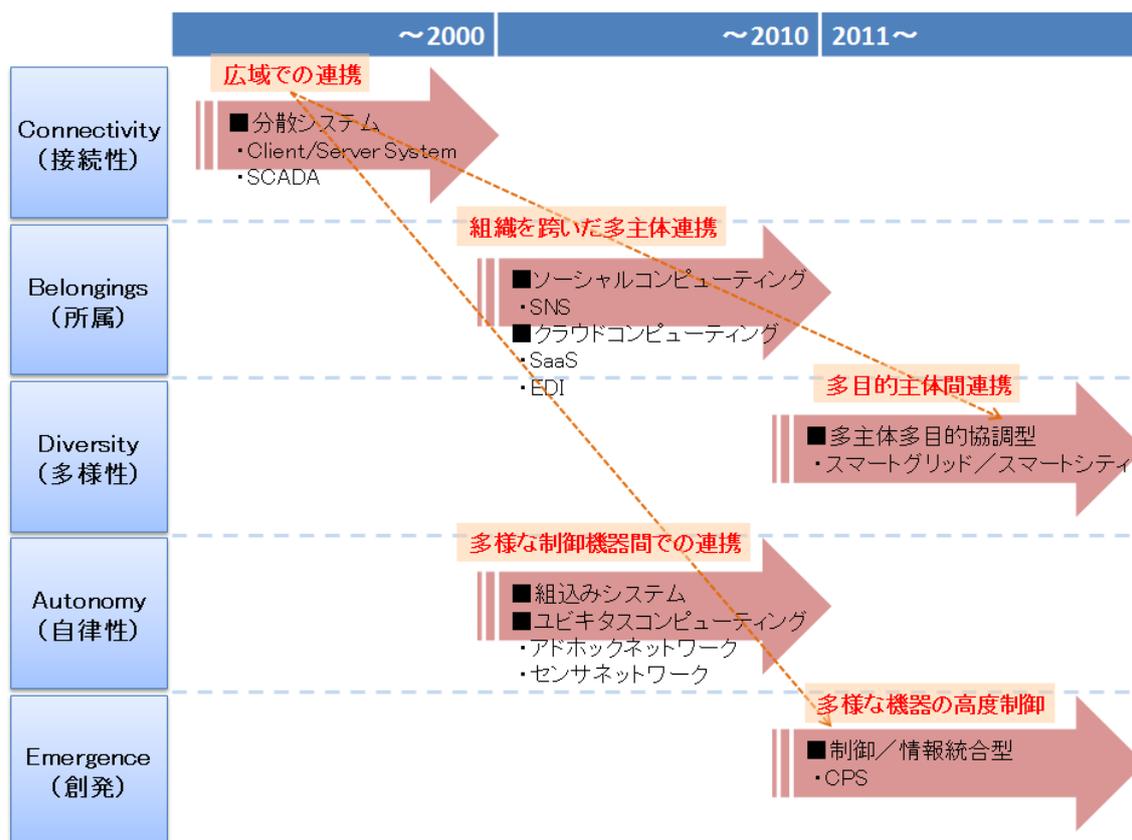


図 2-2 System of Systems (SoS)に関する技術潮流

## 2.2. Cyber Physical System (CPS)

### 2.2.1. CPS の概要

社会インフラに関する新しい研究開発の1つとしてCyber Physical System (CPS)<sup>18)</sup>の研究開発が、情報技術を中心に制御工学、計算機工学、システム工学等の関連する複数の技術を統合して進められている<sup>1)</sup>。

CPS とは、実世界 (Physical System) に浸透している組込みシステム等で構成するセンサネットワーク等からの情報と、サイバー空間 (Cyber System) の強力なコンピューティング能力とを結合したものである。図 2-3 に示すように、本技術を応用することにより多くの社会システムの効率化や新産業の創出、新サービスの創造等が強く期待されている

15)。

さらに実世界のあらゆるモノがネットワークでつながる Internet of Things (IoT)<sup>17)</sup>や多量・多発生頻度・多様性を有するビッグデータ<sup>18)19)</sup>を集積し、分析することで様々な知見・知識を見出すビッグデータ解析、等の近年の技術動向とも相まって、経済産業省産業構造審議会は、実世界とサイバー空間との相互連関するCPSが、社会のあらゆる領域に実装され、大きな社会的価値を生み出していく社会として「データ駆動型社会」を定義している<sup>16)</sup>。

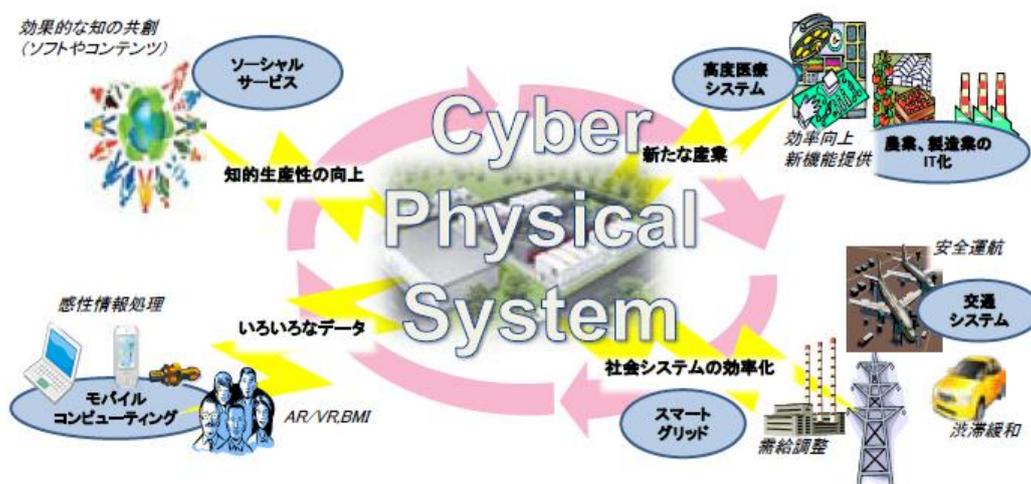


図 2-3 Cyber Physical systems (CPS) (参考文献 15) より引用

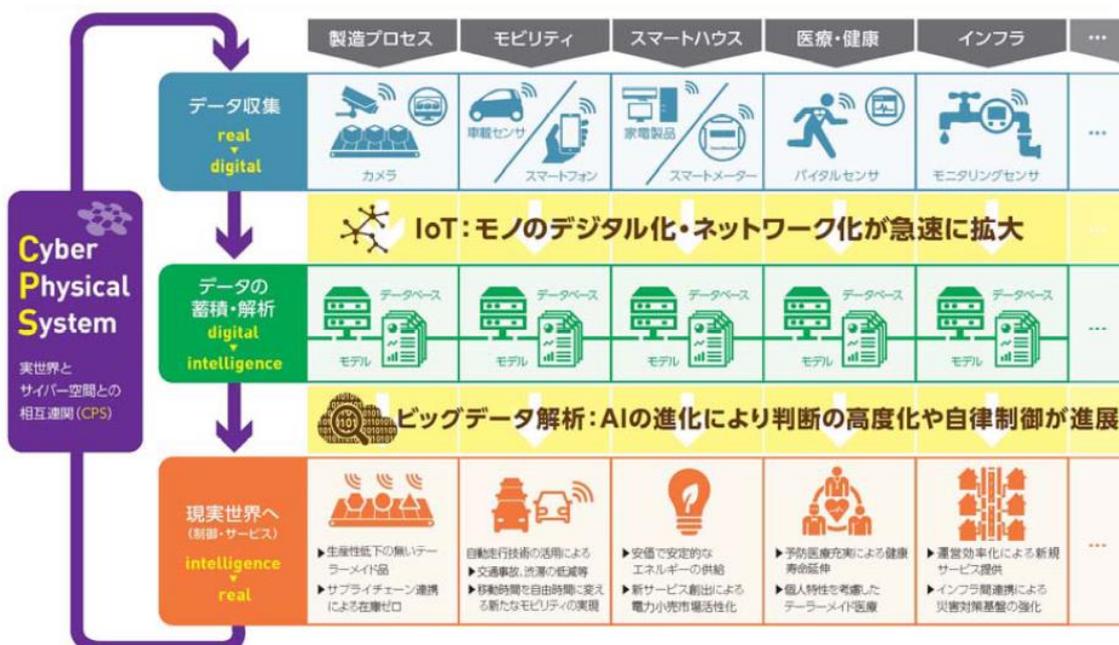


図 2-4 Cyber Physical systems (CPS) (参考文献 16) より引用

## 2.2.2. CPS の潮流

2.2.1にて述べたCPSを実現するための取組みは世界的に大きな潮流となり活発化しているが、まだ始まったばかりであり、技術が本格的な実用に至るのはまだ先である。そのような中で特に産業分野では、IoTやCPSを組合せて、従来の工場内のシステムだけでなくサプライチェーンまでに拡大して全体を最適化することを目的として、具体化が進められている。以下に代表的な例を挙げる。

### 2.2.2.1. Industrie 4.0

Industrie 4.0<sup>20)21)</sup>は、2012年にドイツ政府において閣議決定された「未来プロジェクト10」のうちの1つのプロジェクトである。「未来プロジェクト10」は、2010年に閣議決定された「ハイテク戦略2020」を具体化したものである。ドイツ機械工業連盟（VDMA）、ドイツIT・通信・ニューメディア産業連合会（BITKOM）、ドイツ電気・電子工業連盟（ZVEI）が協力し、2013年4月に「プラットフォーム Industrie 4.0」が設立された。図2-5に示すように、コンセプトは「つながる工場」である。工場内外のモノやサービスがネットワークを介して連携することで、より効率的な生産や、今までにない価値の創出、新しいビジネスモデルの構築が可能になる。ひいては、工場が様々な社会問題を解決する、としている。一例としてCPSを活用して、設計から製造、販売までの生産プロセスを最適化することを目指している。

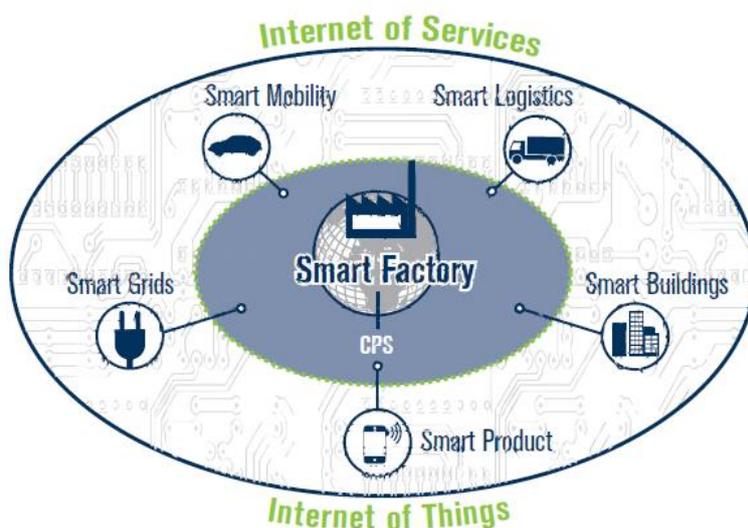


図 2-5 Industrie 4.0 (参考文献 20) より引用)

### 2.2.2.2. Industrial Internet

Industrial Internet は<sup>22)</sup>、2012年に General Electric (GE) が提唱したものである。インテリジェントな機器、高度な分析、働く人の接続によって産業のサービスやメンテナ

ンスを変革するとしている。インテリジェントな機器のデータへのアクセスが基本機能であり、GEは予測型ソフトウェアプラットフォームPredixによるデータ分析ソリューションも提供している。

一例として図 2-6 に示すように、インテリジェント機器、高度な分析、運用者とを連携して機器保守を高度化することを目指している。

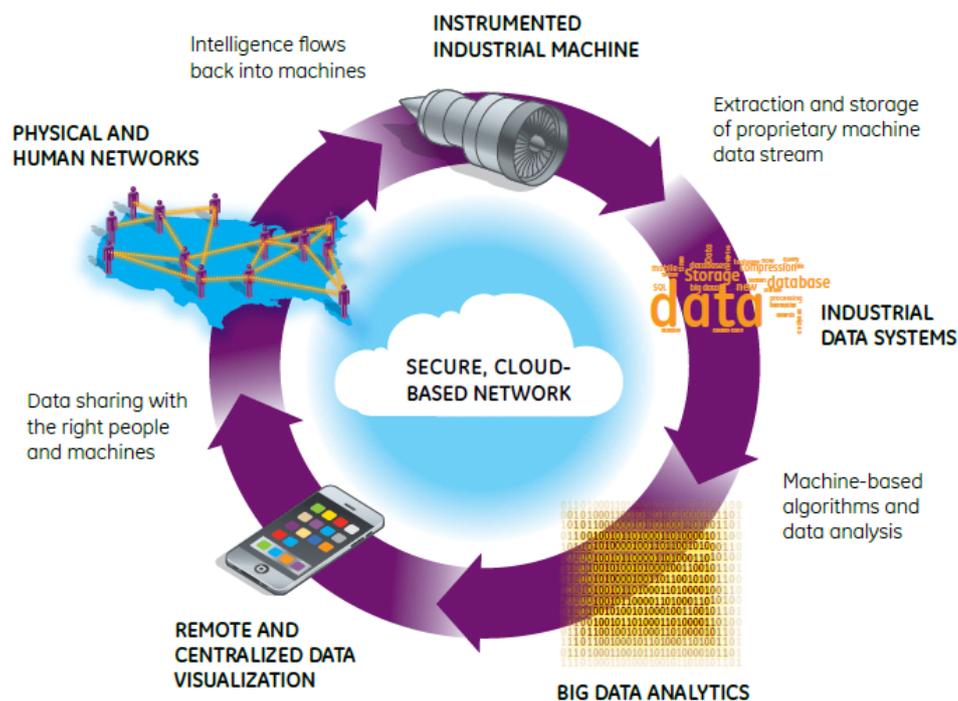


図 2-6 Industrial Internet (参考文献 22) より引用)

## 2.3. 従来研究・関連研究との比較

### 2.3.1. 本研究の位置付け

本研究による異種システム連携は SoS の 1 つの形態であると言える。2.1.2 にて述べた SoS の 5 つの性質のうち、Connectivity (接続性)、Belongings (所属)、Autonomy (自律性) の観点は対象とするシステムやサービスの前提となり得る。特に Diversity (多様性) の観点で、社会インフラに関して異なる多数の事業主体を跨いでの、状況に応じた様々な目的を達成するための、多主体多目的協調型のシステムとして本研究による異種システム連携を捉える。また Emergence (創発) の観点で、非常時対応も含めて状況に関わらず社会インフラとしてのサービスを継続するために、社会インフラシステムを構成する制御系及び情報系の柔軟な連携による、制御/情報統合型のシステムとしても本研究による異種システム連携を捉える。

一方、CPS に関しては、CPS は、従来からの企業等の特定の範囲内だけでの制御系や情報系からシステムの裾野を大きく広げる概念及び技術であり、本研究による異種システム連

携を具体的なシステムにて実現していく際に、実現形態の1つとなり得ると考える。

### 2.3.2. 従来研究・関連研究での限界

2.1.2にて述べた SoS の5つの性質の観点による、従来研究・関連研究との比較として、従来研究・関連研究での内容及び問題点と、それらに対する本研究でのアプローチを表 2-1 にまとめる。なお表にまとめた従来研究・関連研究での内容及び問題点の詳細は本節の以降にて述べる。また本研究のアプローチの詳細は 2.4 にて述べる。

表 2-1 従来研究・関連研究との比較

	従来研究・関連研究での内容	問題点	本研究でのアプローチ	本研究の提案技術
Connectivity (接続性)	<ul style="list-style-type: none"> <li>自律分散プロトコル (ADS-net)</li> <li>⇒データフィールド介したデータ受渡し</li> </ul>	同一ネットワークセグメント内でのみシステム構築、異なるシステム間の接続は未対応。	システム毎に導入される異なるミドルウェア間の相互接続、連携。	異種ミドルウェア連携
Belongings (所属)	<ul style="list-style-type: none"> <li>自律分散サービスシステム (ADSS)</li> <li>⇒メタデータを介したサービス連携</li> </ul>	連携元の元々の業務処理の保護、異なる種別の主体間での全体的な効率化も必要。	サービスと実際のインフラシステム、それらを相互接続し仲介するシステムビュー、の3つの階層から成るアーキテクチャ。	異種システム連携コンセプト
Autonomy (自律性)	<ul style="list-style-type: none"> <li>超分散オブジェクトモデル (SDO)</li> <li>⇒統一モデルを用いた機器連携サービス</li> </ul>	多種多様、大量の機器を一様に扱う場合の管理負荷増大。	多種多様、大量の機器を、各機器の状態やサービスに対する要求内容に応じて制御、管理。	動的グループ形成・制御
Diversity (多様性)	<ul style="list-style-type: none"> <li>スマートシティ (デマンドレスポンス)</li> <li>⇒主体間連携</li> </ul>	同種別の主体間連携のみ達成、異なる種別の主体間での全体的な効率化も必要。	サービスと実際のインフラシステム、それらを相互接続し仲介するシステムビュー、の3つの階層から成るアーキテクチャ。	異種システム連携コンセプト

	従来研究・関連研究 での内容	問題点	本研究での アプローチ	本研究の 提案技術
Emergence (創発)	• SoS による相互運用 時のシステム振舞 いモデル	実システムへの適用 検討はこれから。	サービスと実際の インフラシステム、 それらを相互接続 し仲介するシステ ムビュー、の3つの 階層から成るアー キテクチャ。	異種システム 連携コンセプ ト

表 2-1 にまとめたように、2.1.2 にて述べた SoS の5つの性質のうち、Connectivity (接続性) の観点での1つとして、情報制御システムにおいてデータの共有を行うために、従来から制御系を中心として、自律分散システムアーキテクチャ<sup>23)24)</sup>及び本アーキテクチャにおける通信規約である自律分散プロトコル ADS-net<sup>25)</sup>を導入している。これらによりデータフィールドというデータ共有の場を形成し、システムを構成する計算機間でデータの受け渡しを行っている。ADS-net によるデータフィールドでは、データソース側にて制御タイミングに従って、データを格納したメッセージをマルチキャスト送信する。受信側はこれらのメッセージの中から必要なものを選択し受信する。このアーキテクチャにおいて各計算機は自分が必要とするデータを局所的に持ち、独自の判断でデータフィールドから任意のデータを取得したり、発信できる。このためデータフィールドにいつ接続しても、システム全体の構造を知らなくとも必要なデータを直に取得できる。また送受信タイミングの違いを吸収するために、送信されたデータを一定期間蓄積するストレージデータフィールドを提供する<sup>26)</sup>。

なお分散システム向け通信である Data Distribution Service (DDS)<sup>29)</sup>による Publish/Subscribe 機能を用いるとイベント発生時のメッセージ配信は可能となる。

ここで ADS-net 等を用いる制御系において性能、処理タイミング等の要件を満たすためには、上記データフィールドはネットワークセグメント内に構築する必要がある。制御系と情報系のように異なるシステムにおいては、既実績を持って導入されているプロトコル、基盤・ミドルウェアや運用形態等が異なるだけでなく、データ構造・粒度、処理周期・タイミング等も異なる。またシステム間連携のために全てを新規に置き換えることも困難である。

ゆえに SoS を構成する異なるシステム間で相互接続し、連携して動作を行うためには、各々の系の物理構成やデータの配置等に基づき既に稼働している、異なるミドルウェアの間での連携が必要となり得る。

2.1.2 にて述べた SoS の5つの性質のうち、Belongings (所属) の観点での1つとして、ネットワーク上で複数のサービスを連携、統合させることで新たなサービスを提供する、

自律分散サービスシステム<sup>27)28)</sup> (Autonomous Decentralized Service System : ADSS) が提案されている。本サービスシステムでは、ユーザの要求に対してメディエータが仲介して様々なサービスの中から最適なものを選出し、組合せて提供する。個々のサービスはサービスシステムへの参加、退去は自由とする。情報提供サービスを中心として主に制御システムと情報システムとの連携に適用されてきたが、異なる種別で目的も異なる複数の主体のシステム間の連携となると、連携の際に連携元の元々の業務処理の性能、信頼性を保証すること、主体間の違いを調整した上で全体的な効率化が必要となる。

2.1.2にて述べた SoS の5つの性質のうち、Autonomy (自律性) の観点での1つとして、自律した多様な制御機器間で連携しサービスを提供するために、超分散オブジェクト (Super Distributed Objects : SDO) モデル<sup>30)33)</sup>が提案されている。本モデルはハードウェア機器やソフトウェアコンポーネントの論理的表現であり、統一的なデータモデルと外部アクセスインタフェースを定義している。多様な機器の連携によるサービスを提供する上で、個々の機器の仕様の違いを意識せず一様に扱い、アプリケーションを共通的に作成可能であるが、IoT や CPS 等の技術の進展に伴い、対象とするシステムの裾野が拡大する程に、多種多様で大量の機器を一様に扱うことは、個々の状態の把握も必要となり、管理の負荷が増大してしまう。

2.1.2にて述べた SoS の5つの性質のうち、Diversity (多様性) の観点での1つとして、スマートシティの一環でビル群の協調によるデマンドレスポンスの実証実験の事例がある<sup>36)</sup>。リアルタイムなビル設備状態監視、統合的なエネルギー管理、効果的なデマンドレスポンス計画作成等の技術に伴い、個々のビルの電力使用計画を支障なく遂行するとともに、電力逼迫時も電力需給バランスを維持するという多目的を、システム (ビル) 間で協調しながら達成できることを実証した。ただし主にビルという同じ種別で目的も類似する複数の主体間の連携での達成であった。異なる種別で目的も異なる複数の主体間の連携となると、主体間の違いを調整した上で全体的な効率化が必要となる。

2.1.2にて述べた SoS の5つの性質のうち、Emergence (創発) の観点での1つとして、SoS の要素となるシステムの追加・削除における相互運用性確保のため、要素となるシステムの突発的な振舞いを事前把握するための振舞いモデルの検討が行われている<sup>37)</sup>。ただしまだ机上評価の段階であり、実システムへの適用検討はこれからである。

## 2.4. 本研究のアプローチ

本研究は社会インフラシステムを対象として、異種システム間の状況に応じた動的な連携による新サービス提供及び異常発生時のシステム動作維持、特に状況に応じたシステムの柔軟な拡張・範囲変更を実現するための技術を提案することを目的とする。本目的を達成する上では、目的・要件の異なる既存システムの組み合わせ及び新旧共存による SoS としてのシステムの構築、更新、拡張が必要となる。

ゆえに 2.3 にて述べた従来研究・関連研究との比較から、Diversity (多様性)、Emergence

(創発)の観点から、複数の異なる種別のシステムの連携であっても、全体効率化や様々な新たなサービス提供を実現するために、サービスと実際のインフラシステム、さらにそれらを相互接続し仲介するシステムビュー、の3つの階層から成るアーキテクチャを有する異種システム連携コンセプトを提案する。これらの詳細は3.1にて述べる。

Autonomy (自律性)の観点では、上述の異種システム連携コンセプトを具体化する上で、システムの裾野が拡大され、多種多様で大量の機器を、各機器の状態やサービスに対する要求内容に応じて柔軟に扱うための、動的グループ形成・制御の技術を提案する。これらの詳細は3.2にて述べる。

Connectivity (接続性)の観点では、上述の異種システム連携コンセプトを具体化する上で、実際のシステム間で仕様や動作状況の違いを吸収するために、個々のシステムの物理構成やデータの配置等に基づき既に稼働している異なるミドルウェアの間での連携を行う、異種ミドルウェア連携の技術を提案する。これらの詳細は3.3にて述べる。

## 3. 異種システム連携技術

### 3.1. 異種システム連携のコンセプト

#### 3.1.1. 概要

新しいサービス提供、システム動作のさらなる効率化・高機能化、等のために、元々の目的、業務、導入時期、導入場所、準拠規格、運用方法等の異なる2つ以上のシステムを相互に連携することを、異種システム連携と定義する。ここでサービスとは、インフラシステムにおける業務処理またはインフラシステムの利用者に対して付加価値を提供することを指す。

本研究にて提案する異種システム連携のコンセプトを図 3-1 に示す。図 3-1 に示すように「サービス」、「システムビュー」、「インフラシステム」の3つの階層により構成する。

「サービス」は、1つ以上のシステムにおける1つ以上の機器及びソフトウェアプログラムが連動して提供する。サービスに必要な機能、データ、トリガーとなるイベント及び実行条件、等を定義するサービスシナリオに基づき、サービスは実行される。なお社会インフラにおけるサービスは、インフラの定常状態時だけでなく、異常発生時でも、縮退もしくは一時的な中断は許容してでも維持すべきものである。

「システムビュー」は、異種のシステム間の連携において、システム間の相互接続及び仲介を行い、ユーザまたは他システムからのシステムの見え方となる。他システムに対して公開しアクセス可能とするシステム内のデータ及び機能をノードと定義する。ノードは各々のインフラシステムにおける実際のデータ、機能に対してマッピングされ、データ及び機能の粒度、アクセス権限等は各システムの連携ポリシー及び運用ポリシーに従って決定する。サービスはこれらのノードの組合せ、呼出しにより実行される。

「インフラシステム」は、各種のインフラ毎に構築するハードウェア及びソフトウェアによる、実際のシステムである。なおインフラシステムは本来の各インフラにおける制御等の業務を実施している上で他システムとの連携を行う。このためシステム連携時も各インフラシステムにおける本来の業務実行の性能及び信頼性の維持は前提条件となる。

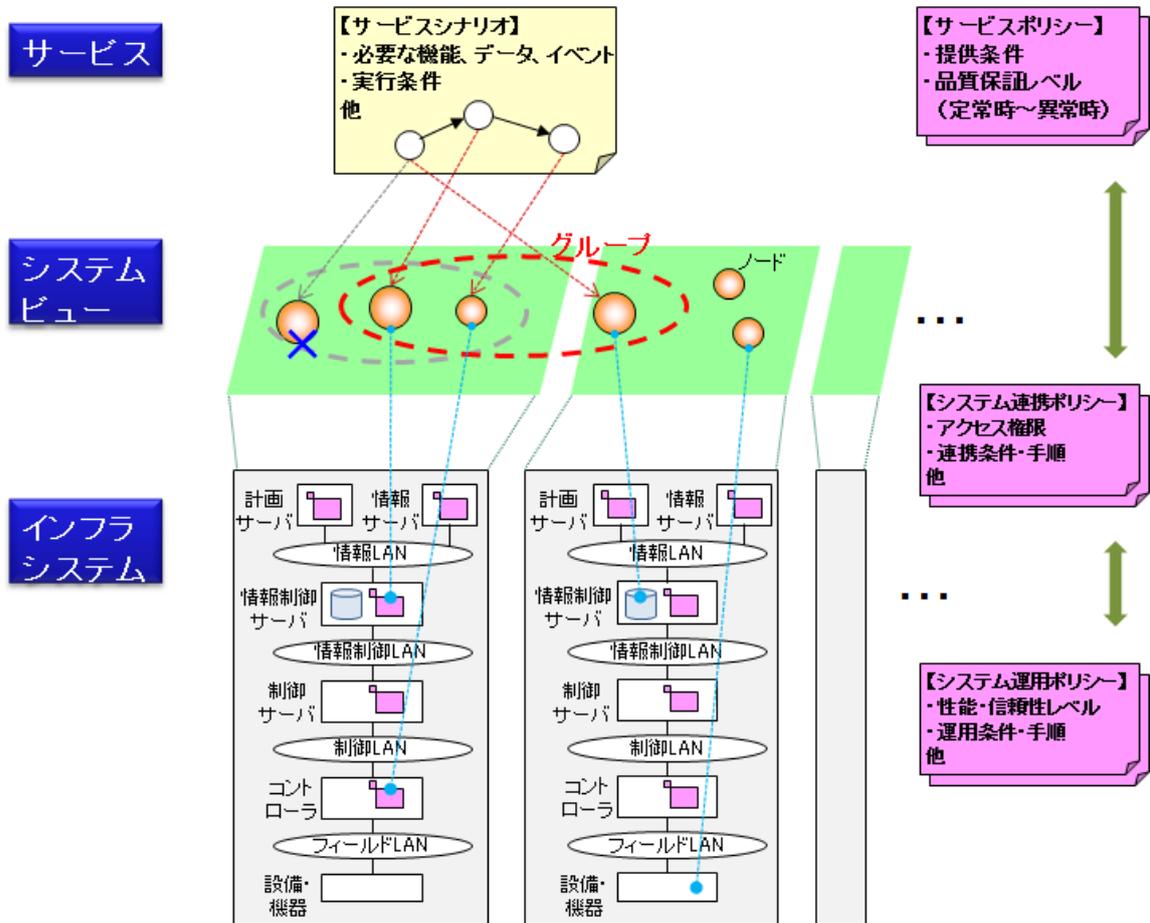


図 3-1 異種システム連携のコンセプト

上記のサービス、システムビュー、インフラシステムの一例として、例えば電力分野では、異なる事業者のインフラシステム間で余剰電力を融通し全体で電力使用量の最適化を図るといふ、電力融通というサービスがある。本サービスを実施するために、システムビュー上に、各事業者のインフラシステムの電力使用実績量及び使用計画量を監視するノード、各インフラシステムの電力需給調整を行うエネルギーマネジメントを実施するノード等を配置し、融通を行うシステムを包含するようにグループを決める。これらのノード間の関係、実行順序等を記述するサービスシナリオに従って、電力の余ったインフラシステムから電力の不足するインフラシステムへの融通を決定する。上記決定に基づき、各エネルギーマネジメントのノードに対して、電力需給調整の要求を行う。各エネルギーマネジメントのノードを通して、実際のインフラシステムの制御系への制御指示等を行う。

### 3.1.2. サービス

3.1.1 にて述べた本研究におけるサービスは、サービスシナリオに基づき実行される。筆者はサービスシナリオの種類として図 3-2 に示す 3 種類に整理する<sup>44)</sup>。図 3-2 に示すよ

うに、サービスの構成要素間の繋ぎ方と実行タイミングに着目して大きく分類すると、直接的に繋ぐインタフェース接続型（Interface Connection）、間接的に繋ぐワークフロー型（Work Flow）、順序を制御するか否かはアプリケーションに依存するルール/ロジック型（Rule/Logic）の3つの類型となる。インタフェース接続型ではコンポーネント間の入出力の接続関係を表現し、複数のコンポーネントがデータの入出力関係を持って同時実行する。ワークフロー型ではあるコンポーネントが1つ以上のコンポーネントを実行フローに従って呼び出し実行していく。ルール/ロジック型では条件判定に基づき条件に適するコンポーネントを選出し実行していく。

インタフェース接続型は映像のストリーミング配信や制御シーケンス等のサービスに用いられる。ワークフロー型はフローに従って実行されるビジネスアプリケーション等に用いられる。ルール/ロジック型は状況に応じた動的な変更が必要となる、適応型アプリケーション等に用いられる。

本研究にて用いるべきサービスシナリオは、上述の3つの類型のうちの1つもしくは2つ以上の組合せによるものである。

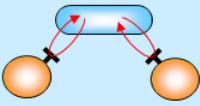
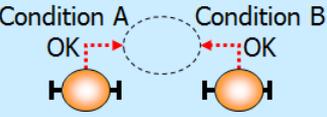
	インタフェース接続型	ワークフロー型	ルール/ロジック型
サービスシナリオの類型	 <p>入出力の接続関係を表現</p>	 <p>実行フローを表現</p>	 <p>条件判定に基づくコンポーネント選出及び実行を表現</p>
アプリケーション	ストリーミングサービス/制御シーケンス	ビジネスアプリケーション	適応型アプリケーション

図 3-2 サービスシナリオの3つの類型（筆者文献 44）より引用、一部加筆）

複数のサービスを統合してのビジネスプロセスを記述するためのワークフロー言語に Business Process Execution Language (BPEL)<sup>45)46)</sup>がある。図 3-3 に示すように、BPEL を用いると、1つ以上のサービスを呼び出すことによりサービスを提供する。また BPEL で統合されたサービスをさらに呼び出すことが可能である。BPEL を用いてサービスを実行する場合、図 3-3 に示す BPEL Process はフローに従ってサービスを順次呼び出し実行していく点では図 3-2 で示したサービスシナリオのうちワークフロー型に該当する。一方、各サービスを呼び出す際に、条件判定に基づき条件に適するものを選出することもできる点ではルール/ロジック型にも該当する。

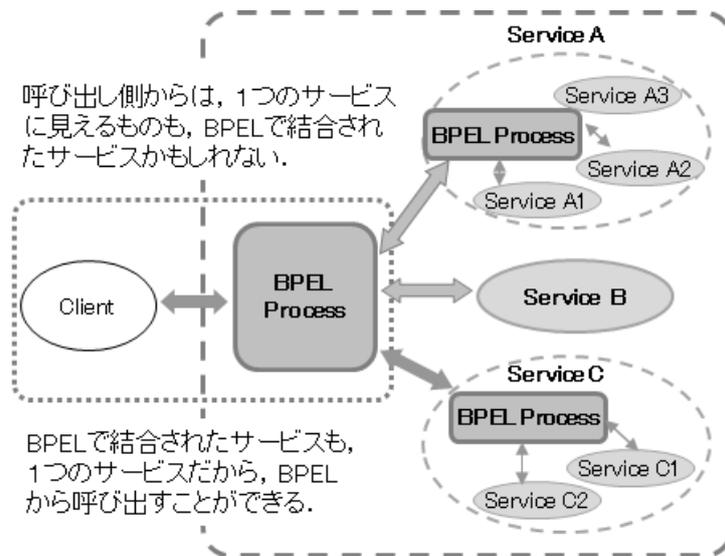


図 3-3 BPEL によるサービスの統合 (参考文献 45)より引用)

ここでBPELによるサービスは、図 3-3に示すようにクライアントからの呼び出し、もしくは他アプリケーションからのメッセージ受信を起点として実行される。ただし本研究が対象とする社会インフラシステムにおける制御系を含める業務にて実行されるサービスは表 3-1に例示するようなイベント・事象を実行開始のトリガーとする。これらはBPELでは対応できないが、本研究にて用いるサービスシナリオでは、サービス実行開始条件として対応する必要がある。

さらに社会インフラシステムにおける業務では、システムの運転モードとして、“本番モード”、“テストモード”、“オフラインモード”等を区別して扱う。これらもBPELでは対応できないが、本研究にて用いるサービスシナリオでは、サービス実行開始条件として対応する必要がある。

制御系でも使用されるワークフロー型のサービスシナリオとしてシェルスクリプトが挙げられる。ただしこれはOSのシステムコールに依存した形式となる。本研究にて用いるサービスシナリオは、異なる複数のシステムの連携を前提としてあらゆる環境で動作すべきであるので、OSに非依存であることが必要である。

表 3-1 制御系サービスの実行開始条件の例

#	実行開始条件	例
1	•データ値が指定の閾値を越える場合	

#	実行開始条件	例
2	<ul style="list-style-type: none"> <li>フラグ（他アプリケーションがセット）が立つ場合</li> <li>スイッチが入る場合</li> </ul>	
3	<ul style="list-style-type: none"> <li>ある状態になる場合 (例：電車が前の閉塞区間に入る)</li> </ul>	
4	<ul style="list-style-type: none"> <li>指定の時刻になる場合</li> </ul>	
5	<ul style="list-style-type: none"> <li>キューからデキューされる場合</li> </ul>	

これらの要件を満たすべく、本研究で提案するサービスシナリオについては3.2.2.2にて述べる。なお上述のようなサービスシナリオを用いることで、図3-4に示すように、これまでシステム毎に個別にアプリケーションにて作り込んでいた、実行順序や実行制御に関する処理がサービスシナリオとして共通化できる。特に異なる仕様・規格に準拠する、複数の異なるシステムを跨いで、アプリケーションの連携によるサービスを作成する場合には、各システムの準拠仕様・規格に依存した連携処理を個別に作り込む必要はなく、ソフトウェアの生産性向上につながる。

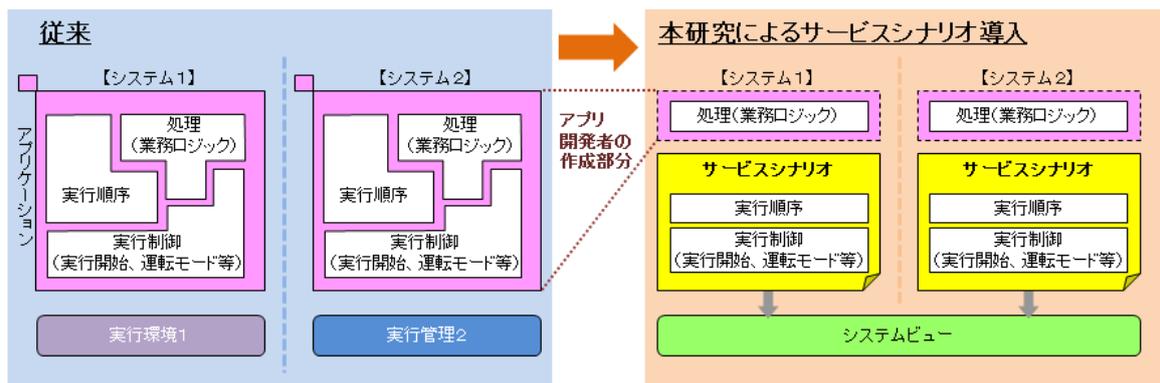


図 3-4 サービスシナリオ使用による生産性向上

### 3.1.3. システムビュー

3.1.1にて述べた本研究におけるシステムビューは、サービスと個々のインフラシステムとを仲介する階層である。これらの3階層を用いてサービスを実行し維持するために、システムビューの階層において、状況に応じて1つ以上のインフラシステムを跨いで、サ

サービスに必要なノードを動的に選出する。これらのサービスに必要なノードの種別はサービスシナリオにて定義しておく。またこれらのサービス実行に必要な1つ以上のノードの集合をグループと定義する。サービスはこのシステムビューにおけるグループ内でノードの連動動作（集中制御もしくは分散制御）により実行される。

各ノードに紐づくインフラシステム側の状態を監視しておき、アクセスが不可となった場合には、該当ノードをグループから削除し、同等のノードで連携可能な他のノードを探索し、発見すれば代わりにグループに追加する。これらのグループの管理及び制御はシステムビューの上で実施する。

### 3.1.4. インフラシステム

3.1.1にて述べた本研究におけるインフラシステムは、各種のインフラ毎に構築するハードウェア及びソフトウェアによる、実際のシステムである。分野毎の差異はあるが、通常、設備の入出力を扱うフィールド系、設備制御及び監視を行う制御系、制御の計画を作成、管理する計画系、複数の制御系及び計画系間の円滑な連携や運用を行う情報制御系、上位の業務を行う情報系、等に階層化される。これらの階層化は主に、性能・信頼性要件、扱うデータの粒度・構造、管理体系、等の違いに起因する。

例として電力・スマートグリッドの分野では、システムのリファレンスアーキテクチャとして Smart Grid Architecture Model (SGAM)<sup>47)</sup>が規定されている。SGAMは、図 3-5 に示すようにビジネスプロセス (Business)、機能 (Function)、情報モデル (Information)、通信プロトコル (Communication)、コンポーネント (Component) を表現した5つのレイヤで構成される。図 3-6 に示すように各レイヤは、ドメイン (Domains) とゾーン (Zones) の2軸で表現される。これらの軸は電気系のプロセスの観点（電気エネルギー変換チェーンの物理的ドメインで分類）と、情報管理の観点（電気系のプロセスを管理するための階層で分類）による。表 3-2 に各ゾーンの概要をまとめる。

本研究では、インフラシステムにおける対象範囲、制御箇所、データや処理の所在を明確化するために SGAM のリファレンスアーキテクチャを参考にする。

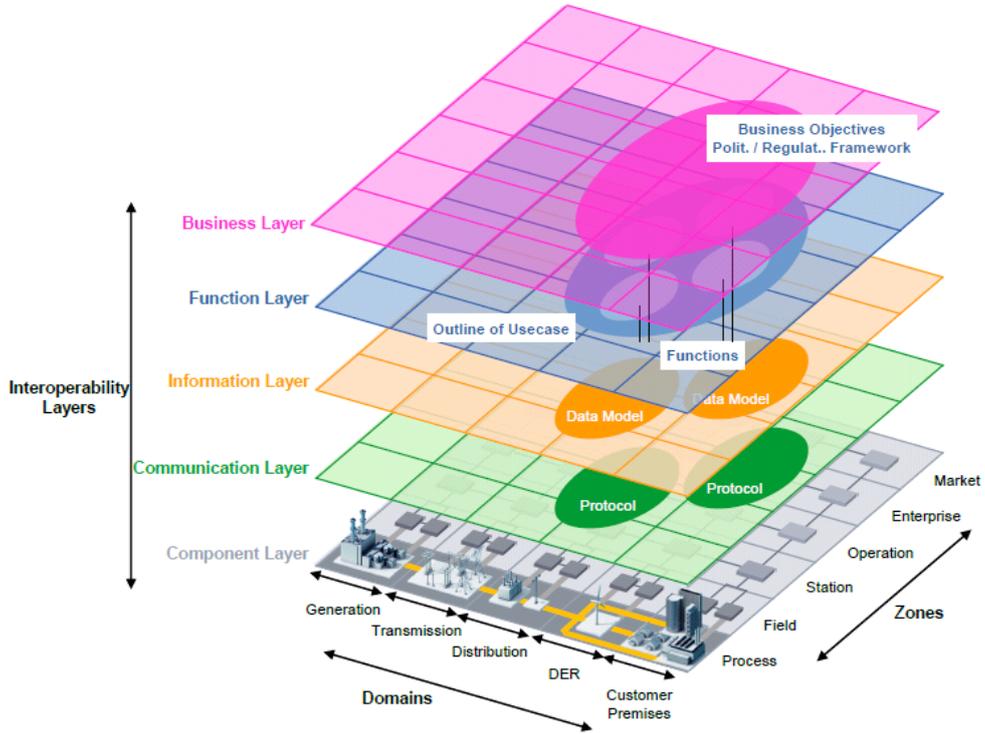


図 3-5 SGAM フレームワーク (参考文献 47) より引用)

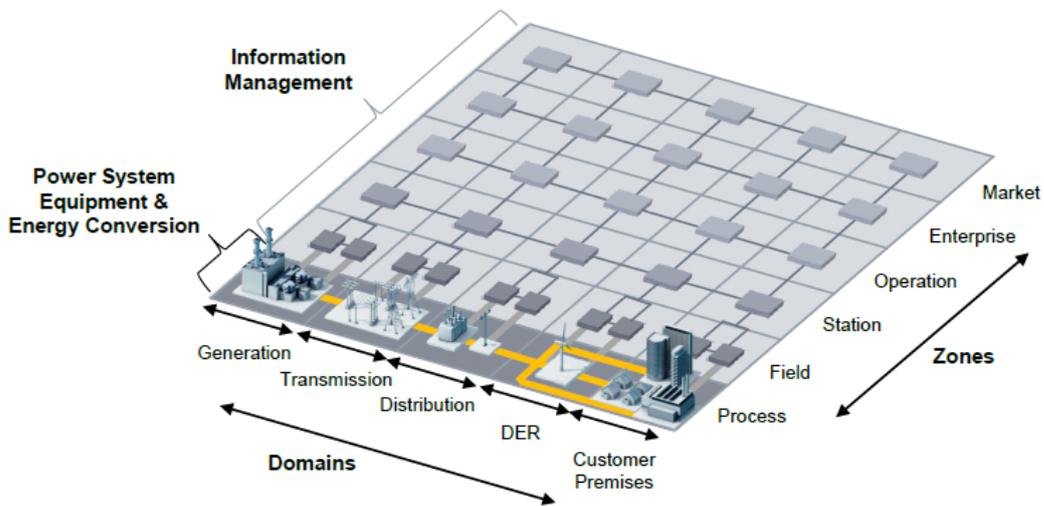


図 3-6 Smart Grid plane (参考文献 47) より引用)

表 3-2 Smart Grid Plane のゾーン (参考文献 47) より引用)

#	ゾーン	説明
1	Process	物理的、化学的、空間的なものを含むエネルギー（電気、太陽光及び太陽熱、熱、水、風等）搬送と、物理的な設備を記載する（例えば発電機、変圧器、サーキットブレーカ、架空線、ケーブル、Process ドメインに部分

#	ゾーン	説明
		的もしくは直結されている全ての電力負荷センサやアクチュエータ)
2	Field	電力系の Process ドメインの機器を保護、制御、監視する設備を記載する。例えば、保護リレー、ベイコントローラ、電力系の Process ドメインのデータを取得し使用する全てのインテリジェントな電気機器、等。
3	Station	Field ドメインに対する空間的な集約を表現する。例えば、データの集約、機能の集約、変電所の自動制御、ローカルな SCADA システム、プラント監視システム、等。
4	Operation	個々のドメインにおける電力系の制御操作を行う対象を記載する。例えば、Distribution Management System (DMS)、発電や送電の Energy Management System (EMS)、マイクログリッド管理システム、Virtual Power Plant (VPP) マネジメントシステム、Electric Vehicle (EV) 群の充電管理システム、等。
5	Enterprise	企業における商業プロセス、組織プロセス、サービス、インフラを含む(設備、サービスプロバイダ、エネルギー売買)。例えば、アセットマネジメント、兵站、ワークフォースマネジメント、スタッフ教育、Customer Relation Management (CRM)、請求、調達、等
6	Market	エネルギー変換チェーンに沿って実施可能なマーケット操作を反映。例えば、エネルギー売買、電力市場、小売市場、等。

## 3.2. 動的グループ形成・制御

### 3.2.1. グループの定義と動作原理

#### 3.2.1.1. グループの定義

3.1 で述べた本研究におけるシステムビューの階層にて、サービス実行に必要な1つ以上のノードの集合であるグループは以下に挙げる構成要素から成る。

- コンダクタ (Conductor) : グループメンバの1つで、3.1 で挙げたサービスシナリオに基づき、グループメンバである1つ以上のノードを指揮して、サービス実行を制御する。
- ノード (Node) : グループメンバの1つで、サービス実行のために呼出される。実システムにおけるソフトウェアにより提供される機能やデータ等にマッピングされる。
- モニタ (Monitor) : グループの構成要素及びシステムの状態を監視する。

メディエータ : 連携場間の仲介をする。  
(Mediator)

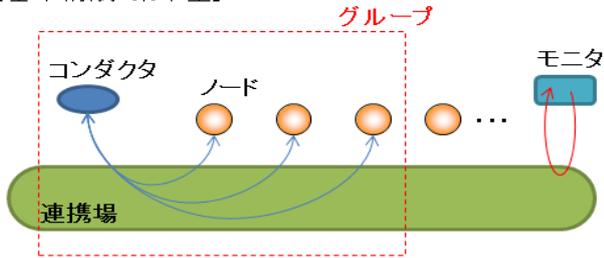
上記の各々のグループ構成要素が相互に連携しグループとして動作するための環境として、ノードをはじめ、グループ構成要素の間の連携及びデータ共有の場である連携場を導入する。連携場は自律分散システムアーキテクチャ<sup>23)24)</sup>に基づくものであり、データ生成側となるグループ構成要素にて制御タイミングに従って、生成したデータを連携場に投入する。データ利用側のグループ構成要素はこれらのデータの中から必要なもののみを選択し取得する。このアーキテクチャにおいて各々のグループ構成要素は自分が必要とするデータを局所的に持ち、独自の判断で連携場から任意のデータを投入し取得できる。このため連携場にいつ接続しても、システム全体の構造を知らなくとも必要なデータを直に取得できる。

図 3-7 にグループの構成要素と構成パターンを示す。図 3-7 に示すように、基本構成と連携構成とを挙げる。

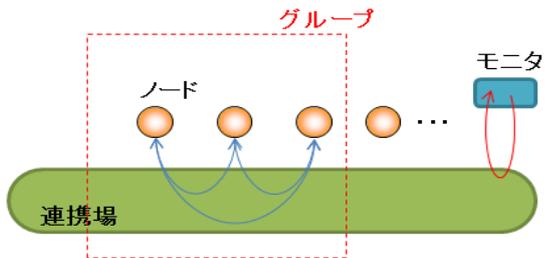
基本構成では、集中型と分散型とがある。集中型ではグループは連携場上のコンダクタと1つ以上のノードにより構成される。コンダクタは連携場を介して 3.1 で挙げたサービスシナリオに基づいた呼出し手順によりグループメンバであるノードを順次呼出し、実行させる。各ノードの実行結果を受けてコンダクタがグループとしてのサービスの実行制御を集中的に実施する。一方、分散型ではコンダクタは無く、グループは1つ以上のノードで構成される。各ノードは連携場を介してデータのやり取りを行い、これらのデータの入出力に従って自律的に動作する。これらのノードの動作の結果としてサービスが提供される。なおモニタは集中型、分散型いずれにおいてもコンダクタ及びノードの状態、サービス実行状態を監視し、障害発生等により実行不能なコンダクタ、ノードの検出またはサービス実行不能等の検出を行うと、グループの構成変更を促す。

連携構成では、集中型、分散型及びコンダクタ連携がある。集中型及び分散型では、グループメンバであるコンダクタ及びノードの動作は上記の基本構成の場合と同様である。これに加えて、メディエータの仲介により他の連携場に参加するノードともデータのやり取りを行い、該ノードもグループメンバとなり連携、動作する。一方、コンダクタ連携では、メディエータの仲介により複数のコンダクタがデータのやり取りを行う。これにより複数のグループが連携動作を行う。なおコンダクタ連携は同一連携場上の場合及び複数の連携場を跨ぐ場合でも可能である。

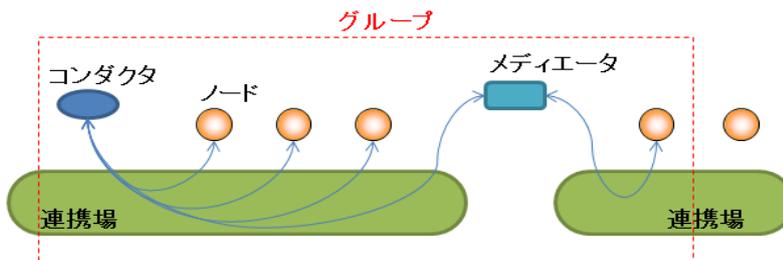
【基本構成: 集中型】



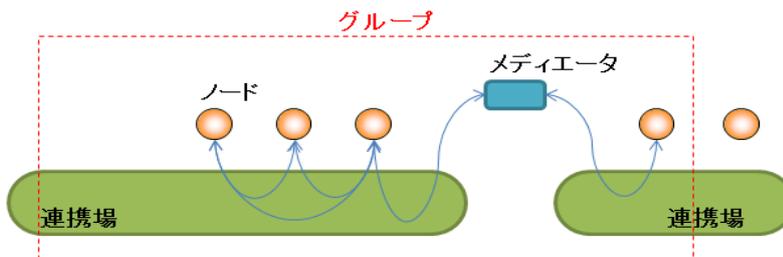
【基本構成: 分散型(コンタクタ無し)】



【連携構成: 集中型】



【連携構成: 分散型(コンタクタ無し)】



【連携構成: コンタクタ連携】

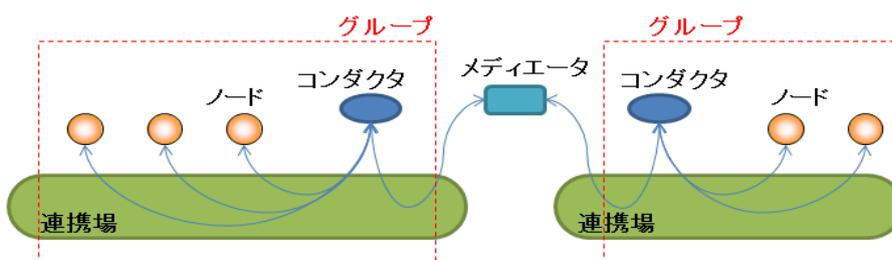


図 3-7 グループの構成要素と構成パターン

### 3.2.1.2. 連携場の階層化

本研究が対象とする社会インフラシステムは、性能要件や確実な処理実行に関する信頼性要件が厳しい制御系から、これらの要件は緩くなるが、代わりに扱うデータの種別が多くサイズが大きい情報系までを含めている。このため 3.2.1.1 にて述べたグループを形成するコンダクタ及びノードに対する性能要件及び信頼性要件、データの種別及びサイズもグループにて扱うサービスの内容に従って一様ではない。このためコンダクタ及びノードが動作する連携場を、性能・信頼性の要件に基づいて階層化する。ここでは性能要件・信頼性要件が厳しい順に監視制御向け連携場、緩い制御向け連携場、業務・情報系向け連携場の 3 階層とする。これらの階層間でのやり取り・連携は 3.2.1.1 にて述べたメディアータを介して実施する。

上述の連携場の階層化を社会インフラシステムの構成に適用した場合の概要を図 3-8 に示す。なお図 3-8 のシステムの階層は 3.1.4 で述べた電力・スマートグリッド分野における Smart Grid Architecture Model (SGAM) にて定義する階層及び産業分野における ISA-95 の製造モデル<sup>80)</sup>に当てはめている。監視制御向け連携場ではフィールド (ISA-95 の L0~L1。SGAM の Process~Field) からのデータを参照し制御指示を行う。緩い制御向け連携場では制御系 (ISA-95 の L0~L2。SGAM の Process~Station) からの実績情報を参照し、制御系に対して計画情報を作成し配信する。業務・情報系向け連携場では、フィールド及び制御系、計画系 (ISA-95 の L0~L3。SGAM の Process~Operation) から取得した情報を基にバックオフィス系の業務、上記の情報を用いた新サービスの提供等を行う。

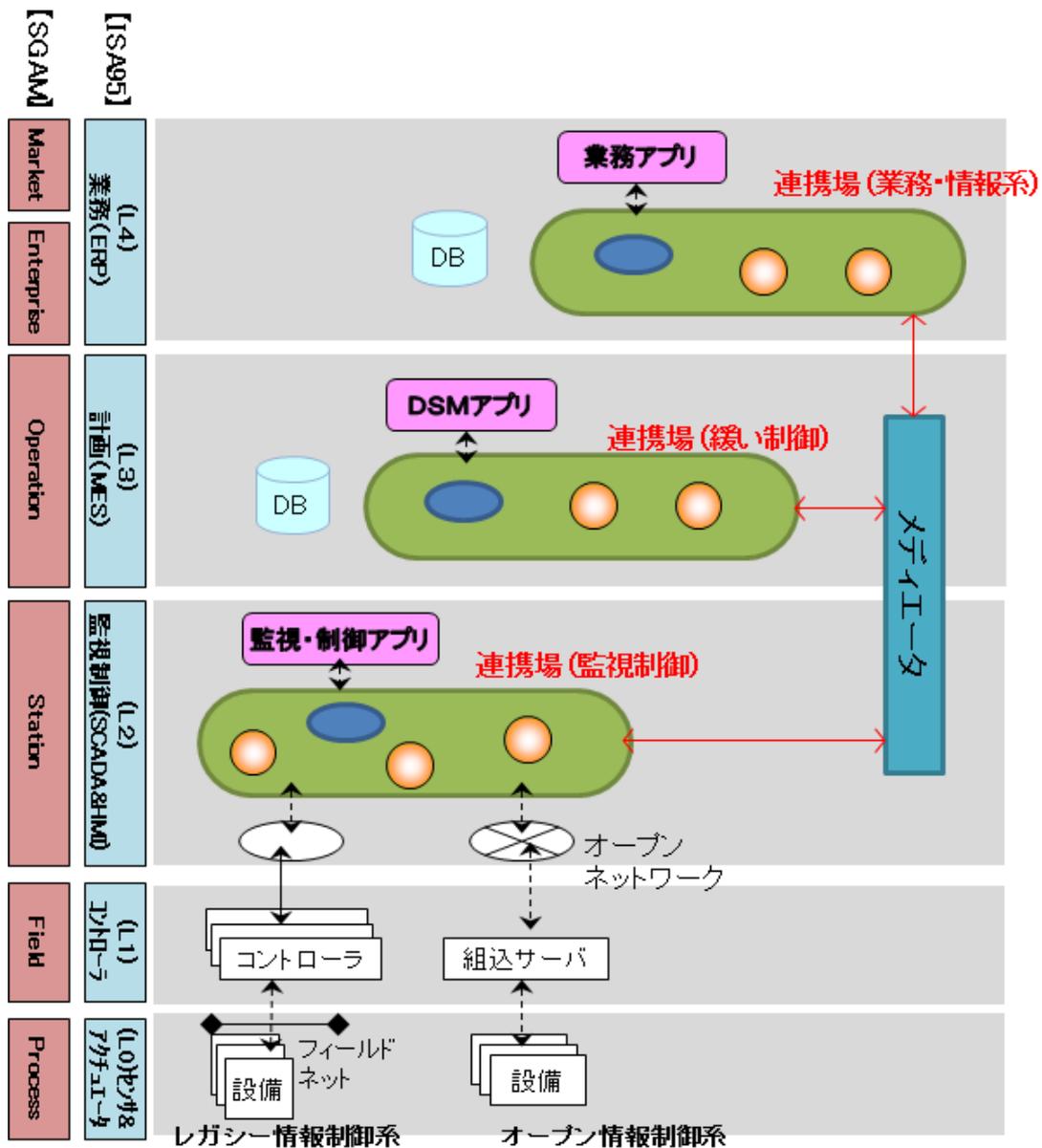


図 3-8 社会インフラシステムにおける連携場の階層化

### 3.2.1.3. グループの構成要素と実システムとのマッピング

3.2.1.1にて述べた本研究におけるグループの構成要素であるノードは、実システムにおけるソフトウェアにより提供される機能やデータ等にマッピングされる。ここで従来技術では、個々の実システムにおいて個別に準拠するインターフェース規格やデータモデルに関わらず、統一の形式・粒度によるインターフェースやデータモデルに基づくオブジェクトに一樣にマッピングして、連携する他システムに対して一樣なアクセスを提供させる場合が多い。ただしこれらの場合、システムのセキュリティ保護の観点からの外部からアクセス可能なシステム階層の違い、外部に公開なデータ及び機能の違いには対応することが困

難である。また外部との連携の影響を受けて、システム内部の本来の制御処理の性能・信頼性維持が困難となり得る。

そこで本研究では、グループの構成要素であるノードに対する実システムへのマッピングのレベルを規定し、システム毎の個別の連携要件に適したマッピングを提供する。本研究におけるグループの構成要素であるノードと実システムとのマッピングの指針を表 3-3 にまとめる。表 3-3 の項目#1~3 はノードに対するマッピング対象を選定するための指標であり、各システムにて設定される各項目のレベルに合致するものが選定される。表 3-3 の項目#4,5 はノードに対するマッピング対象の実行条件監視の指標であり、各システムにて設定される各項目のレベルを達成できない場合は、該当ノードを用いたサービス実行を中断して、マッピング関係を直ちに解消させる。

表 3-3 マッピングの指針

#	分類	項目	レベル (例)	補足
1	マッピング対象選定の指標	システム階層	Process, Field, Station, Operation, Enterprise, Market	SGAM の階層 (図 3-6、表 3-2 参照) で指定。
2		制御周期	小 (msec~sec) , 中 (sec~min) , 大 (min~)	システム階層との依存関係あり。
3		データアクセス	Read Only, Read/Write (計画情報) , Read/Write (制御コマンド)	
4	実行条件監視の指標	信頼性	性能保証重視、 トランザクション保証重視	指定レベルを達成できない場合は実行中断、マッピング解消。
5		リアルタイム性	ハード・リアルタイム <sup>1</sup> 、 ファーム・リアルタイム <sup>2</sup> 、 ソフト・リアルタイム <sup>3</sup>	指定レベルを達成できない場合は実行中断、マッピング解消。

図 3-9 にグループの構成要素であるノードと実システムとのマッピングの概要を示す。図 3-9 に示すようにインフラシステム (実システム) 毎にシステム運用ポリシーを規定、

<sup>1</sup>ハード・リアルタイム：規定時間内の処理完了を厳守しなければならない。

<sup>2</sup>ファーム・リアルタイム：規定時間内に処理が完了しなかった時点で価値が無くなり、結果は破棄される。

<sup>3</sup>ソフト・リアルタイム：規定時間内に処理が完了しない場合、徐々に価値が減少するが、処理は最後まで完了させる。

公開する。本ポリシーにて表 3-2 に挙げたマッピングの指針におけるレベルを定義する。なお各ノードにマッピングされる実システムにおけるソフトウェアプログラム（プロセス）のインタフェース、データモデル、制御周期等は個別に異なり得るため、これらの情報はプロファイルとして記載し公開するものとする。

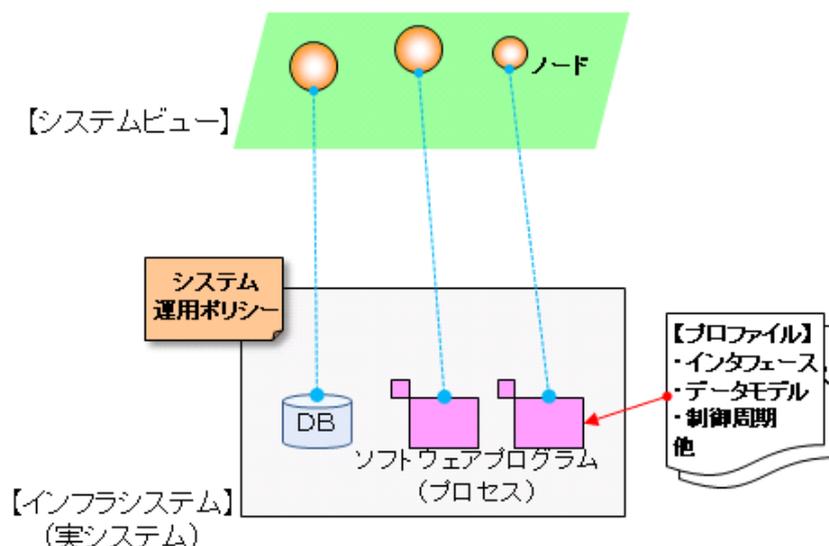


図 3-9 グループの構成要素と実システムとのマッピング

## 3.2.2. グループの制御・ライフサイクル管理

### 3.2.2.1. グループの動作原理

グループの動作原理の概要を図 3-10 に示す。図 3-10 では、3.2.1.1 にて定義したコンダクタ、ノード、モニタとノードにマッピングされる機能等を提供し、実際にシステム上で動作するソフトウェアプログラム等（Program）の構成で示している。サービス提供時にグループは以下に述べる(1)～(6)の一連の動作を行う。(1)コンダクタ及びモニタがサービスシナリオを読み込む。ここでコンダクタはサービス実行を制御するためにサービスシナリオを参照し、モニタはサービス実行に伴うノードの挙動を監視するためにサービスシナリオを参照する。(2)コンダクタはサービスシナリオに定義される条件に基づいてサービス実行に必要なノードをシステム内から検索し、該当するものでグループ化する。(3)コンダクタはサービスシナリオに定義される実行順序に従って(2)にて形成したグループに含まれるノードを呼び出しと実行結果の取得を行う。(4)(3)のノード呼出しに対して、実際に各ノードにマッピングされる Program が実行される。(5)モニタはサービス実行中に、サービスを構成する各ノードの実行状態等を監視する。(6)(5)の監視結果としてノードの異常を検出した場合は、コンダクタに対してグループの再構成の指示等を実施する。

上記の一連の動作における、サービスシナリオ及びグループの状態遷移については以降の節で詳細を述べる。

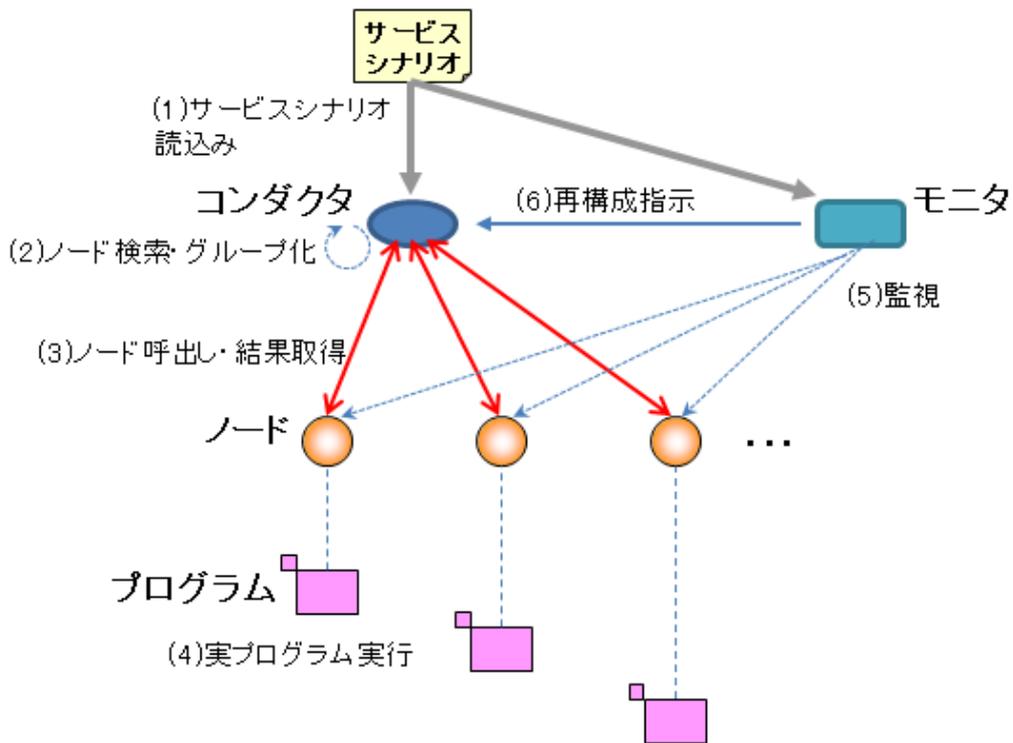


図 3-10 グループの動作原理の概要

### 3.2.2.2. サービスシナリオ

本研究におけるサービス実行に必要な機能またはデータ、トリガーとなるイベント及び実行条件、等を定義するサービスシナリオの主な内訳を表 3-4 にまとめる。これらの詳細は以降にて述べる。

表 3-4 サービスシナリオの主な内訳

#	項目	概要
1	必要ノード条件	サービス実行に必要な機能またはデータを提供するノードの種別、条件等を定義する。
2	実行開始条件	サービスの実行開始の条件を定義する。
3	実行順序	サービスを構成する1つ以上のノードの実行順序を定義する。
4	運転モード	サービスが動作する運転モード、条件等を定義する。

#### (1) 実行開始条件

実行開始条件のパターンとして、実行開始のトリガーによる実行に関する“実行条件”、周期実行に関する“周期”、キューへの格納順番に従った実行による“キュー”のいずれかを定義する。表 3-5 に実行開始条件の項目をまとめる。

表 3-5 実行開始条件

#	項目	設定情報	設定値・範囲	説明
1	実行条件	監視対象データ	参照するデータの識別情報	サービス（一連のノード実行）の開始のトリガーとして監視対象のデータの識別情報。複数の指定が可能。
2		条件式	実行条件発生の判定のための条件式	上項の監視対象データに対する実行条件発生の判定のための条件式に関する定義。
3	周期	周期時間	周期時間	サービス（一連のノード実行）の処理開始の周期。
4		周期維持不可時の対応方針	以下のいずれか、 ・途中中断 ・1周期分完了まで次周期実行省略 ・遅延許容で全処理実行	実行遅延等により周期維持が不可となった場合の、実行中及び実行待ちのノードの制御方針。
5	キュー	登録モード	以下のいずれか、 ・登録順 ・優先順番指定	キューに実行するサービスを登録する方法の選択。
6		順番	実行順番	サービスの実行順番。（優先順番指定の場合）

(2) 実行順序

実行順序のパターンとして、サービスを構成するノードを1つずつ順に実行する“順次”、サービスを構成する複数のノードを並列して実行する“並列”、複数のノード（もしくは一連のノード）の中から条件に従い選択されたものを実行する“選択”、1つ以上のノードを条件が成立する間はループ実行する“条件ループ”のいずれかを定義する。表 3-6 に実行順序の項目をまとめる。またこれらの詳細は以下に述べる。

表 3-6 実行順序

#	項目	設定情報	説明
1	順次	実行順序	順次実行するノードの順序を定義する。
2	並列	実行順序	並列実行するノードの順序を定義する。

#	項目	設定情報	説明
3	選択	条件判定データ	選択のための条件判定において参照するデータの識別情報を定義する。
4		選択の条件式	上項の参照データに対する選択条件判定のための条件式に関して定義する。
5		実行順序	選択毎の実行するノードの順序を定義する。
6	条件ループ	ループ条件判定データ	ループ継続の条件判定において参照するデータの識別情報を定義する。
7		条件式	上項の参照データに対するループ条件判定のための条件式に関して定義する。
8		実行順序	ループ内の実行するノードの順序を定義する。

図 3-11 に“順次”の概要を示す。サービスを構成するノードを1つずつ順序を維持して、順次（シーケンシャルに）実行していく。上記サービスを構成する各ノードの実行順序を定義する。

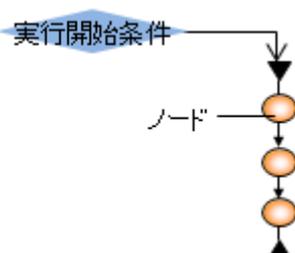


図 3-11 実行順序：順次

図 3-12 に“並列”の概要を示す。サービスを構成するノードを順番に並べた2つ以上の列を同時に並列して実行していく。なお並行する各列は開始のタイミングは合わせるが、処理中に列間での同期は実施しない。上記の並列実行する各列に含まれるノードの実行順序を定義する。

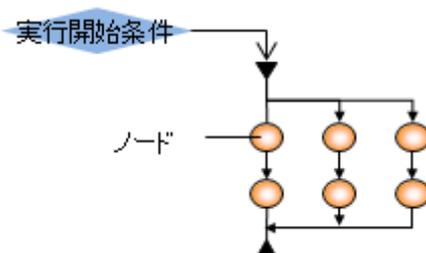


図 3-12 実行順序：並列

図 3-13 に“選択”の概要を示す。業務アプリケーションを構成する実行単位であるノードを順番に並べた2つ以上の列に対して、選択の条件に合致する列のノードを順番に実行する。上記の並列実行する各列に含まれるノードの実行順序と選択の条件判定を定義する。

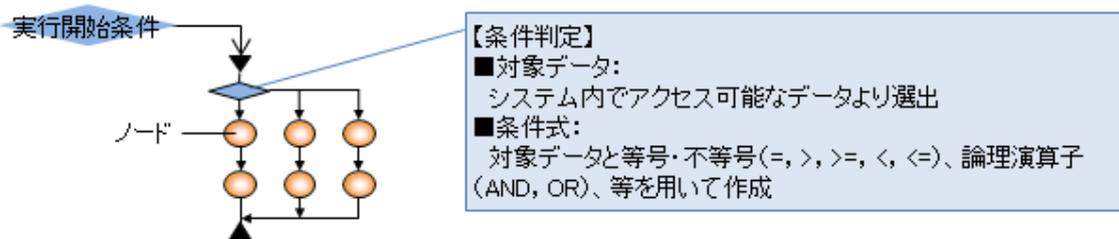


図 3-13 実行順序：選択

図 3-14 に“条件ループ”の概要を示す。サービスを構成するノードを順番に並べた列を、ループ条件が成立する間はループ実行する。上記のループ実行する列に含まれるノードの実行順序とループの条件判定を定義する。

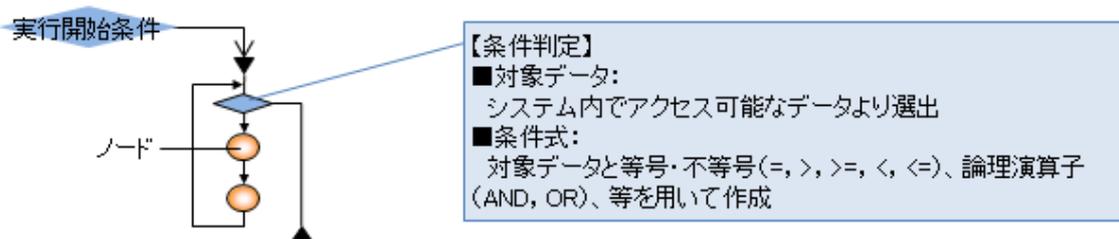


図 3-14 実行順序：条件ループ

### (3) 運転モード

サービス実行によるシステムの動作のモードとして、“本番モード”、“テストモード”、“オフラインモード”、等を定義する。その他、実行の優先順位、ログの出力レベル等を定義する。

#### 3.2.2.3. グループの状態遷移

本研究におけるグループの状態遷移を表 3-7 及び図 3-15 にまとめる。表 3-7 及び図 3-15 に示すように、サービス実行に伴うグループの状態として、“構築”、“正常”、“処理不具合”、“構成不具合”、“稼動不能”の5種類あり、サービス実行時に発生するイベントや対応内容に従い状態間で遷移する。

表 3-7 グループの状態遷移

#	状態分類	概要	オンライン動作
1	構築	グループのメンバとなるノードを選出、構成する状態。	
2	正常	グループが正常に稼動可能な状態。	継続（全稼動）
3	処理不具合	グループにおけるノードの処理実行の中で障害が発生する状態。	継続（遅延等有り）
4	構成不具合	グループにおいて利用不可の状態にあるノードが存在する状態。	継続（縮退もしくは全稼動）
5	稼動不能	障害等のためにグループが稼動不能な状態。	全停止

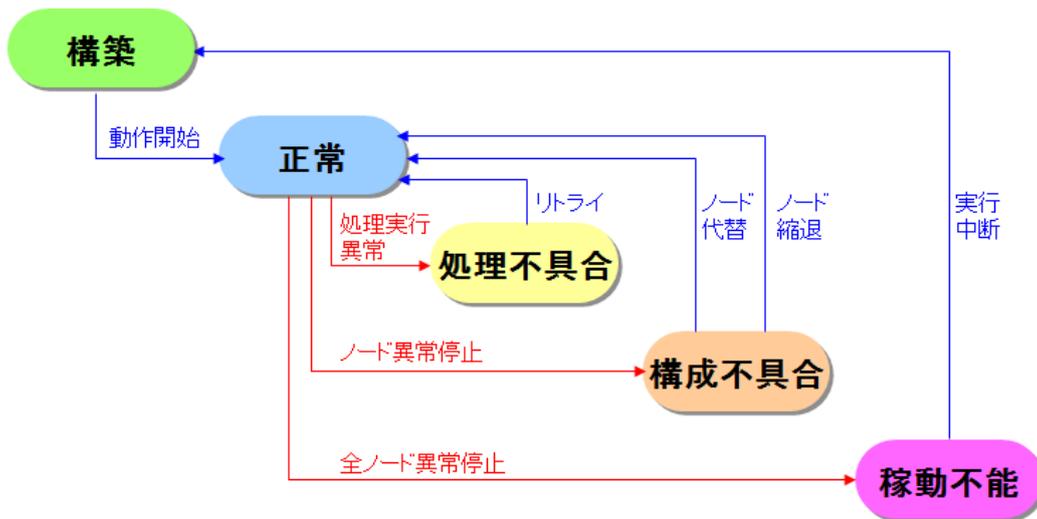


図 3-15 グループの状態遷移

システム状況の定義とグループ形成・制御の方針と、これらに対する表 3-7 及び図 3-15 に示したグループ状態遷移との対応を表 3-8 にまとめる。

表 3-8 システム状況の定義とグループ形成・制御の方針

#	分類	システム状況	グループ形成・制御の方針	グループ状態遷移
1	設計・構築時		事前に決められた範囲・対象に対してグループ形成。	構築→正常
2	契約時	異な 2 つ以上の主体間で	契約で決められた範囲・対象に対	構築→正常

#	分類	システム状況	グループ形成・制御の方針	グループ状態遷移
		連携範囲・方法決定。	してグループ形成。	
3	オンライン稼動時	高負荷発生等により性能劣化。	ボトルネック部分の切り離し、代替の探索、組込み。	処理不具合→正常
4		ICT システムの障害発生（異常処理含む）。	リカバリ実施。リカバリ不可なら代替の探索、組込み。	処理不具合→正常
5	保守時	システム拡張（ハードウェア、ソフトウェアの新規追加）。	新規追加分でグループの条件に合うものをメンバに追加。	構築→正常
6		システム更新（ハードウェア、ソフトウェアの更新・切替）。	更新後のシステムにて条件に従ってグループを再形成。	構築→正常
7	緊急時	業務レベルでの支障発生。	アプリケーションの入替。	構成不具合→正常
8		設備・機器の破損。	代替の探索。代替不可なら縮退実施。	構成不具合→正常
9		ICT システムの破損。	接続可能な範囲の限定とその範囲内でグループ再形成。（システム・グループの形成レベルの定義要）	構成不具合→正常

### 3.3. 異種ミドルウェア連携

#### 3.3.1. 異種ミドルウェア連携の位置付け

3.1 にて述べた異種システム連携のコンセプトを実現し、3.2 にて述べた動的グループ形成・制御の方法に基づいて、図 3-1 におけるシステムビュー上で動作するグループ及びグループの構成要素であるノードを用いてサービスを提供するためには、上記のノードとマッピングされる、実システムにおけるソフトウェアで実現する機能やデータ等を相互に連携させる必要がある。ここで各々のノードにマッピングされる、実システムにおけるソフトウェアで実現する機能やデータ等は、それぞれシステム内の異なる階層、範囲に位置し、それぞれ異なるミドルウェア上で実現され得る。これらの異なるミドルウェアを相互に連携させることで、上記の各々のノードにマッピングされる、実システムにおけるソフトウェアで実現する機能やデータ等を連携させる。これを異種ミドルウェア連携と呼ぶ。

本研究における異種ミドルウェア連携の位置付けを図 3-16 に示す。図 3-16 に示すように大きく 2 方向があり得る。同一のインフラシステム内で制御系から情報系までの相互接

続及び仲介と、異なる主体もしくは異なる業種のインフラシステム間での相互接続及び仲介である。なお2方向とも異種の標準規格間の相互接続があり得る。これらの詳細は以降にて述べる。

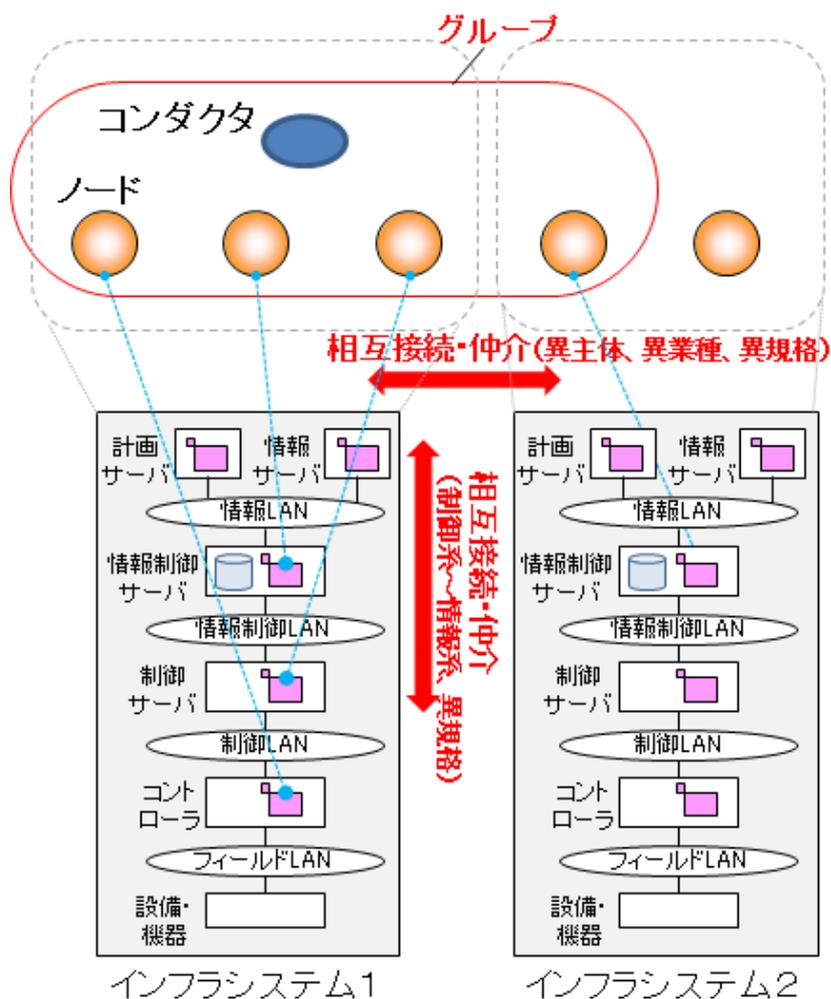


図 3-16 異種ミドルウェア連携の位置付け

### 3.3.2. システムの異種性

本研究が対象とするインフラシステムにおいて、準拠する標準規格・プロトコルや処理タイミング等はシステムの階層、範囲等により異なり得る。これらの標準規格・プロトコルを実装する各々の異なるミドルウェアを相互接続、連携するためには、これらの異種性を解決する必要がある。上記の解決すべき異種性について表 3-9 にまとめる。つまり図 3-17 に示すアプリケーション、データ、アクセスの3つの階層において、それぞれ静的 (Static) な異種性、動的 (Dynamic) な異種性を解決することが必要となる<sup>61)</sup>。ここで静的な異種性とは、システムにおける定数データ、単位系、準拠するインタフェース仕様・

データモデル等の時間に非依存な、扱う形式・構造の違いに起因する異種性であり、マッピング・変換により解決でき得る。また動的な異種性とは、アプリケーションの処理周期、データ発生頻度、通信プロトコルにおける処理シーケンス等の時間に依存した、システムの動作の違いに起因する異種性であり、動作制御等により解決する必要がある。

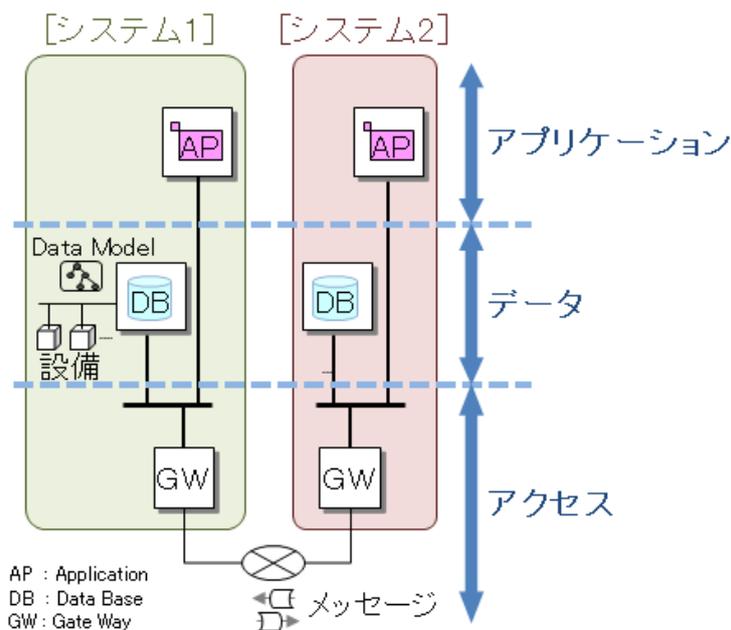


図 3-17 制御系～情報系のシステム連携の階層（筆者文献 61）より引用、一部加筆）

表 3-9 システム連携における異種性（筆者文献 61）より引用、一部加筆）

#	階層	静的な異種性	動的な異種性
1	アプリケーション	<ul style="list-style-type: none"> <li>サービス記述</li> </ul>	<ul style="list-style-type: none"> <li>処理周期</li> <li>起動タイミング・トリガー</li> </ul>
2	データ	<ul style="list-style-type: none"> <li>データモデル</li> <li>リソースプロファイル</li> </ul>	<ul style="list-style-type: none"> <li>発生タイミング</li> <li>頻度、データ量</li> </ul>
3	アクセス	<ul style="list-style-type: none"> <li>インタフェース</li> <li>プロトコル種別</li> </ul>	<ul style="list-style-type: none"> <li>異常時も含めたプロトコルにおける処理シーケンス</li> <li>マスタ/スレーブの挙動</li> <li>可能なアクセス範囲</li> </ul>

### 3.3.3. 規格間相互接続

#### 3.3.3.1. 静的な異種性の解決方法

表 3-9 で述べたシステム連携における、静的な異種性を解決するために、サービスの構成要素及び実行順序等に関する記述であるサービス記述（アプリケーション階層）の正規

化と、データモデルや設備・機器に関する情報であるリソースプロファイル（データ階層）の正規化や、アクセスするために使用するインタフェース（アクセス階層）のマッピング等、標準規格に着目した相互変換を行う。これらによりサービス実行の際の機器管理及び検索の一元化が容易となる。

### 3.3.3.2. 標準規格対応

システムの分野、階層、範囲毎に標準規格は多種多様なものが存在する。また個々の標準規格は仕様は固定ではなく、バージョンアップ等により常にアップデートし得る。一例として、IEC（TC57）による電力システム向けの標準規格の構成<sup>51)</sup>を図 3-18 に示す。

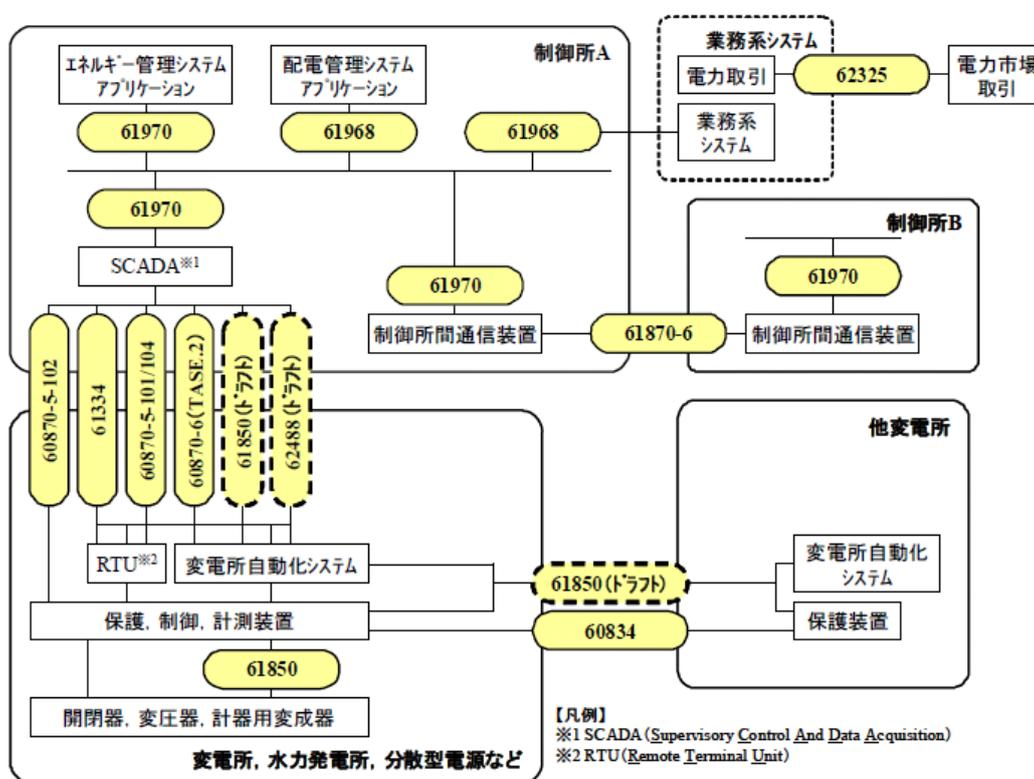


図 3-18 IEC (TC57) による電力システム向け標準規格の構成（参考文献 51）より引用

### 3.3.3.3. 3階層による相互変換

3.3.3.2にて述べたように、多種多様に存在し、アップデートし得る標準規格に対する規格間相互変換について述べる。標準規格は一般的に以下の3つの要素を規定している。

- A) データモデル : 共有するデータの共通的な意味付け
- B) インタフェース : 要求する処理内容の指定の仕方
- C) プロトコル : やり取りするための手続き・通信の様式

上記から標準規格をデータモデル、インタフェース、プロトコルで階層化して、階層毎に相互接続・変換を実施するものとする。概要を図 3-19 に示す。3つの階層では各々、以下の変換処理を実施する。各階層では外部定義による変換ルールの入替を容易とする。

- A) データモデル変換 : 内部データモデルと外部システムにて用いる規格 Common Information Model (CIM) との間の相互変換を実施。
- B) インタフェース変換 : 内部インタフェースと外部システムにて用いる規格インタフェースとの間の相互変換を実施。
- C) プロトコル変換 : 接続先の対応規格に準拠する手続き、通信を実施。

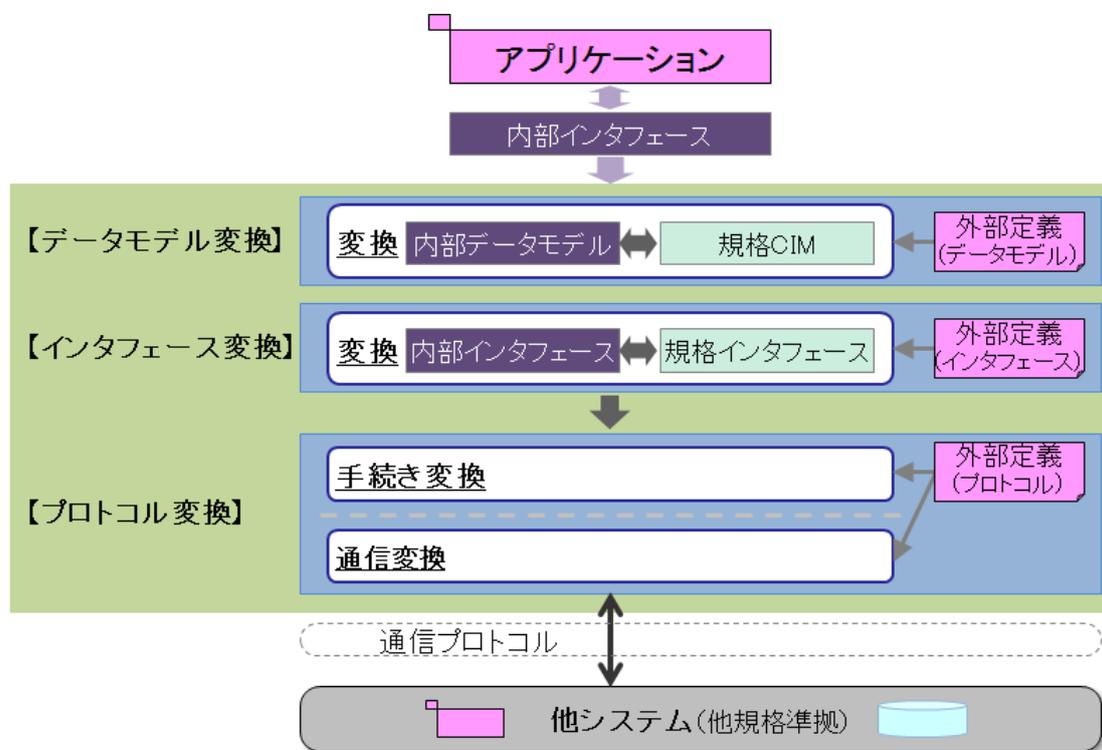


図 3-19 3 階層による相互変換

### 3.3.4. 仲介

#### 3.3.4.1. 動的な異種性の解決方法

表 3-9 で述べたシステム連携における、動的な異種性を解決するために、本研究ではシステム間相互接続時の動作仲介と、システム内の物理構成に依存しない論理的なグループ形成によるアクセス範囲仲介という2つのアプローチを取る。上記1つ目のアプローチは、システム間で接続してデータのやり取りを行う際に、システム毎に固有の動作の差異を吸収するための仲介・制御を行う。本アプローチにより、動的な異種性として、処理周期・

起動タイミング（アプリケーション階層）、データ発生タイミング・頻度（データ階層）、異常処理等も含めたプロトコルにおける処理シーケンス（アクセス階層）の違いを解決する。また2つ目のアプローチは、従来の物理構成に依存した1つ以上のネットワークセグメントを跨いで1つ以上のノードを含めて、ユーザが定義する関係に基づいて論理的なグループ（本研究では“スコープ”と定義）を形成し、該グループに含まれる範囲内のデータへのアクセスをユーザアプリケーションに提供する。本アプローチにより、システムを跨いで可変なアクセス範囲（アクセス階層）の違いを解決する。

以降では上記のアプローチにより、特に表 3-9 で述べた異種の系に各々導入される異種ミドルウェアの接続に必要な動的な異種性を解決するための方法について述べる。動的な異種性に対するアプローチ及び解決方法を表 3-10 にまとめる。動的な異種性に対する解決方法として、アプリケーションの処理周期・起動タイミング、データ階層のデータ発生トリガー等に対するタイミング仲介（Timing Intermediation）、データ層のデータ発生頻度等に対する時間的粒度仲介（Time Series Granularity Intermediation）、アクセス層の、異常時も含めたプロトコルの処理シーケンス等に対する手続き仲介（Procedure Intermediation）を挙げる。これらはエージェント及びアダプタにより実現する。

またアクセス層のシステムを跨いで可能なアクセス範囲に対するアクセス範囲仲介（Access Range Intermediation）は、システムを跨いで可変なアクセス範囲を提供する論理的なグループとしてのスコープとして実現する。詳細を以降で述べる。

表 3-10 動的な異種性の解決方法

#	階層	動的な異種性	アプローチ	解決方法
1	アプリケーション	• 処理周期	動作仲介（エージェント・アダプタ）	タイミング仲介（Timing Intermediation）
2		• 起動タイミング		
3	データ	• データ発生トリガー		時間的粒度仲介（Time Series Granularity Intermediation）
4	• データ発生頻度 • データ量			
5	アクセス	• 異常時も含めたプロトコルにおける処理シーケンス		手続き仲介（Procedure Intermediation）
6		• マスタ/スレーブの動作		
7		• 可能なアクセス範囲	アクセス範囲仲介（スコープ）	アクセス範囲仲介（Access Range Intermediation）

### 3.3.4.2. エージェント・アダプタ

表 3-10 に挙げた動作仲介を実現するための、上述の3種類のエージェントによる動作を図 3-20 に示す。

タイミング仲介は、異なる系間で処理開始やデータ発生等の事象発生タイミングの違いを仲介するものであり、アプリケーションの要求時に常に応答するための(1)バッファリング、事象が発生したタイミングでアプリケーションに通知するための(2)登録・発行(Publish / Subscribe)により対応する。

時間的粒度仲介は、異なる系間での時間軸で見たデータの発生頻度・量の違いを仲介するものであり、アプリケーションにとって適切な頻度、量でデータを提供するための(3)フィルタリング、及び(4)統合・分割により対応する。

アクセス範囲仲介は、系毎のアクセス可能な範囲の変化を吸収し、アクセス先及び範囲の指定方法の系毎の違いを仲介するものであり、系間で通信の中継を行う(5)中継及びスコープにより対応する。スコープの詳細は4.6.2.4にて述べる。

ここで(1)バッファリングは、ある系内を流れるデータを蓄積しておき、他系からの要求に対して応答性能を保証してデータを提供する。ここで扱うデータサイズが大きいほど、また系間で処理開始タイミングやデータ発生頻度の差が大きいほど、バッファリング用領域のサイズは大きくしておくべきである。(2)登録・発行は、ある系にてデータを受け取るタイミング(イベントまたは周期)を登録しておくこと、他系に対してデータを上記の登録タイミングにて提供する。(3)フィルタリングは、ある系からのデータを条件に従って間引いてから他系に提供する。(4)統合・分割は、ある系からのデータをある範囲で収集、統合してから他系に提供する、または反対にある系からのデータを分割して他系に提供する。また(3)(4)を用いることで、連携に伴うトラフィック量増大によるシステムまたはアプリケーションへの負荷増を回避できる。

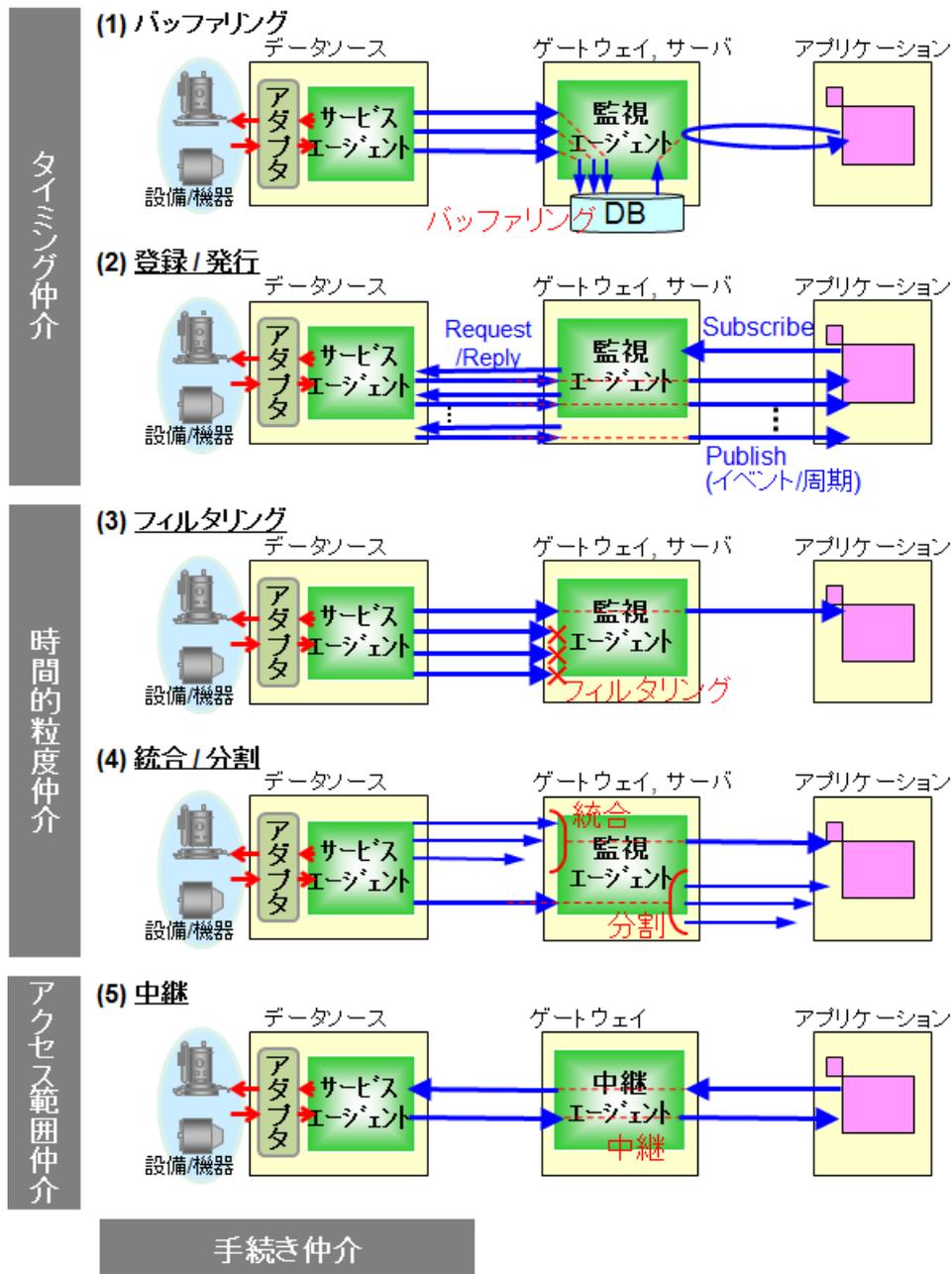


図 3-20 エージェント・アダプタを用いた動作仲介 (筆者文献 61) より引用、一部加筆)

手続き仲介は、データソースまたは系毎の準拠するプロトコルの処理シーケンス等の手続きの違いを仲介するものであり、アダプタにより対応する。

### 3.4. 必要な要素技術

本研究によるシステムを実現する上で必要な主要な要素技術について述べる。特に本研究によるシステムまたはアプリケーションの動作を実施する上での機能要件として必要な技

術と、システムを運用する上での非機能要件として必要な技術とに分けて述べる。

なお機能要件とは、業務を実現するにあたり、システム側に作業分担させたい処理部分の要件のことである。また非機能要件とは、機能を快適、あるいは安全、確実に実現するための要件のことである。

### **3.4.1. 機能要件対応**

#### **3.4.1.1. システム間ネゴシエーション**

オンライン稼働中に異主体システム間で連携の可否、連携手段、連携範囲・期間等について、情報のやり取りによるネゴシエーションを行う。

#### **3.4.1.2. 状態監視**

自システム内の設備もしくはアプリケーションの稼働状態、連携する他システムの状態、他システム内の設備もしくはアプリケーションの稼働状態の監視を行う。なお他システムに関しては、情報の公開範囲が制限され得る、自システムとは監視対象項目、情報粒度、状態定義、等が異なり得る。このため状態情報の翻訳、粒度調整、等が必要となる。

#### **3.4.1.3. アクセス制御**

システム内で管理するデータもしくは設備への、自システム内または外部システムからのアクセスを管理、制御する。

#### **3.4.1.4. 通信経路制御**

アプリケーションを実行する上で必要な性能要件、信頼性要件を達成するために、自システム内、システム間の通信に関して、経路制御・切替、通信の優先制御、帯域制御、等を行う。

#### **3.4.1.5. 機器管理・ユーザ管理**

システム内の機器・設備の管理、システムの提供するサービスを利用するユーザの管理を行う。

### **3.4.2. 非機能要件対応**

非機能要件として、独立行政法人情報処理推進機構（IPA）が定義する非機能要求<sup>48)</sup>の分類を表 3-11 に挙げる。本研究にて扱う異種システム間の連携の観点から、表 3-11 の中から、特に「運用・保守性」と「セキュリティ」に着目する。「運用・保守性」や「セキュリティ」に関しては、通常は1つのシステム内に閉じて実施される。一方、本研究ではシステムを跨っての運用を前提とするが、定常的な連携とは限らない、連携先のシステムは状況に応じて変更し得ることを考慮して、以降にて新たな対応を検討する。

表 3-11 非機能要求グレードの6大項目 (参考文献 48) より抜粋)

#	要求項目	説明	要求の例
1	可用性	システムサービスを継続的に利用するための要求	<ul style="list-style-type: none"> <li>運用スケジュール (稼動時間・停止予定等)</li> <li>障害、災害時における稼動目標</li> </ul>
2	性能・拡張性	システムの性能、及び将来のシステム拡張に関する要求	<ul style="list-style-type: none"> <li>業務量および今後の増加見積もり</li> <li>システム化対象業務の特性 (ピーク時、通常時、縮小時等)</li> </ul>
3	運用・保守性	システムの運用と保守のサービスに関する要求	<ul style="list-style-type: none"> <li>運用中に求められるシステム稼動レベル</li> <li>問題発生時の対応レベル</li> </ul>
4	移行性	現行システム資産の移行に関する要求	<ul style="list-style-type: none"> <li>新システムへの移行期間および移行方法</li> <li>移行対象資産の種類および移行量</li> </ul>
5	セキュリティ	情報システムの安全性確保に関する要求	<ul style="list-style-type: none"> <li>利用制限</li> <li>不正アクセスの防止</li> </ul>
6	システム環境・エコロジー	システムの設置環境やエコロジーに関する要求	<ul style="list-style-type: none"> <li>耐震/免震、重量/空間、温度/湿度、騒音等、システム環境に関する事項</li> <li>CO<sub>2</sub> 排出量や消費エネルギー等、エコロジーに関する事項</li> </ul>

#### 3.4.2.1. 運用・保守性: 連携システム間障害解析支援

本研究が対象とするシステム内には複数の計算機が導入され、これらの計算機間でデータや処理の流れが発生する。また他システムとも連携して、システムを跨いでデータや処理の流れが発生する場合もある。さらにこれらのデータや処理の流れは可変であり得る。通常、ソフトウェアや通信等のログは計算機毎、ソフトウェア毎に個別の形式で個別に管理されている。このため本研究によるシステム間の連携動作において障害が発生した場合、解析作業及び原因特定が困難となる。

上記から障害解析等のためにこれらのシステム間連携においてトレーサビリティを保証する必要がある。この際の要件及びアプローチを表 3-12 にまとめる。また表 3-12 におけるアプローチの概要を図 3-21 に示す。なお図 3-21 中の A)B)は表 3-12 中の A)B)に該当する。

表 3-12 連携システム間のトレーサビリティの要件

#	要件	アプローチ
A)	システム・レイヤを跨いでのデータ・メッセージの流れの追跡容易化	局所的な紐付け情報を相互に関係付けて、数珠つなぎによるデータ・メッセージ追跡 (システムにて統一の紐付け情報無し)
B)	ユーザ視点に基づく粒度のビュー (パケット単位～メッセージ) の提供	パケット～メッセージの関係付け管理

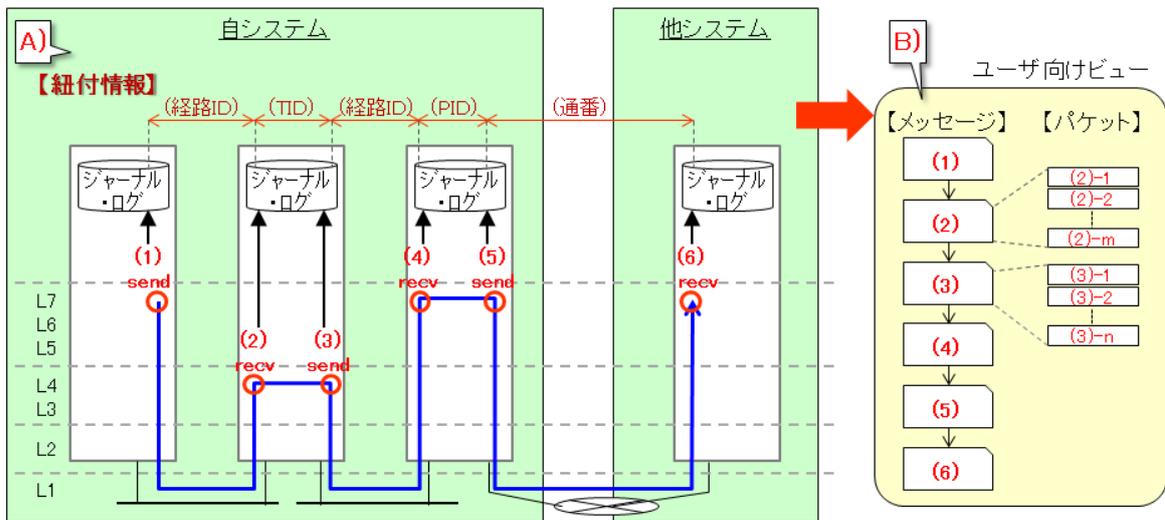


図 3-21 連携システム間トレーサビリティ

### 3.4.2.2. セキュリティ

本研究が対象とする情報制御システムにおいて、システム間連携を実施する場合に着目して想定されるセキュリティ脅威の例 (①～⑫) を図 3-22 に示す。

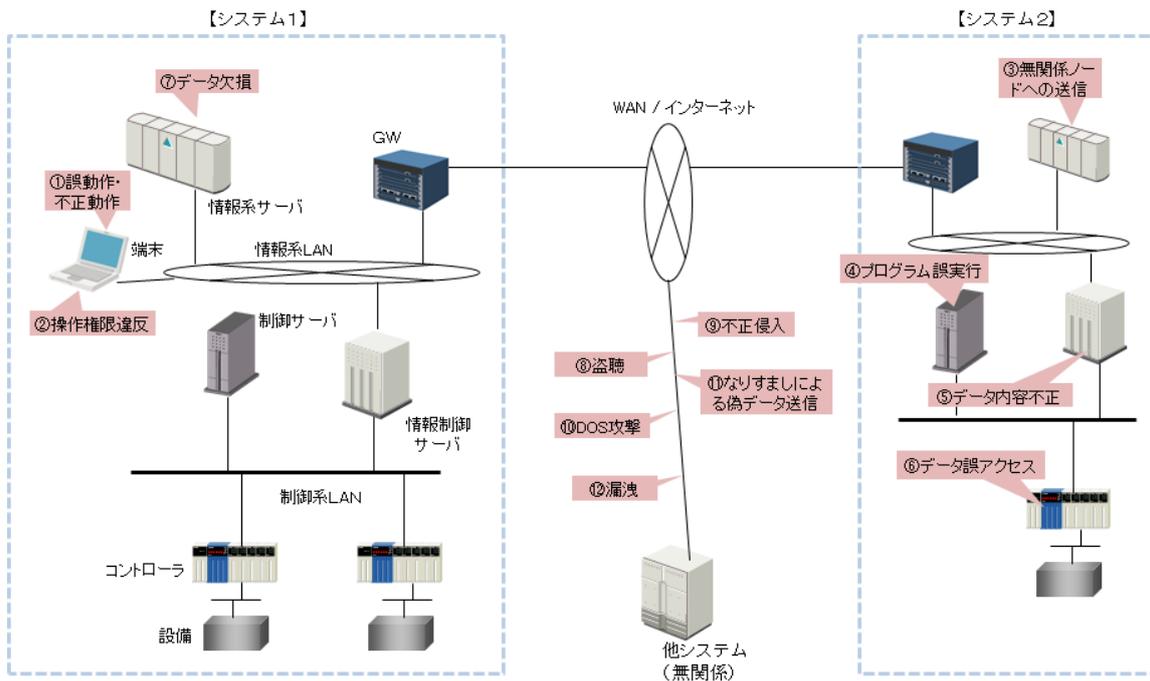


図 3-22 対象システムにおける脅威の例

図 3-22 に示すようなセキュリティ脅威に対して、以下に挙げる A)～D) のセキュリティ方針に従って対応する。

- A) 他システムとの接続に関して、相手がホワイトボックスである場合、相手側の動作を確認して対応する。（プロトコルによる処理実行成否確認、ログ等）
- B) 他システムとの接続に関して、相手がオープン対応（ブラックボックス）である場合、証明書等の情報システムにおける一般的な方法で対応する。
- C) 他システムとの接続箇所（Gate Way: GW 等）では、内部の状態・動作を監視して、異常があれば即座に停止させる。
- D) 各システム内では、ホワイトリストを作成し、リスト外のもの（計算機、プログラム）は除外する。（システム内で稼動する計算機、プログラムは事前に登録されたもののみと仮定）

### 3.5. システム要件

3.1 にて述べた異種システム連携のコンセプトを実現し、3.2～3.4 にて述べた技術を導入するシステムを構築するためのシステム要件を以下のように定める。

- (1) 本システムにおけるサービスは、1つ以上のインフラシステムにおける1つ以上の機器及びソフトウェアプログラムを連動して提供されるが、サービスの作成及び実行は、

実際のインフラシステム、機器及びソフトウェアの個別の仕様及び状態を意識させない。

- (2) 本システムにおけるサービス実行のために、必要な情報のやり取りにより異なるインフラシステム同士を連携させる。この場合、情報のやり取り及びシステムの連携は、必要最小限の範囲内で実施する。
- (3) 稼働中の既設インフラシステムを連携させる場合に、該インフラシステムでの本来の業務における処理の性能及び信頼性を保証する。

### 3.6. 異種システム連携の境界条件

3.5にて述べたシステム要件(3)から、システム連携時も連携元のインフラシステムの性能及び信頼性を保証するためには、異種のシステム間で常に連携を許容するのではなく、条件を明確にして連携可否や連携の度合いを判断する必要がある。

ただし3.3.2にて述べたシステムの異種性を解決して、異種システム間で接続、連携する場合、システム間の連携の可否、連携の実施度合いは、実際の個々のシステムの構成やサービス容等により異なり得る。よって本節では異種システム間の連携の可否、連携の実施度合いを評価するための境界条件について規定し、境界条件の判定例について述べる。

#### 3.6.1. 境界条件の定義

異種システム間連携における境界条件を、以下に定義する。

- 異種システム間で連携を実施することが可能か否かの境界を判定するための条件。
- インフラシステムの安定稼働を保証するため、所定の条件を満たす場合のみに連携実施可能とする。
- 以下の2つの観点を含める。

(1) 非機能要件 : 異種システム連携による機能を安全、確実に実現するための要件

(2) 運用要件 : 異種システム連携により実施される業務運用を円滑に実施するための要件

上記の境界条件を用いての異種システム間連携の可否判定は、境界条件の全項目において、指定されたレベルを超えている場合に連携可とする。なお各条件項目のレベルは、業務やシステム毎の要件に基づいてサイト毎に個別に設定されるものとする。

以降では境界条件に含める2つの観点について、それぞれ詳細に述べる。

(1) 非機能要件

異種システム連携により実施される機能を安全、確実に実現するための要件である。その全ての条件項目においてユーザが指定するレベルを超えていることが、連携を実施するために必要となる。非機能要件による境界条件を表 3-13 にまとめる。非機能要件の分類毎の指標とユーザが指定するレベルの例をまとめている。これらは一般的なシステムにおける非機能要件<sup>48)</sup>から、本研究にて提案する異種システム連携の内容を踏まえて再整理したものである。

表 3-13 境界条件（非機能要件）

#	分類	指標	説明	レベル（例）
1	信頼性	機能充足	• 連携先が要求する機能・データを有しているか	• 全て有す，一部有す，無し
2		トランザクション保証	• 連携先が要求する一連の処理を実行可能か • 処理失敗時は元の状態に戻すことが可能か	• YES, NO • YES, NO
3		可用性	• 連携先と継続的に連携可能か	• YES, NO
4	性能	応答性能保証	• 連携先への要求に対する応答が要求時間内か	• YES, NO
5		リアルタイム性	• 連携先とのやり取りが要求するレベルでのリアルタイムに実行可能か	• ハード・リアルタイム，ファーム・リアルタイム，ソフト・リアルタイム
6	セキュリティ	正当性	• 連携先の正当性は保証できるか	• YES, NO
7	セキュリティ	アクセス範囲	• 連携先における外部（連携元）からのアクセス可能な範囲 • 連携先における外部（連携元）からの実行可能な操作	• 範囲指定 • Read Only, Read/Write, Create, Delete
8		秘匿性	• 連携先との間でのやり取りが秘匿されるか	• YES, NO

(2) 運用要件

異種システム連携により実施される業務運用を円滑に実施するための要件である。その全ての条件項目においてユーザが指定するレベルを超えていることが、連携を実施するために必要となる。運用要件による境界条件を表 3-14 にまとめる。運用要件の分類毎の指標とユーザが指定するレベルの例をまとめている。これらは一般的なシステムにおける運用

要件から、本研究にて提案する異種システム連携の内容を踏まえて再整理したものである。

表 3-14 境界条件（運用要件）

#	分類	指標	説明	レベル（例）
1	業務形態	運用期間	業務としての運用期間（時間帯、開始日時～終了日時）	・期間内，期間外
2		業務内容	連携による業務内容として許容されているもの	・見える化/データ収集・分析，オペレーション，直接制御
3		業務フェーズ・規定範囲	業務フェーズ・システム状況とその際の連携の規定範囲	<ul style="list-style-type: none"> <li>・構築時：設計段階で事前に決められた範囲内</li> <li>・契約時：契約で決められた範囲内</li> <li>・運用時：サービス維持のために規定内で可能な範囲内</li> <li>・緊急時：異常状況に応じ早期復旧のため可能な範囲内</li> </ul>
4	保守性	追跡性	連携に伴う自システム及び他システムの一連の処理内容・履歴が記録されるか（証跡）	・証跡有り，証跡無し
5		責任分界点	連携に伴う処理の実行保証または障害発生時の対応における、自システム及び他システムの責任範囲	・自他システムの責任範囲が明確，責任範囲が不明確

### 3.6.2. 境界条件の判定例

異種システム間連携における境界条件（非機能要件、運用要件）の位置付けと判定例を図 3-23 に示す。あるシステム（システム2）におけるユーザの操作によりアプリケーションが他システム（システム1）のデータを取得する場合のフローを例示している。さらに上記フローに対して境界条件の各項目（表 3-13 の非機能要件#1～8 及び表 3-14 の運用要件#1～5）を考慮すべき箇所にマッピングし、各項目の設定例を示している。

ここで本研究にて提案する異種システム連携技術を適用して、異種のシステム間で実際に連携実施可能か否かを判定するためには、上述の境界条件の全項目に対して充足しているか否かを評価する必要がある。ただし個々の境界条件については一様な基準があるわけ

ではなく、その程度は個々のシステムの仕様や業務内容等に依存し、条件の設定には個別の調整が必要となる。

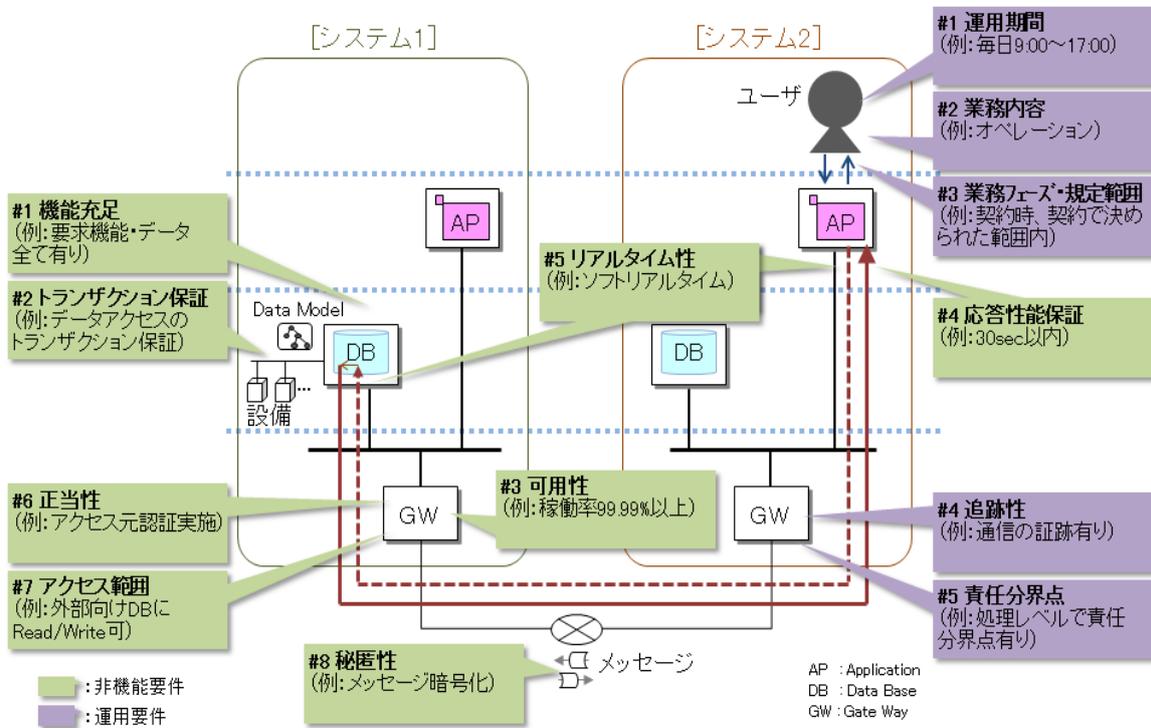


図 3-23 異種システム連携の境界条件の判定例

### 3.7. システムリファレンスアーキテクチャ

3.5にて挙げた3つのシステム要件を満たすための本研究にて提案するシステムのリファレンスアーキテクチャを図 3-24に示す。図 3-1にて示した異種システム連携コンセプトにおける3つの階層「サービス」、「システムビュー」、「インフラシステム」のうち、特に「システムビュー」の階層に対して、上記のシステム要件を満たすための構成要素及び動作を導入する。「サービス」及び「インフラシステム」はそれぞれ3.1.2、3.1.4にて述べたものである。

3.5にて挙げたシステム要件(1)に対して、「サービス実行基盤」を導入する。サービス実行基盤では、サービスシナリオの管理と、サービスシナリオに従ったサービスの実行管理及びサービスに対するシステムの見え方であるビューの提供を行う。

3.5にて挙げたシステム要件(2)に対して、3.2.1にて述べた「連携場」を導入する。連携場では、インフラシステムに紐づく1つ以上のノード及びコンダクタがサービス実行のために動作する。なお連携場は性能要件・信頼性要件が厳しい順に監視制御向け連携場、緩い制御向け連携場、業務・情報系向け連携場の3階層とする。監視制御向け連携場は3.2.2.1にて述べたISA-95モデルにおけるL2(監視制御)及び3.1.4にて述べたSGAMモ

モデルにおける Station、緩い制御向け連携場は ISA-95 モデルにおける L3（計画）及び SGAM モデルにおける Operation、業務・情報系向け連携場は ISA-95 モデルにおける L4（業務）及び SGAM モデルにおける Enterprise/Market に位置する。これらの各階層の連携場間のやり取り及び各連携場とサービス基盤との間のやり取りはメディエータを介して行う。

3.5 にて挙げたシステム要件(3)に対して、「仲介」を導入する。仲介は、インフラシステムを保護して連携場に接続するためのものであり、連携場に対して公開可能な情報の管理と、サービス実行に使用されるノードに対するインフラシステム内の実際の構成要素とのマッピングを行う。

なおサービス実行に伴い、サービス実行に必要な1つ以上のノードに紐付くインフラシステムに対して「連携場」及び「仲介」を介して計画等の情報が伝達される。本情報に対する動作結果または状態は実績情報として開示され、「仲介」及び「連携場」を介してサービス実行基盤側に伝達される。またサービス実行に伴う異なるインフラシステム間の情報のやり取りも「連携場」及び「仲介」を介して実施される。

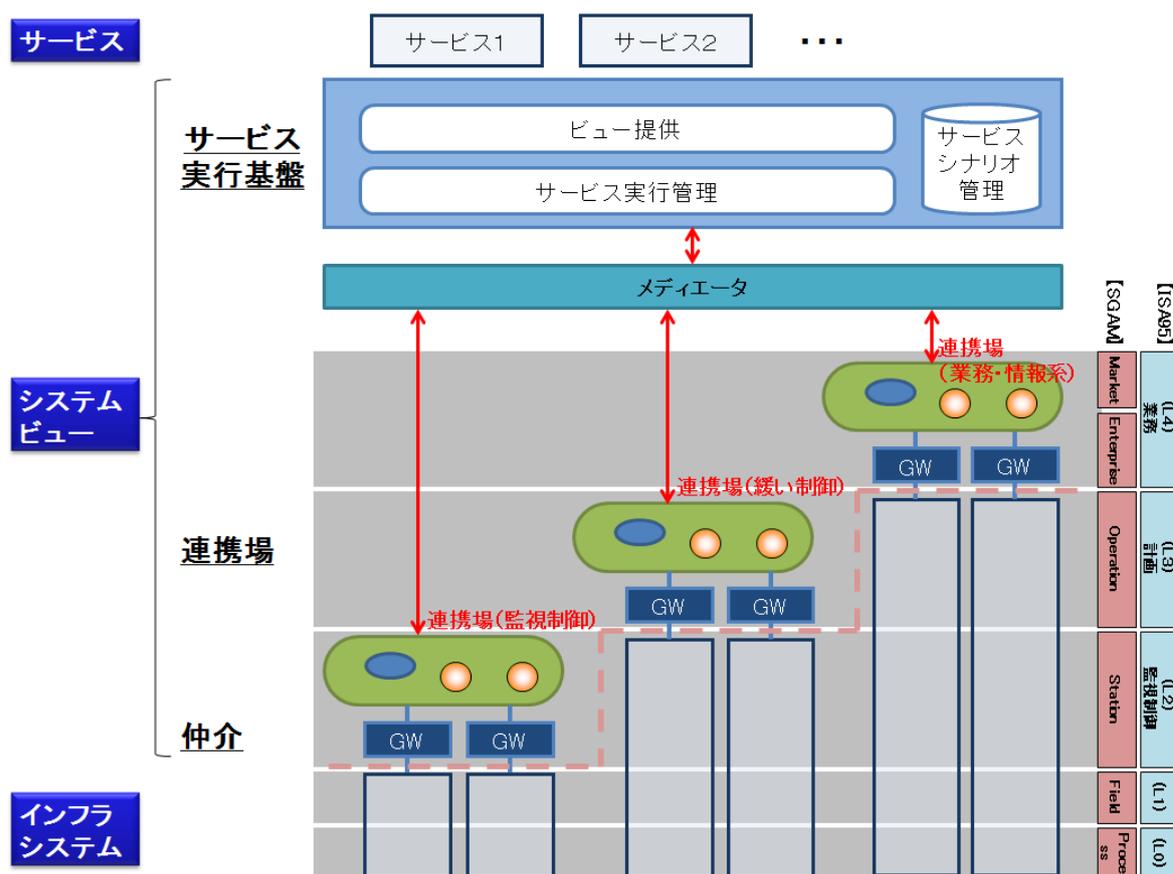


図 3-24 リファレンスアーキテクチャ

## 4. システムの構築

### 4.1. 位置付け

本章では、第3章にて述べた、本研究にて提案する異種システム連携技術を実装し、実際のシステムを構築するための方式について述べる。ここでは本研究に基づくシステムを構成する上で想定するソフトウェア構成及びハードウェア構成について述べる。また特に上記のソフトウェア構成において実現すべき機能を挙げて、その実現方式について述べる。

ただしその実現すべき機能の全てが新規ではなく、個々の機能及び技術の中には既存のものを有効活用できるものもある。ゆえに本章では特に本研究の特長となる、「動的グループ形成・制御」、「異種ミドルウェア連携」の実現方式について具体的に述べる。また既存方式・技術も活用してその他の機能を実現し、それらを組み合わせて、本研究に基づくシステムを実現する方法について述べる。ここで本研究では既存のインフラシステムの連携による新しいサービスの柔軟な提供を目的としているが、この新サービス提供時も各々のインフラシステムにおける既存業務に影響を与えないことが前提である。またインフラシステムは一般的に規模が大きいため、運用としてシステムを構成するハードウェア及びソフトウェアの段階的な拡張及びリプレースが前提となる。ゆえに上記の既存方式・技術も活用した、本研究に基づくシステムの実現はこれらの前提を考慮したものとなる。

また「異種ミドルウェア連携」の一部について、実現方式及び実施手段の観点から有用性を評価する内容及び結果について述べる。

### 4.2. 想定するソフトウェア構成

3章にて述べた異種システム連携のコンセプト及びシステムのリファレンスアーキテクチャを実現するためのソフトウェア構成を図4-1に示す。



図 4-1 ソフトウェア構成

### 4.3. 想定するハードウェア構成

3 章にて述べた異種システム連携のコンセプト及びシステムのリファレンスアーキテクチャを実現するためには、表 4-1 に挙げた、ネットワーク型、DB サーバ型、クラウド型の、IT システムアーキテクチャにおいて典型的な 3 つのハードウェア構成パターンへの適用が考えられる。表 4-1 にはハードウェア構成の各パターンの概要、長所、短所をまとめる。

表 4-1 ハードウェア構成のパターン

	ネットワーク型	DB サーバ型	クラウド型
概要			
長所	<ul style="list-style-type: none"> <li>既設システムの延長としての構成が容易</li> </ul>	<ul style="list-style-type: none"> <li>大量データの一元管理が容易</li> </ul>	<ul style="list-style-type: none"> <li>スモールスタートが容易</li> </ul>

	ネットワーク型	DB サーバ型	クラウド型
短所	・信頼性・性能保証のための設備投資が発生	・DB サーバがシステムのボトルネック箇所となる	・複数拠点間で連携する場合に性能限界あり

#### 4.4. 想定する機能配置

3.7にて述べた異種システム連携を実現するためのシステムのリファレンスアーキテクチャを、4.1にて述べた想定するハードウェア構成の3つのパターンにて実現する。このための3.7にて述べたシステムリファレンスアーキテクチャを構成するサービス実行基盤、連携場、仲介の各機能の配置を表4-2にまとめる。ネットワーク型では、サービス実行基盤及び連携場を構成するための機能はサービスシステム側（GW）に分散配置され、連携場及び仲介を構成するための機能は各インフラシステム側（GW）に分散配置される。DBサーバ型では、サービス実行基盤を構成するための機能はサービスシステム側に、仲介を構成するための機能は各インフラシステム側に配置され、連携場を構成するための機能はDBサーバに集約される。クラウド型では、サービス実行基盤、連携場、仲介の各々を構成するための全ての機能がクラウドに集約される。

表 4-2 機能配置のパターン

	ネットワーク型	DB サーバ型	クラウド型
構成	<p>サービスシステム、サービスシステム、サービス実行基盤、サービス実行基盤、ネットワーク、GW、連携場、仲介、インフラシステム、インフラシステム</p>	<p>サービスシステム、サービスシステム、サービス実行基盤、サービス実行基盤、DBサーバ、連携場、DB、仲介、インフラシステム、インフラシステム</p>	<p>サービスシステム、サービスシステム、クラウド、サービス実行基盤、連携場、DB、仲介、インフラシステム、インフラシステム</p>
特徴	・専用 GW を準備することで拡張性、信頼性確保	・データ及び共通機能の集約	・サービス提供によるスモールスタート容易

#### 4.5. 実現すべき機能一覧

図 4-1 に示したソフトウェア構成における機能の一覧を表 4-3 にまとめる。これらの各機能の詳細は以降の節で述べる。

表 4-3 機能一覧

大分類	中分類	小分類	説明	
サービス実行基盤	1	ビュー提供	本システムにて実行されるサービスに対して、利用可能なシステムの要素を参照、制御するためのインタフェースを提供する。	
	2	サービスシナリオ管理	本システムにて実行されるサービスに関するシナリオの管理を行う。	
	3	サービス実行管理	本システムにおけるサービスシナリオに基づくサービスの実行管理を行う。	
	4	動的グループ形成・制御	グループメンバ管理・制御	グループ形成及びグループのメンバとなるノードの管理と制御を行う。
	5		グループライフサイクル管理	作成したグループの状態及び状態遷移に応じて必要なグループの制御を行う。
	6		状況判断	作成したグループの状態及び状態遷移と、システムの状態を監視する。
	7		引継ぎ・リカバリ	グループ及びシステムの状態に応じたグループ間の実行状態の引継ぎ、グループメンバの再構成、等を行う。
	8		グループポリシー管理	グループの作成及び制御に関するポリシーを管理する。
連携場	9	ネゴシエーション	異種システム間で、連携の可否、連携手段、連携範囲・期間等について、情報のやり取り、交渉を行う。	
	10	基盤間連携動作制御	通信経路制御	自システム内、システム間の通信に関して、経路制御・切替、通信の優先制御、帯域制御、等を行う。
	11		状態監視（システム/設備）	自システム内の設備もしくはアプリケーションの稼動状態、連携する他システムの状態、等の監視を行う。

大分類	中分類	小分類	説明	
	12	ジャーナル・ログ管理	連携場での通信のデータ内容、送信元及び受信先、実行成否、発生イベント内容、等の情報を記録及び保持する。	
	13	通信	送信・受信インタフェース	連携場における通信（送信及び受信）のためのインタフェースを提供する。
	14	構成管理		連携場を構成するシステムの構成管理を行う。
	15	運用管理		連携場を構成するシステムの運用管理を行う。
	16	セキュリティ	アクセス管理	連携場にアクセスするシステムの管理を行う。
	17		認証	連携場にアクセスするシステムの正当性を確認するため認証を行う。
	18	異種ミドルウェア連携	論理グループ形成	物理構成に依存しないノードの論理的なグループの形成及び管理を行う。
	19		エージェント	個々のシステムへのアクセス代行、監視、システム間の中継を行う。
	20		アダプタ	個々のシステム、設備等に接続するための処理を行う。
	仲介	21	規格間相互変換	異なる標準規格間のデータモデル、インタフェース、プロトコルの相互変換を行う。
		22	ノードマッピング	ノードと個々のシステムにおけるサブシステム、プログラム、設備、等との関係付けを行う。
		23	プロファイル管理	ノードに関係付ける個々のシステムにおけるサブシステム、プログラム、設備、等のプロファイル情報を管理する。
		24	セキュリティ	他システムからの接続に対してセキュリティを保証する。

## 4.6. 実現方式

### 4.6.1. 動的グループ形成・制御

4.5にて述べた実現すべき機能のうち、本節では3.2にて方法を述べた動的グループ形式・制御の実現方式について述べる。

#### 4.6.1.1. 方式概要

3.1にて述べた本研究にて提案する異種システム連携コンセプトを構成する方法の1つである、3.2にて述べた動的グループ形成・制御を実現するために必要な機能の構成を図4-2に示し、各機能の概要を表4-4にまとめる。これらは図4-1及び表4-3にて示した該当機能について詳細化したものである。グループの構成要素であるノード、モニタ、コンダクタ、メディエータのオブジェクトを生成し、これらの構成要素及びグループの管理及び制御を行うグループ管理と、ノードとリソース（個々のシステムにおけるサブシステム、プログラム、設備、等）との関係付けを行うノードマッピングとで構成される。詳細は以降にて述べる。

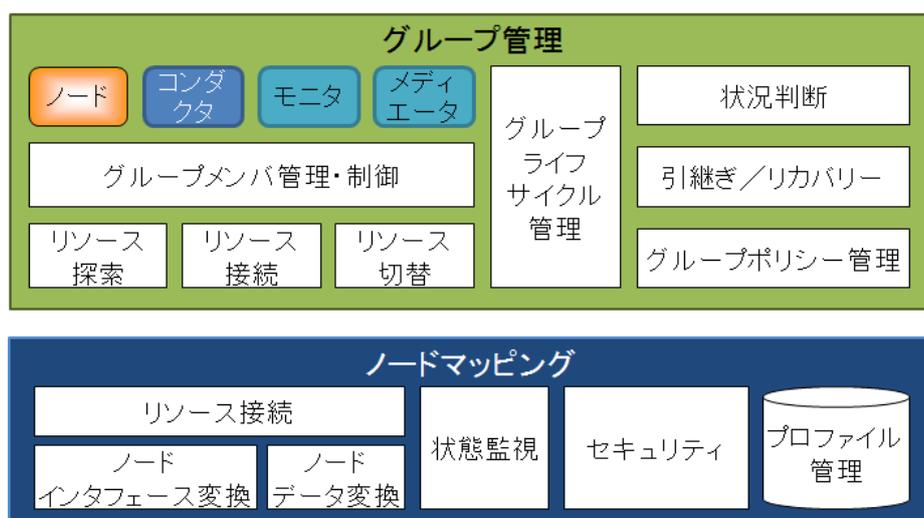


図 4-2 動的グループ形成・制御の機能構成

表 4-4 動的グループ形成・制御の機能一覧

大分類	小分類	説明	
1	グループ管理	ノード	グループにおける「ノード」のオブジェクト、「グループメンバ管理・制御」により生成、管理・制御、削除される。他システムから呼出し可能とする。
2	グループ管理	モニタ	グループにおける「モニタ」のオブジェクト、「グループメンバ管理・制御」により生成、管理・制御、

大分類	小分類	説明	
		削除される。	
3	コンダクタ	グループにおける「コンダクタ」のオブジェクト、「グループメンバ管理・制御」により生成、管理・制御、削除される。	
4	メディエータ	グループにおける「メディエータ」のオブジェクト、「グループメンバ管理・制御」により生成、管理・制御、削除される。	
5	グループメンバ管理・制御	グループ形成及びグループのメンバとなる「ノード」「モニタ」「コンダクタ」「メディエータ」の生成、管理・制御及び削除を行う。	
6	リソース探索	ノードに対してマッピングするリソース（個々のシステムにおけるサブシステム、プログラム、設備、等）を条件に従って探索する。	
7	リソース接続	ノードに対してマッピングするリソースと接続し、双方向の接続関係を維持する。	
8	リソース切替	ノードに対してマッピングするリソースを、条件に従って他のリソースへと切替を行う。	
9	グループライフサイクル管理	作成したグループの状態及び状態遷移に応じて必要なグループの制御を行う。	
10	状況判断	作成したグループの状態及び状態遷移と、システムの状態を監視する。	
11	引継ぎ・リカバリ	グループ及びシステムの状態に応じたグループ間の実行状態の引継ぎ、グループメンバの再構成、等を行う。	
12	グループポリシー管理	グループの作成及び制御に関するポリシーを管理する。	
13	ノードマッピング	リソース接続	ノードに対してマッピングするリソース（個々のシステムにおけるサブシステム、プログラム、設備、等）と接続する。
14		ノードインタフ	ノードの提供インタフェースに対してリソース側

大分類	小分類	説明
15	エース変換	のインタフェースとの対応付け、相互変換を行う。
	ノードデータ変換	ノードにて扱うデータに対してリソース側のデータとの対応付け、相互変換を行う。
	状態監視	ノードにマッピングするアプリケーションプログラムもしくは設備の稼動状態、等の監視を行う。
	プロファイル管理	ノードに関係付ける個々のシステムにおけるサブシステム、プログラム、設備、等のプロファイル情報を管理する。
	セキュリティ	他システムからの接続に対してセキュリティを保証する。

#### 4.6.1.2. グループ構築

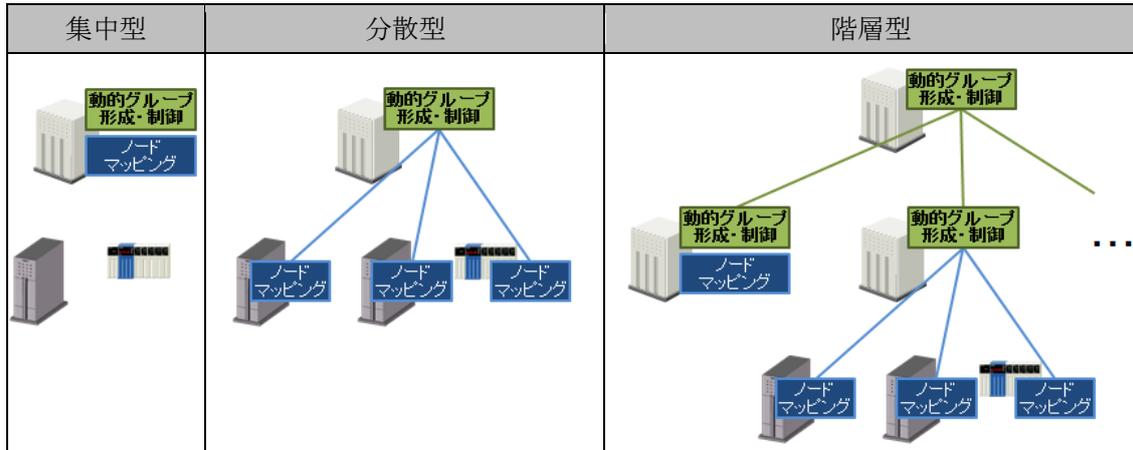
グループを構築するためのシステム構成パターン、グループの構成要素であるノード・モニタ・コンダクタ・メディアータの仕様、ノードと実際のシステム内のプログラム等のリソースとのマッピング方法について述べる。

##### (1) グループ構築のためのシステム構成パターン

グループ構築のため、図 4-2 にて示した動的グループ形成・制御の機能構成を実現するためのシステム構成のパターンを表 4-5 にまとめる。集中型、分散型、階層型の3パターン挙げられる。

集中型は、マッピングされるシステムの外部接続用の GW サーバ等にグループ管理及びノードマッピングの機能を配置する。システム内部の処理を外部には直接公開しない場合、システム内部の既存構成に対して改修できない場合等に本パターンを用いる。一方、分散型は、マッピングされるシステムの外部接続用の GW サーバ等にグループ管理の機能を配置し、システム内の処理を行う各サーバにノードマッピングの機能を配置する。システム内部の処理を外部に公開可能な場合、処理性能が求められる場合等に本パターンを用いる。階層型は、上述の集中型、分散型の組合せであり、グループ構築の対象とするシステムの範囲を段階的に拡張していくために用いる。

表 4-5 グループ構築のためのシステム構成パターン



(2) ノードの仕様

ノードは実際の連携先システムにおけるリソース（個々のシステムにおけるサブシステム、プログラム、設備、等）を抽象化したオブジェクトであり、各々のシステムの上位に位置付けられる。ここでリソースを抽象化した標準のオブジェクトモデルは多数存在するが、分野やシステム階層に固有の仕様であるものが多い。ゆえに本研究では、分野非依存の汎用的なモデルとして、超分散オブジェクト（Super Distributed Objects : SDO）モデル<sup>30)31)32)33)44)</sup>を用いてノードを表現する。モニタ、コンダクタ、メディエータについてもインタフェースを統一するために、同様に超分散オブジェクトモデルに準拠するものとする。

超分散オブジェクト（SDO）モデルを図 4-3 に示す。SDO は機器やアプリケーションの自律単位であり、共通的なインタフェース（SDO Interface、Configuration Interface、Monitoring Interface、Organization Interface）の API リストと自身の情報を記述したプロファイルとして、マッピングされる機器の属性に関する情報、提供可能なサービスや機能に関する情報等（Device Profile、Service Profile、Configuration Profile）を有する。さらに 1 つ以上の SDO 間の依存関係、階層関係等を表現する Organization を持つ。

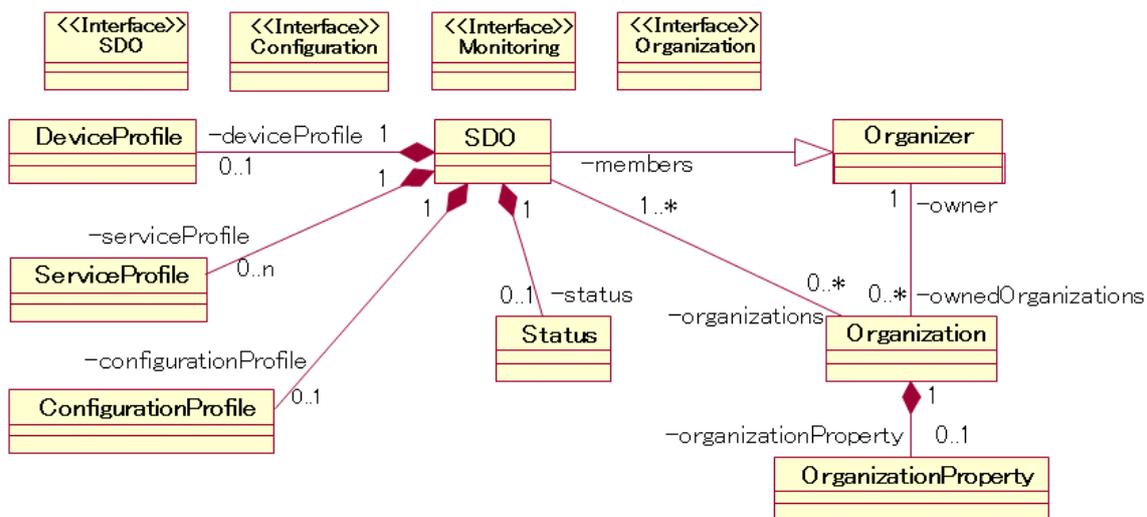


図 4-3 超分散オブジェクト (SDO) モデル (参考文献 33)より引用

超分散オブジェクト (SDO) モデルを用いてのノードの表現を表 4-6 及び図 4-4 に示す。表 4-6 には、モデルに準拠してノードが持つべきインタフェース、プロファイル、ステータスの各々についての内訳をまとめる。なお個々のプロファイル (DeviceProfile, ServiceProfile, ConfigurationProfile) の有無、数はノードのマッピング先のリソースの種別、仕様により異なる。

表 4-6 超分散オブジェクトモデルによるノードの表現

分類	項目	概要	主な内訳
インタフェース	SDO Interface	ノード (SDO) を管理するために使用する基本インタフェース。	<ul style="list-style-type: none"> <li>• Configuration Interface 取得</li> <li>• Monitoring Interface 取得</li> <li>• Organization Interface 取得</li> <li>• ServiceProfile にて公開する提供サービス呼出し</li> <li>• プロファイル情報</li> <li>• ステータス取得</li> </ul>
	Configuration Interface	ノード (SDO) に対して SDO 仕様にて定義したデータに関して追加、削除を実施するためのインタフェース。 DeviceProfile、ServiceProfile、	<ul style="list-style-type: none"> <li>• DeviceProfile 取得・追加・削除</li> <li>• ServiceProfile 取得・追加・削除</li> <li>• ConfigurationProfile 取得・追加・削除</li> <li>• Organization 取得・追加・削除</li> </ul>

分類	項目	概要	主な内訳
		ConfigurationProfile、Organization の変更を行うために仕様。	
	Monitoring Interface	ノード (SD0) にて定義されるプロパティ (属性値等) を監視するためのインタフェース。	<ul style="list-style-type: none"> <li>• ポーリング型監視</li> <li>• 予約型監視 (コールバック)</li> </ul>
	Organization Interface	グループ (Organization) を管理するために使用するインタフェース。	<ul style="list-style-type: none"> <li>• Organization のプロパティ取得・追加・削除</li> <li>• 依存関係取得・設定</li> <li>• メンバ取得・追加・削除</li> <li>• 所有者取得・設定</li> </ul>
プロファイル	DeviceProfile	ノード (SD0) にマッピングされる機器の属性に関する情報。	<ul style="list-style-type: none"> <li>• 機器のタイプ</li> <li>• 機器の製造者の名前</li> <li>• 機器のモデル名</li> <li>• 機器のバージョン番号</li> <li>• その他の機器固有のプロパティ情報</li> </ul>
	ServiceProfile	ノード (SD0) が提供可能なサービスや機能に関する情報。	<ul style="list-style-type: none"> <li>• サービス・機能の識別情報</li> <li>• サービス・機能提供のためのインタフェース情報</li> <li>• サービス・機能に関するプロパティ情報</li> <li>• サービス・機能へのリファレンス情報</li> </ul>
	ConfigurationProfile	ノード (SD0) を設定するためのパラメタ値等の組み合わせに関する情報。	<ul style="list-style-type: none"> <li>• 設定するパラメタのリスト</li> <li>• 設定パラメタ値組合せのリスト</li> </ul>
ステータス	Status	ノード (SD0) の現在の状態を表す情報。	<ul style="list-style-type: none"> <li>• ステータス情報のリスト</li> </ul>

図 4-4 にはこれらのインタフェースを介してのコンダクタ、モニタとノードとの間の呼び出し関係を示す。コンダクタは表 4-6 に挙げた Organization Interface を用いてノード

のグループを形成、制御する。グループは SDO の集合である Organization として表現する。またコンダクタは表 4-6 に挙げた SDO Interface を用いて各ノードの実行指示、制御を行う。モニタは表 4-6 に挙げた Monitoring Interface を用いてノードの状態監視を行う。

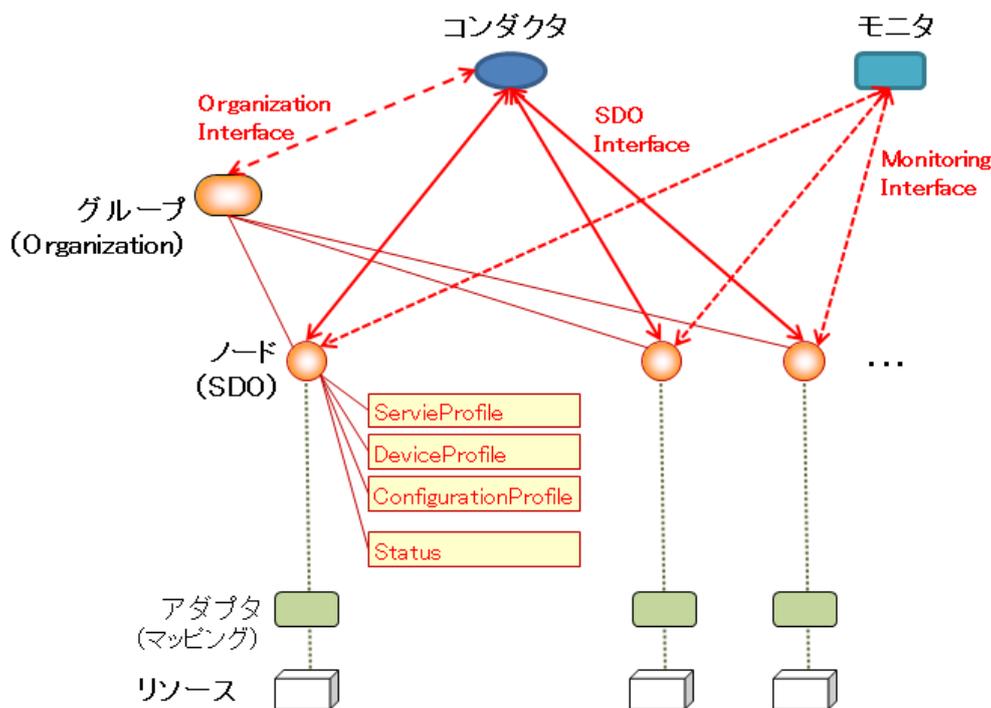


図 4-4 超分散オブジェクトモデルによるノードの表現

### (3) ノードマッピング

前項にて述べたノードと、実際のシステムにおけるリソース（個々のシステムにおけるサブシステム、プログラム、設備、等）とのマッピングについて図 4-5 に示す。

超分散オブジェクト (SDO) モデルに準拠するノードと個々のシステムにおける仕様に従ったリソースとの間に、データやインタフェースを対応付けるためのアダプタを配置する。前項にて述べた超分散オブジェクトモデルによる標準インタフェースである Monitoring Interface (監視用) 及び Configuration Interface (パラメタの設定及び参照用) に対しては、アダプタにパラメタの名称と値の組合せによるパラメタキャッシュをおく。ここにリソース固有のインタフェースを用いてリソースから取得した値をキャッシングしておき、標準インタフェースを介して外部から要求を受けた場合にデータを提供する。

リソース固有の機能・処理またはデータのうち外部に公開可能なものは、アダプタにて固有インタフェースを備える。これらの固有インタフェースに対しては、前項にて述べた超分散オブジェクトモデルにおける ServiceProfile にリファレンス情報を記載し公開することで、SDO Interface を介して外部から呼び出し可能とする。

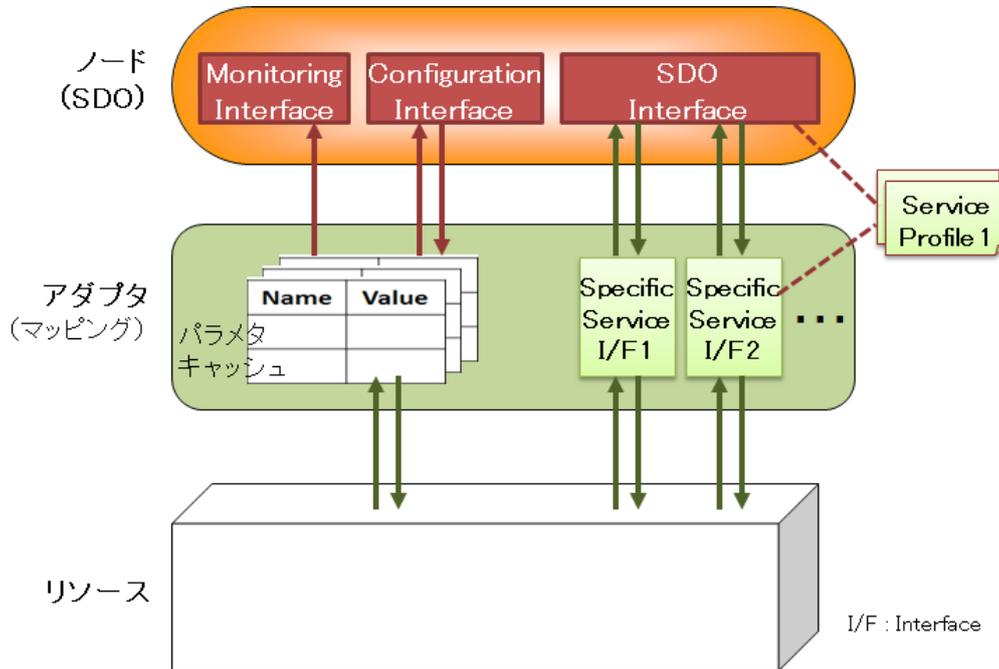


図 4-5 ノードとリソースとのマッピング

#### (4) グループ動作

前項までに述べた超分散オブジェクトモデルを活用したノードの仕様を用いて実現する、3.2.2.1にて述べたグループの動作原理に従ったグループの動作について述べる。グループ動作のシーケンスを図 4-6 に示す。図 4-6 に示すように、モニタ及びコンダクタが 3.2.2.2にて述べたサービスシナリオを読み込む。コンダクタは上記サービスシナリオに基づきサービス実行に必要なノードを SDO Interface を用いて検索、選出し、Organization Interface を用いてグループ化 (Organization 形成) を行う。モニタに対してこれらのグループメンバとなるノードを通知する。以後、モニタは Monitoring Interface を用いてこれらのノードの監視を開始する。ここでは定期的に該当するノードの属性値を取得するポーリング型の監視もしくは該当するノードの属性値に変化が発生した際に通知を受ける予約型の監視のいずれかを行う。コンダクタはサービスシナリオに従って、SDO Interface を用いてノードを起動・初期化し、順次ノードに対して処理実行を指示し、サービスを実行していく。

モニタが Monitoring Interface を介してノードの異常・実行不能を検出すると、代替となるメンバの選出を行い、グループ再構成をコンダクタに指示する。コンダクタはモニタの指示を受けて、Organization Interface を用いて異常ノードの切り離し及びモニタの選出した代替ノードの組込みによりグループの再構成を実施し、サービス実行を再開する。

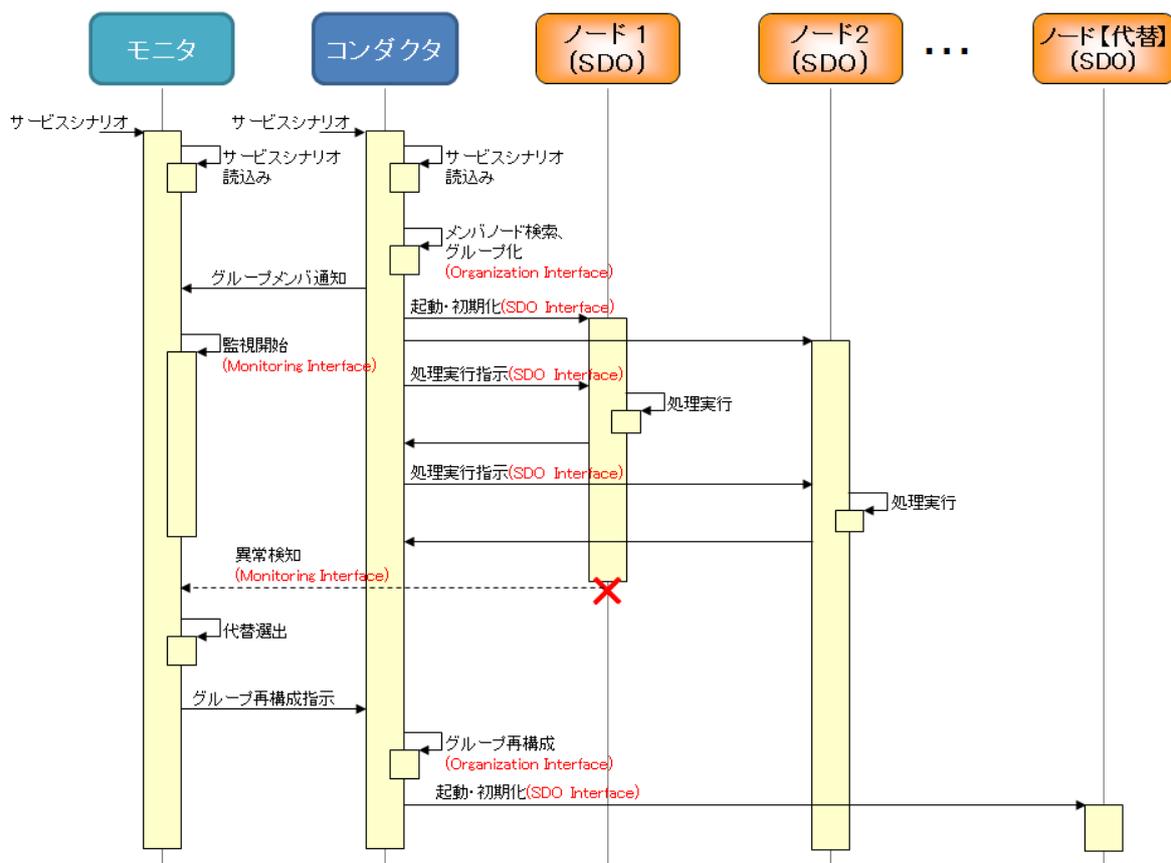


図 4-6 グループ動作のシーケンス

#### 4.6.1.3. グループメンバ管理

4.6.1.2(4)にて述べたグループ動作のシーケンスにて、モニタによる代替ノード選出におけるグループメンバ代替の方針を表 4-7 にまとめる。ここでグループメンバの代替として、「拡張」、「完全代替」、「部分代替」、「縮退」の4パターンを考える。「拡張」は、サービスの質向上もしくは対象範囲拡大のために、ノードを追加することを指す。「完全代替」は、対象ノードと同一のデータ、機能を提供し、同一の動作をするノードで代替することを指す。「部分代替」は、対象ノードの一部のデータ、機能を提供するノードもしくは類似する動作をするノードで代替することを指す。「縮退」は、代替は行わず、対象ノードは停止させて、残ったノードのみでサービスを継続する。なお「部分代替」、「縮退」はアプリケーション要件が合致する場合のみ実施可能である。また代替実施に対する要件として、サービス無停止での実施、もしくは中断可能での実施のいずれかがある。

表 4-7 グループメンバ代替の方針

#	再構成時のサービス実行要件	グループメンバ代替の実施方針			
		拡張	完全代替	部分代替	縮退
1	無停止（即時切	—	代替ノードの事前	キャッシュ、履歴	代替なし。該当ノ

#	再構成時のサービス実行要件	グループメンバ代替の実施方針			
		拡張	完全代替	部分代替	縮退
	替)		確保、即時切替。 制御周期を保証する範囲内で実施 (多重系構成。性能要件によりホットスタンバイ)。	等を用いての該当ノードに対するデータ提供、機能実行結果の提供(最新値ではない)。	ードのデータ、機能を閉塞して処理続行。
2	中断可能	新規ノードを追加、グループ再構成。	該当ノードの探索、切替(該当ノードが存在しない場合は部分代替へ)。	完全代替可能なものが無い場合、必要なデータの一部を有するノードの探索、1つ以上の組合せによる代替(必要なデータ全ての提供は保証できない)。	—

表 4-7 のグループメンバ代替の方針のうちの「部分代替」、「縮退」に関して、適用可能か否かを判断するための指標を表 4-8、表 4-9 にまとめる。これらの指標での変化を考慮して、提供可能なサービスに対してグループメンバ代替を適用する。

表 4-8 部分代替の指標

#	指標	概要	部分代替による変化
1	応答性能	要求に対する応答時間。計算機性能、要求先との物理的距離、ネットワーク性能等に依存する。	サービス実行の時間遅延が発生する。
2	データ頻度	データ取得可能な頻度。代わりにキャッシュや履歴値を使用する場合もある。	サービス実行において利用可能なデータの更新頻度が低下し、サービスレベルが低下する。
3	データの新鮮さ	取得可能なデータが生データか蓄積された履歴データか、または加工後のデータか。	サービス実行において利用可能なデータがデータソースの出力するデータでなくなる。

表 4-9 縮退の指標

#	指標	概要	縮退による変化
1	制御範囲	制御可能な機器・システムの範囲。	サービス実行において、利用可能な機器、システムの範囲が縮小する。
2	アクセスデータ数	アクセス可能なデータ数。	サービス実行において利用可能なデータが減少し、サービスレベルが低下する。

1.1にて、社会インフラを支えるシステムの要件として、従来からの安定・安心だけでなく、以下の3つを挙げている。(1)様々な需要家やサービス事業者のインフラ供給への参入、新サービスの提供、供給と需要の関係の変動等の様々な変化に柔軟に対応できる多種多様性、(2)従来からの既存設備だけでなく、様々な需要家やサービス事業者の大量の機器やアプリケーションが様々なネットワークを通じて社会インフラに加わった場合、災害発生等が原因でシステムの一部が動作不能となった場合でも安定的にインフラの稼働とサービス提供を実現する信頼性、(3)社会インフラの成長や長期維持に対応した拡張性、である。

上記の要件(1)に対しては、表 4-7 のグループメンバ代替の方針の、サービス中断可能での拡張または完全代替が対応する。要件(2)に対しては、表 4-7 の、サービス無停止による部分代替または縮退、サービス中断可能での部分代替が対応する。要件(3)に対しては、表 4-7 の、サービス中断可能での拡張が対応する。さらに従来からの安定・安心の要件を満たすためには、表 4-7 の、サービス無停止での完全代替が対応する。

上記を踏まえて、本節で述べたグループメンバ管理の電力・スマートグリッド分野におけるユースケース例を表 4-10 にまとめる。なお本研究にて提案する技術を電力・スマートグリッド分野に適用する場合の想定ユースケースの詳細は5.1にて述べる。

表 4-10 電力・スマートグリッド分野におけるユースケース例

#	再構成時のサービス実行要件	グループメンバ代替の実施方針			
		拡張	完全代替	部分代替	縮退
1	無停止 (即時切替)	—	<ul style="list-style-type: none"> <li>• 系統安定化</li> </ul>	<ul style="list-style-type: none"> <li>• Demand Response (DR)</li> <li>• Virtual Power Plant (VPP)</li> </ul>	<ul style="list-style-type: none"> <li>• DR, VPP</li> <li>• 検診、見える化</li> </ul>

#	再構成時のサービス実行要件	グループメンバ代替の実施方針			
		拡張	完全代替	部分代替	縮退
2	中断可能	<ul style="list-style-type: none"> <li>DR（需要家追加）</li> </ul>	<ul style="list-style-type: none"> <li>Enterprise Asset Management (EAM)</li> </ul>	<ul style="list-style-type: none"> <li>DR, VPP</li> <li>系統復旧（災害時）</li> </ul>	—

#### 4.6.1.4. 非機能要件による限界

動的グループ形成・制御を実施する上での非機能要件に関して、4.6.1.2にて述べたグループ構築では、グループのメンバとなるノードの数が増える程、またメンバのノードのマッピング先のインフラシステムの数が増えるほど、グループに含まれる複数のノードを用いてサービスを実行する際に、サービス実行に必要なノードの検索・選出や、複数のノード間の連携に時間を要して、サービスの処理の性能が低下し得る。

上記のような非機能要件に関するトレードオフは、実際の適用先システム毎の状況に応じて調整する必要がある。

#### 4.6.2. 異種ミドルウェア連携

4.5にて述べた実現すべき機能のうち、本節では3.3にて方法を述べた異種ミドルウェア連携の実現方式について述べる。

##### 4.6.2.1. 方式概要

3.1にて述べた本研究にて提案する異種システム連携コンセプトを構成する方法の1つである、3.3にて述べた異種ミドルウェア連携を実現するために必要な機能の構成を図4-7に示し、各機能の概要を表4-11にまとめる。これらは図4-1及び表4-3にて示した該当機能について詳細化したものである。詳細は以降にて述べる。

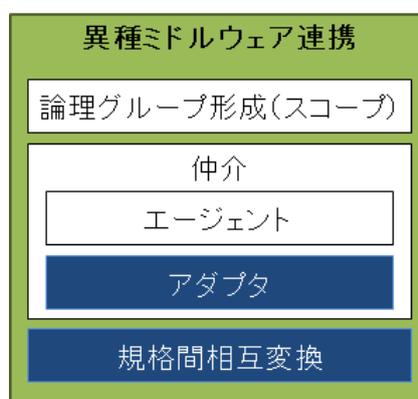


図 4-7 異種ミドルウェア連携の機能構成

表 4-11 異種ミドルウェア連携の機能一覧

大分類	小分類	説明
1 規格間相互変換		異なる標準規格間のデータモデル、インタフェース、プロトコルの相互変換を行う。
2 仲介	エージェント	個々のシステムへのアクセス代行、監視、システム間の中継を行う。
3	アダプタ	個々のシステム、設備等に接続するための処理を行う。
4 論理グループ形成 (スコープ)		物理構成に依存しない計算機ノードの論理的なグループの形成及び管理を行う。

#### 4.6.2.2 規格間相互接続

3.3.3 にて述べた規格間相互変換は、“データモデル変換”、“インタフェース変換”、“プロトコル変換”の3層構造にて実施する。これらの処理の流れを図 4-8 に示す。各処理の詳細を以降にて述べる。

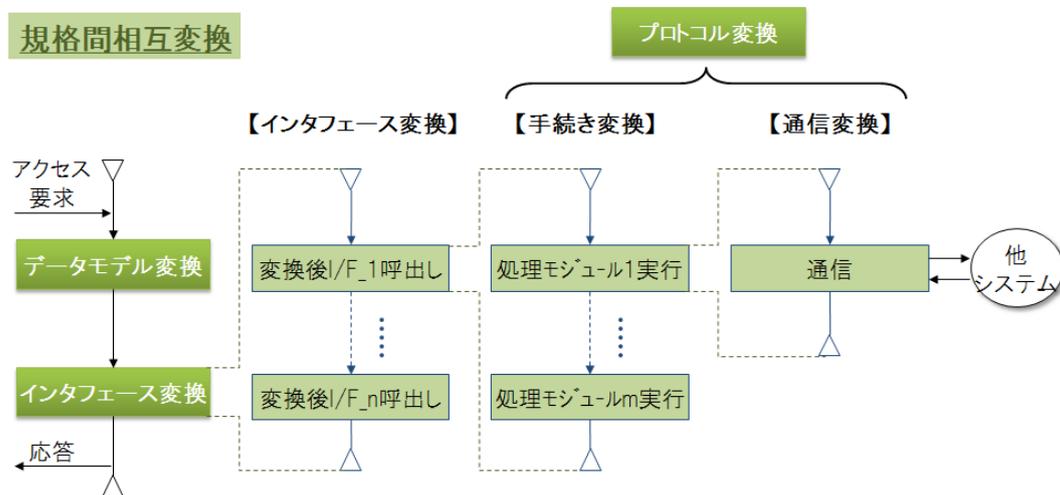


図 4-8 規格間相互変換の処理の流れ

##### (1) データモデル変換

異なる標準規格のデータモデル間の相互変換として、各々のデータモデルにおけるデータクラス・属性間のマッピング及びデータの値、単位等の変換を実施する。従来は相互変換する標準規格の組合せ毎にマッピング及び変換の処理を個別にアプリケーションにて作り込んでいた。また個々の標準規格におけるデータモデルのデータクラス仕様が膨大であ

る上に、実際にデータクラス・属性のどの部分をどのように使用するかは適用先システムやアプリケーションに大きく依存する。

そこで本研究にて提案する規格間相互変換のデータモデル変換では、図 4-9 に示すように、入力メッセージデータの解析及びデータ抽出の処理と、指定されたデータモデルへのマッピング及びデータ値変換の処理を実施する。ユーザが作成するマッピング及び変換に関する定義の一覧として表 4-12 にまとめる。これらは本データモデル変換処理にて使用する。またマッピング定義の例を図 4-10 に示す。

データモデル変換では図 4-9 に示すように、最初に入力のメッセージデータに対してデータ解析・抽出処理を実施する。本処理では、ユーザ作成の解析処理関数を呼び出す。本処理関数では、ユーザが各々作成する定義（表 4-12 の#1、 2）を参照し、入力メッセージから抽出した個々のデータ項目を共通データ形式（“キー”、“値”、“単位”）に格納して出力する。ここで入力メッセージデータ構造は準拠する標準規格により異なり、データ構造に対してどのようにデータを格納するかはアプリケーションにより異なり得る。このため表 4-12 に挙げたメッセージデータ解析用定義とユーザ作成解析関数を使用可能とすることで、個別対応が可能としている。

データ抽出処理が出力した共通形式による個々のデータ項目に対して、データマッピング及び変換処理を実施する。本処理では、表 4-12 に挙げたユーザ作成のデータマッピング定義を参照して、変換先のデータ項目にマッピングする。データ値の変換やデータ構造の変換等が必要なデータ項目に対しては、ユーザ作成の変換関数を呼び出し、個別の変換処理を実施する。なお入力メッセージのデータ内容の参照及び抽出が不可であるものは、表 4-12 に挙げたユーザ作成の非解析メッセージデータマッピング定義を参照し、データ構造及び値は無変換のままでマッピングのみを実施する。これらの処理も表 4-12 に挙げたユーザ作成の定義とユーザ作成の変換処理関数を使用可能とすることで個別対応が可能としている。

上記のデータマッピング及び変換処理を実施後のデータ全てをリストとして出力する。

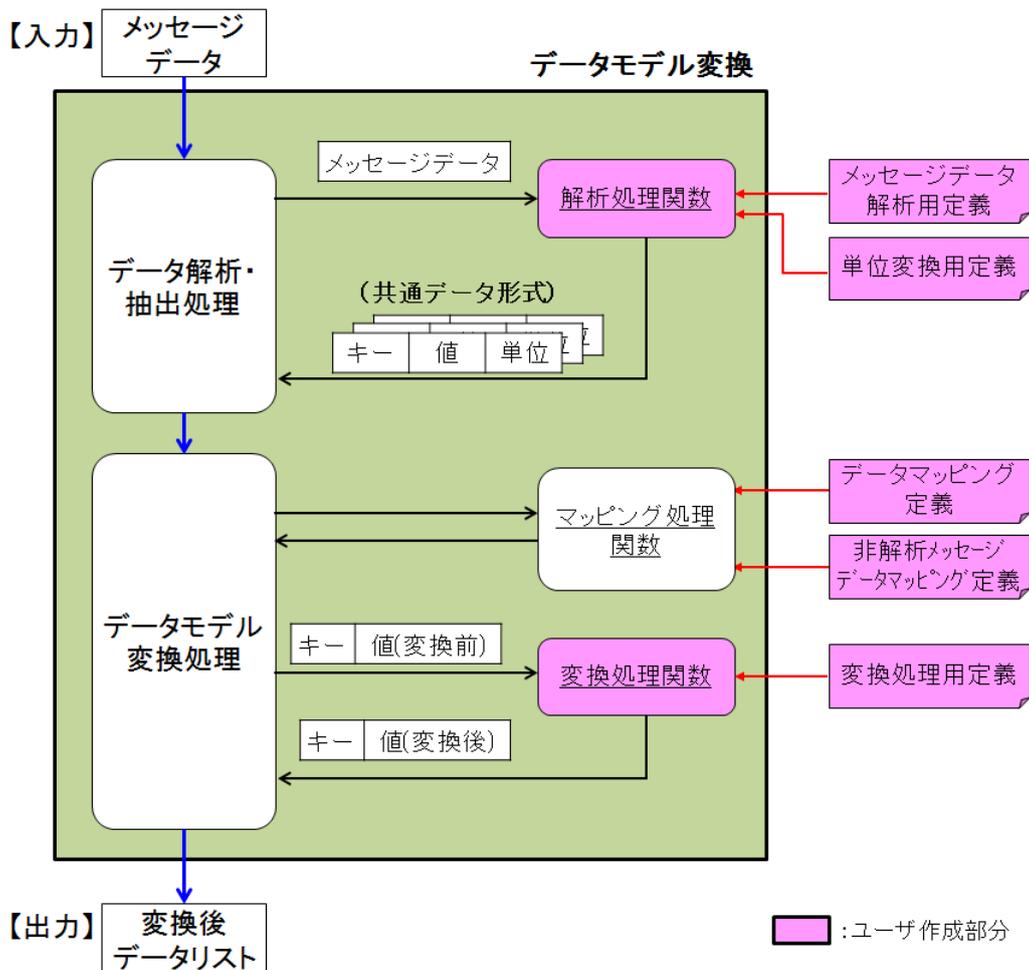


図 4-9 データモデル変換

表 4-12 マッピング及び変換の定義一覧

#	定義	内容
1	メッセージデータ解析用定義	入力メッセージのデータ構造等に関する定義。ユーザが作成する解析処理用関数にて参照する。
2	単位変換用定義	単位の対応付けに関する定義。ユーザが作成する解析処理用関数にて参照する。
3	データマッピング定義	入力データに含まれるデータ項目を変換後のデータ構造でのデータ項目へとマッピングするための定義。
4	非解析メッセージデータマッピング定義	変換不可能な入力データ（非解析メッセージ）を変換後のデータ構造でのデータ項目へとマッピングするための定義。

#	定義	内容
5	変換処理用定義	データ値の変換に必要なパラメタ情報等の定義。ユーザが作成する変換処理用関数にて参照する。

```
#(テーブル名), (変換先データ名) = (変換元データ名)
Table_A, Target1           = SourceKey3
Table_B, Target2           = SourceKey4
```

図 4-10 マッピング定義例

### (2) インタフェース変換

図 4-8 に示すようにインタフェース変換では、自システムのアクセスインタフェースに相当すると定義される他システムのアクセスインタフェースを呼び出す。上記の他システムのアクセスインタフェース呼出しに際して、次項にて述べるプロトコル変換の処理を実行する。

### (3) プロトコル変換

システム間でやり取りするためのプロトコルは、実施すべき処理の方法、順序、等である手続きと、システム間のメッセージデータの伝送である通信とで構成される。よって図 4-11 に示すようにプロトコル変換は、手続き変換及び通信変換から成る。

手続き変換は、アクセスインタフェース使用に関連する一連の処理やシーケンス形態を選択し実行する。通信変換は、アクセスインタフェース呼出しの際に発生する通信に関して、呼び出し元（連携元）システム、呼出し先（連携先）システム側で使用する通信プロトコルを再現する。これらの処理の内訳は表 4-13 にまとめる。

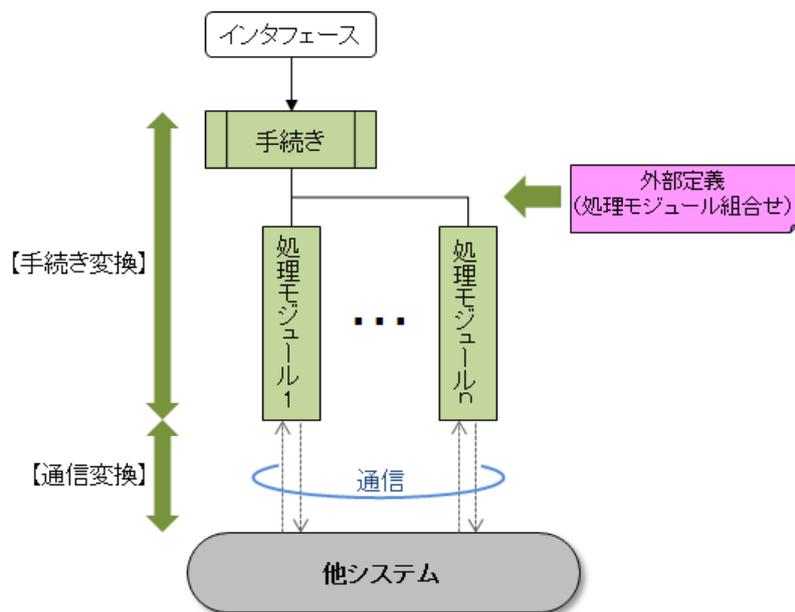


図 4-11 プロトコル変換

表 4-13 プロトコル変換の処理の内訳

分類	処理内容	補足説明
手続き変換	1 アクセス形態選択	<ul style="list-style-type: none"> <li>例えば連携先の準拠規格が IEC61970<sup>51)55)56)</sup>である場合、上記規格にて規定する Generic Data Access (GDA)、High Speed Data Access (HSDA)、等のアクセス形態を選択。</li> </ul>
	2 アクセス処理選択	<ul style="list-style-type: none"> <li>規格にて規定される処理内容、またはインタフェース呼出しに伴う前処理、アクセス処理、後処理、等を選択。</li> <li>上記の処理はモジュール化しておき、手続きは1つ以上の処理モジュールの組合せで構成できる。</li> <li>IEC61970、IEC61968<sup>51)57)58)</sup>、IEC61850<sup>52)53)54)</sup>の各仕様を検証した結果として、手続きを構成する処理(モジュール化対象)の種別の例を以下に挙げる。 <ul style="list-style-type: none"> <li>- 前処理：セッション管理、ブラウズ、グループ管理</li> <li>- データ保守処理：データ管理、スキーマ管理</li> <li>- アクセス処理：データアクセス、イベント、コールバック、データ再現</li> <li>- 後処理：データ保存</li> </ul> </li> </ul>
	3 シーケンス形態選択	<ul style="list-style-type: none"> <li>インタフェース、アクセス処理に依存するシーケンス形態を選択。以下の4パターンが挙げられる。 <ul style="list-style-type: none"> <li>- Request/Reply 型</li> </ul> </li> </ul>

分類	処理内容	補足説明
		<ul style="list-style-type: none"> <li>- Publish/Subscribe 型</li> <li>- 一方向型</li> <li>- 待受け型</li> </ul>
通信 変換	4 通信プロトコル選択	<ul style="list-style-type: none"> <li>• 連携先システムにて実装される通信プロトコルを選択。</li> <li>• #3 のシーケンス形態に対応付くものを選択。</li> </ul>
	5 ネーミングサービス	<ul style="list-style-type: none"> <li>• 送信先アドレスの解決。</li> </ul>
	6 メッセージ生成	<ul style="list-style-type: none"> <li>• 規格によるフォーマットのメッセージデータ生成。</li> </ul>

#### 4.6.2.3. 仲介(エージェント・アダプタ)

表 3-10 に挙げた動作仲介のために、エージェント及びアダプタを、各系の通信基盤上に導入する。本研究にて提案する異種ミドルウェア連携のアーキテクチャを図 4-12 に示す。

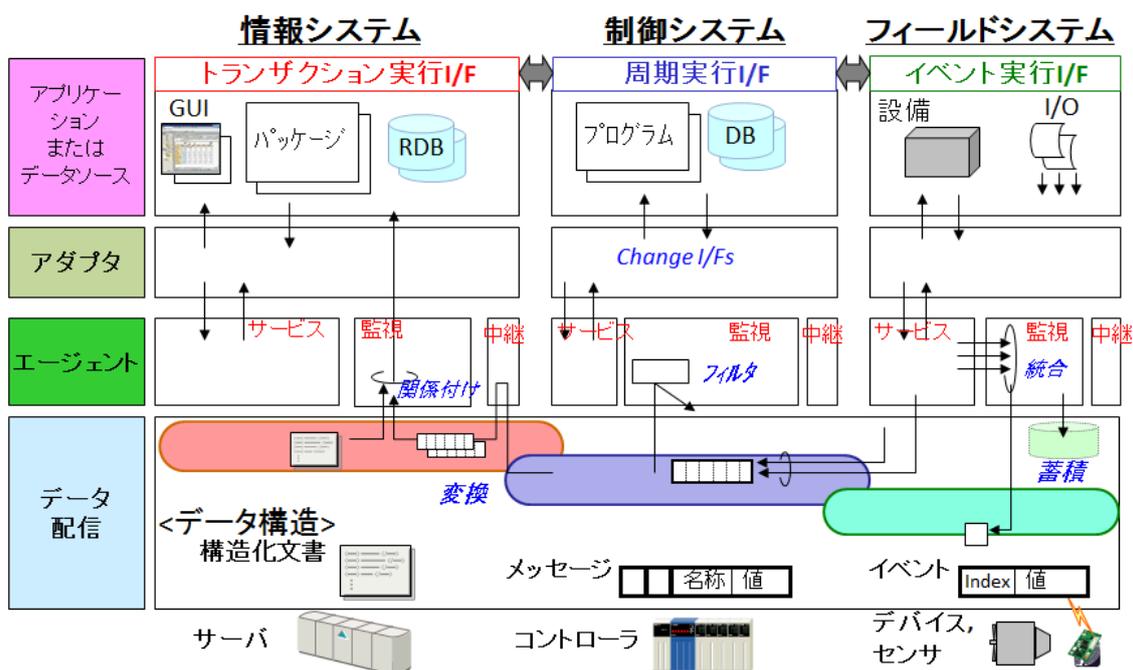


図 4-12 異種ミドルウェア連携のアーキテクチャ (筆者文献 61) より引用、一部加筆)

エージェントには、データソース毎に配置しアダプタを用いてデータアクセスを行うサービスエージェント (Service Agent)、系内を流れるデータの蓄積、フィルタリング、統合・分割等を行う監視エージェント (Monitoring Agent) 及び系間で通信の中継を行う中継エージェント (Translating Agent) の3種類ある。一方アダプタは、データソース毎に

配置し、インタフェースやプロトコルの変換、データソース毎の個別のデータから共通形式のデータへのマッピング、演算等を行う。

アダプタの構成及び概要を図 4-13 に示す。ライブラリ (Library) にて実装する、接続対象となる設備・機器の準拠する規格に応じたインタフェースやプロトコルを用いた設備・機器等との接続処理をアダプタから呼び出す。これらのライブラリを介して得られる設備・機器のデータをエージェントがアプリケーションに提供するための、データ定義や演算処理等を用いてのデータ変更・構造作成をアダプタにて行う。またデータソースのマスター・スレーブ判定を行い、判定結果に基づきデータ出力可否等のエージェントの動作を変化させる。上記からエージェントを介してアプリケーションには個々の設備・機器の準拠規格等の違いを意識させずに、システムで規定する共通インタフェースによるデータアクセスを可能とする。

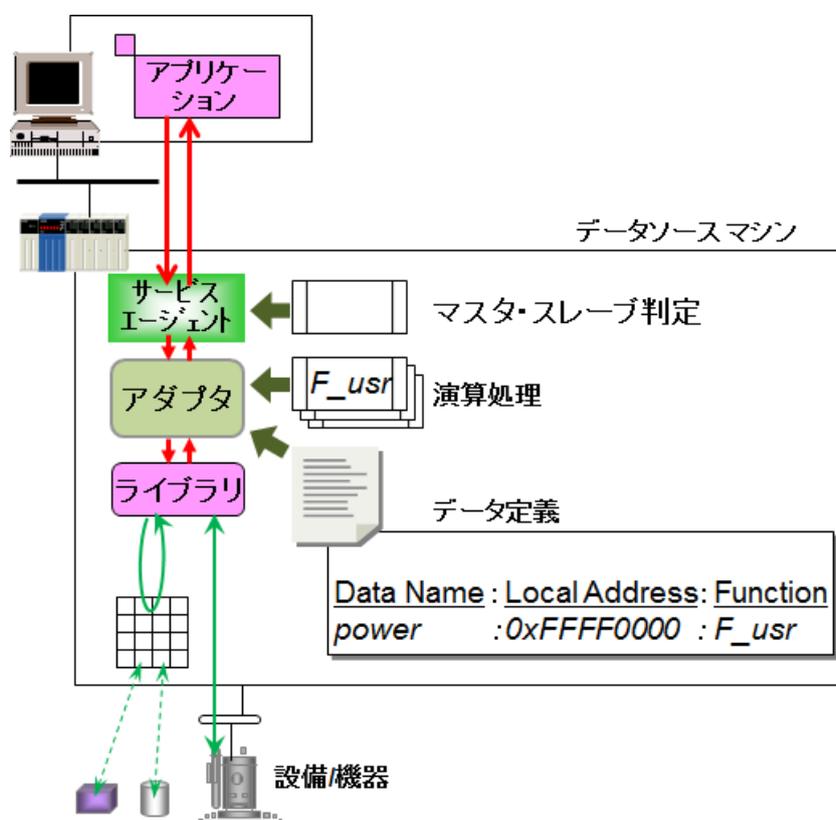


図 4-13 アダプタの構成及び概要 (筆者文献 61) より引用、一部加筆)

#### 4.6.2.4. 論理グループ形成(スコープ)

本研究にて提案する論理的な計算機ノードのグループであるスコープ (Scope) の概要を図 4-14 に示す。アプリケーションがシステム内のデータにアクセスする際に、アクセス範囲としてスコープを指定する。スコープは業務上の管理体系や用途等に応じて、実際のネットワーク構成を意識することなく設定可能とする。

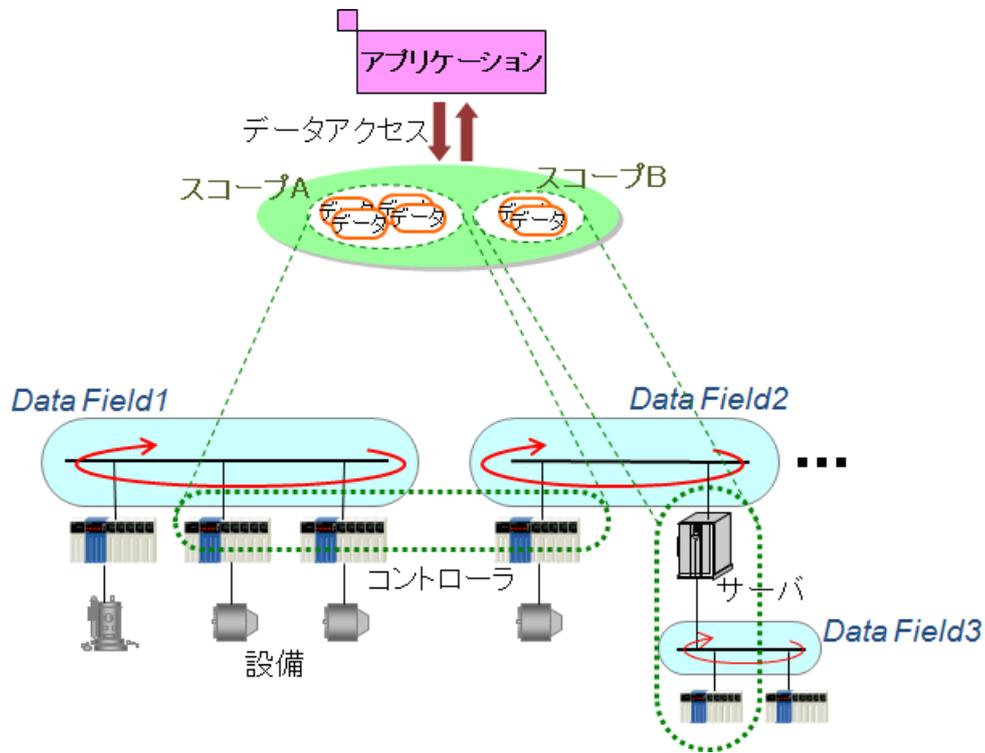


図 4-14 スコープの概要 (筆者文献 61)より引用、一部加筆)

スコープの制御方法を図 4-15 に示す。データアクセスの際にはマルチキャストによりメッセージが送信される。メッセージのヘッダにはアクセス先のスコープを示すコードを含む。

ゲートウェイ (GW) には通信の中継を行う中継エージェントを配置している。中継エージェントはスコープに対して所属するエージェント、関連するネットワークセグメントに関する情報を有するスコープ管理テーブルを保持し、メッセージのヘッダから抽出したスコープ情報と上記テーブルを照会して接続先のネットワークに該当するスコープに所属するエージェントが含まれる場合のみデータを転送する。上記のテーブルはサービスエージェントが新規に追加される度に更新される。

また各計算機ノードにて稼動するサービスエージェントは同様にメッセージのヘッダからスコープ情報を抽出・参照し、自らが所属するスコープである場合のみメッセージを受信する。

このようにゲートウェイ及び計算機ノードに搭載するエージェントによるフィルタリングによりスコープというデータアクセス範囲の制御を行っている。

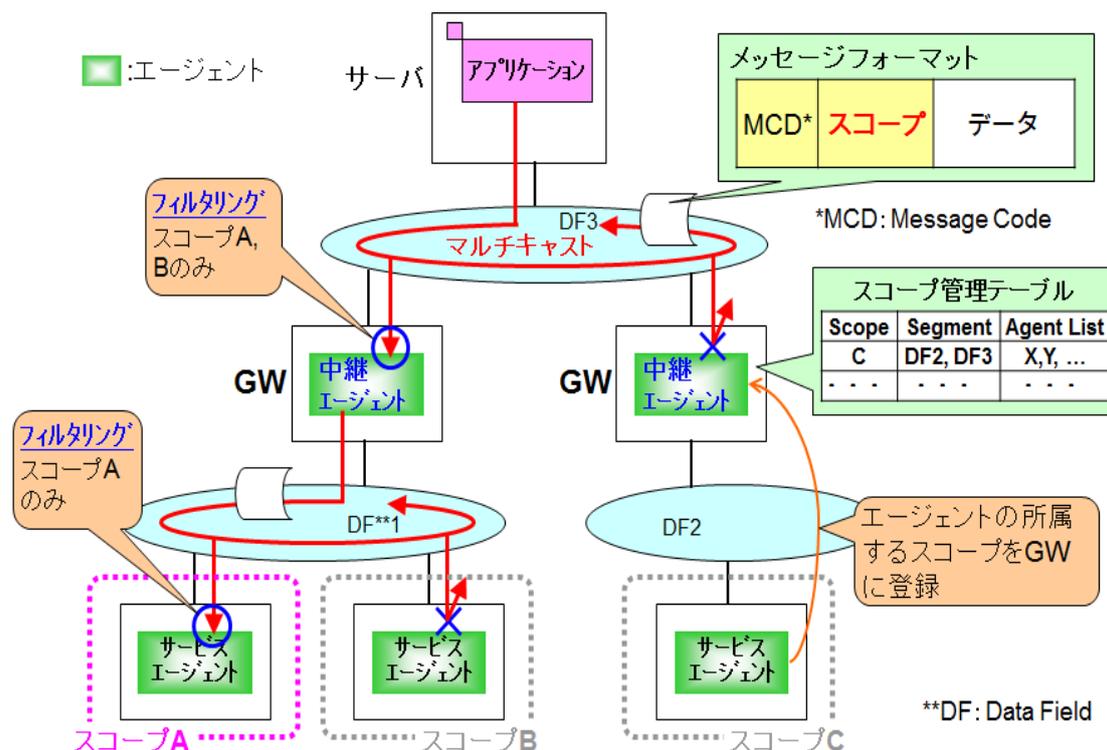


図 4-15 スコープの制御方法 (筆者文献 61) より引用、一部加筆)

#### 4.6.2.5. 非機能要件による限界

##### (1) 非機能要件に関するトレードオフ

異種ミドルウェア連携を実施する上での非機能要件に関して、4.6.2.2にて述べた規格間相互変換では、変換対象となるデータのサイズが大きくなる程、変換処理に要する時間が増加し性能が低下し得る。また4.6.2.3にて述べた仲介では、他系からのアクセス増加に伴う加負荷、他系からのアクセスに対する認証等によるセキュリティ保証に伴う加負荷、等により、自系の処理の性能が低下し得る。

上記のような非機能要件に関するトレードオフは、実際の適用先システム毎の状況に応じて調整する必要がある。

##### (2) 異種システム連携の境界条件 (非機能要件) の回避

3.6にて述べた異種システム連携の境界条件を充足できずに、異種システム間の連携が実施できなくなることを回避するためには、該当する指標を充足できるように、条件を緩和することや連携動作の内容を変更することが考えられる。ただしその場合、トレードオフとして他の指標において劣化する場合があります。表 3-13 にまとめた異種システム連携の境界条件 (非機能要件) に関して、充足できない場合の回避方法例と影響を受ける他の指標及び影響内容 (トレードオフ) について表 4-14 にまとめる。

上記から充足できていない境界条件を回避して異種システム連携を実現する上では、ト

レードオフとなる他の指標での影響内容も考慮して、異種システム連携におけるアプリケーションの要件を満たす範囲内であるか否かを確認する必要がある。

表 4-14 異種システム連携の境界条件（非機能要件）の回避方法（例）

#	分類	充足できない指標	回避方法（例）	影響受ける指標	影響内容（トレードオフ）
1	信頼性	機能充足	・類似する代替機能の提供	機能充足	・連携先が要求する機能が実行できない場合あり
2		トランザクション保証	・トランザクション保証可能な処理のみ実行	機能充足	・連携先が要求する機能が実行できない場合あり
3		可用性	・ダウンタイムの許容	運用期間	・運用不可の期間が発生
4	性能	応答性能保証	・GWに配置したデータキャッシュ用いて、連携先に応答	機能充足	・データ値が最新でない場合あり
5		リアルタイム性	・GWに配置したデータキャッシュ用いて、連携先に応答	機能充足	・データ値が最新でない場合あり
6	セキュリティ	正当性	・不特定多数に公開可能な範囲に限定してアクセス許容	アクセス範囲	・アクセス可能範囲が限定される
7		アクセス範囲	—	—	—
8		秘匿性	・不特定多数に公開可能なデータのみ連携先とやり取り	機能充足	・連携先が要求するデータを取得できない場合あり

## 4.7. システムの実現

### 4.7.1. システム連携パターン

複数の異なる主体及び異なるシステムが存在する社会インフラシステムにおいて、3.1にて述べた異種システム連携コンセプトの適用先となるシステム間の連携のパターンを表4-15にまとめる。パターン毎のシステム連携の組合せ例を図4-16に示す。

表 4-15 システム連携のパターン

#	連携パターン	説明
(1)	情報制御連携	1主体内における情報系と制御系の連携。

#	連携パターン	説明
(2)	同種連携	1 主体内における同種の系の連携。例えば、(2-1)情報系と情報系の連携、(2-2)制御系と制御系の連携。
(3)	既設連携、新旧並存	1 主体内における既設の系との連携。例えば、(3-1)情報系と既設情報系の連携、(3-2)制御系と既設制御系の連携、(3-3)情報系と既設制御系の連携。
(4)	異主体連携	異なる複数の主体の間での連携。例えば、(4-1)主体を跨いでの情報系と情報系の連携、(4-2)制御系と制御系の連携、(4-3)情報系と制御系の連携。
(5)	サービス事業者連携	サービス事業者と情報系及び制御系を有する主体との間の連携。例えば、(5-1)サービス系と情報系の連携、(5-2)サービス系と制御系の連携。

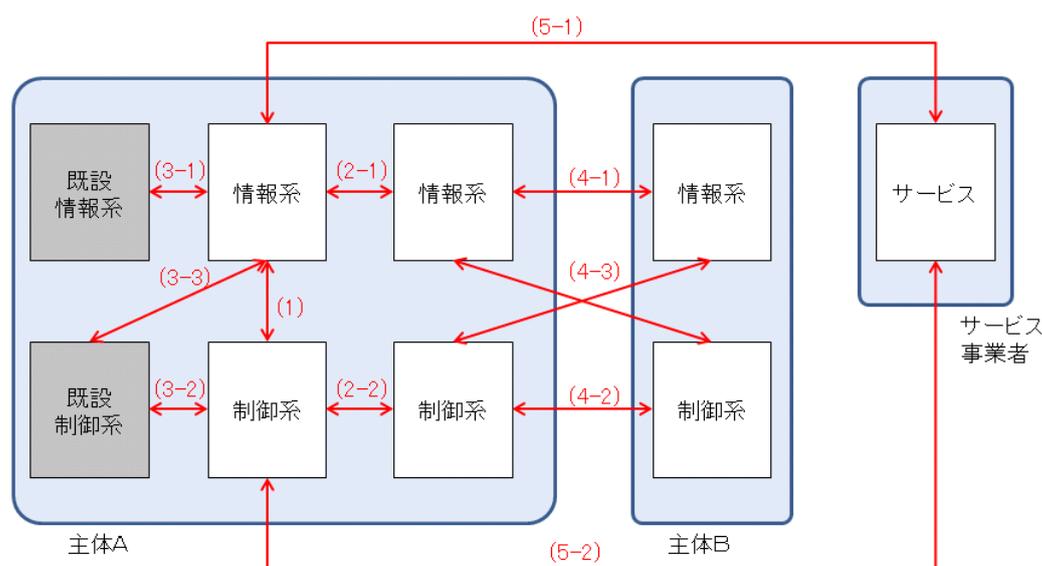


図 4-16 システム連携のパターン

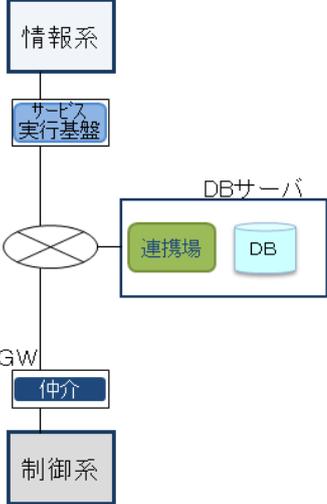
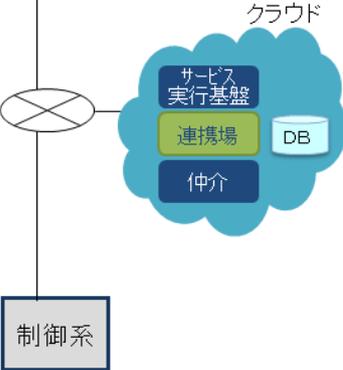
#### 4.7.2. システム構成パターン

4.7.1にて述べた5つのシステム連携パターンに対して、本研究による異種システム連携コンセプトを適用したシステムを実現するためのシステム構成パターンを表4-16にまとめる。ここでは4.7.1にて述べたシステム連携パターンに対して、4.3にて述べた想定するハードウェア構成パターン及び4.4にて述べた想定する機能配置を当てはめて、各パターンでのシステム構成の概要及び制約を挙げる。

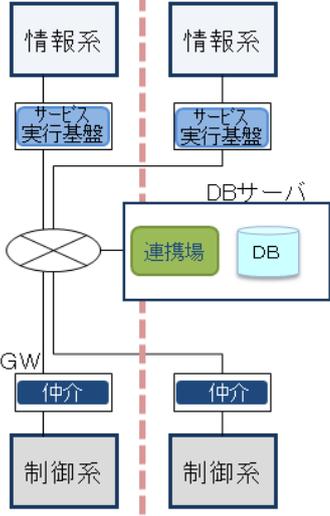
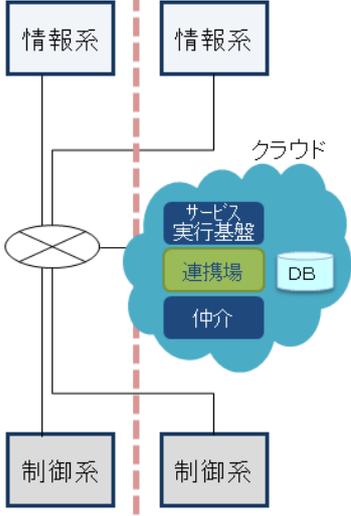
なおここで3.1にて述べた本研究にて提案する異種システム連携コンセプト(図3-1)に基づくシステムを実現するために、3.6にて述べたシステムリファレンスアーキテクチャ

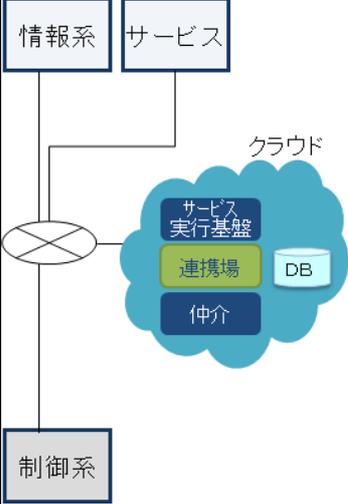
(図 3-24) を導出した。さらに上記システムリファレンスアーキテクチャを具体化するために、4.2 にて述べたソフトウェア構成 (図 4-1) を導出し、これを 4.3 にて述べたハードウェア構成パターンに当てはめて、4.4 にて述べた機能配置パターン (表 4-2) とて整理した。表 4-16 にて示すシステム構成パターンは、4.7.1 に述べた想定されるシステム連携の観点から、上記の機能配置パターンをさらに具体化したものである。

表 4-16 システム構成パターン

#	連携パターン	ネットワーク型	DB サーバ型	クラウド型
(1)	情報 制御 連携	<p><b>【概要】</b></p>  <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>システム間接続のための設備 (GW 等) の設置要。</li> <li>信頼性・性能保証のためには限られた範囲内でシステム構築すべき。</li> </ul>	<p><b>【概要】</b></p>  <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>DB サーバがシステムのボトルネック箇所となる。</li> <li>システム間接続に性能限界あり。</li> </ul>	<p><b>【概要】</b></p>  <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>情報系及び制御系にクラウドにアクセスするプログラムの導入必要。</li> <li>システム間接続に性能限界あり。</li> <li>セキュリティ保証が必要。</li> <li>クラウド上に公開可能なデータの選定要。</li> </ul>

#	連携パターン	ネットワーク型	DB サーバ型	クラウド型
(2)	同種連携	<p><b>【概要】</b></p> <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>システム間接続のための設備（GW等）の設置要。</li> <li>信頼性・性能保証のためには、限られた範囲内でシステム構築すべき。</li> </ul>	<p><b>【概要】</b></p> <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>DBサーバがシステムのボトルネック箇所となる。</li> <li>システム間接続に性能限界あり。</li> </ul>	<p><b>【概要】</b></p> <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>情報系及び制御系にクラウドにアクセスするプログラムの導入必要。</li> <li>システム間接続に性能限界あり。</li> <li>セキュリティ保証が必要。</li> <li>クラウド上に公開可能なデータの選定要。</li> </ul>
(3)	既設連携、新旧並存	<p><b>【概要】</b></p>	<p><b>【概要】</b></p>	<p>—</p>

#	連携パターン	ネットワーク型	DB サーバ型	クラウド型
		<p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>システム間接続のための設備（GW 等）の設置要。</li> <li>信頼性・性能保証のためには、信頼性・性能保証のためには限られた範囲内でシステム構築すべき。</li> <li>既設情報系/制御系の改修不要とするために、GW（仲介）での対応必要。</li> </ul>	<p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>DB サーバがシステムのボトルネック箇所となる。</li> <li>システム間接続に性能限界あり。</li> <li>既設情報系/制御系の改修不要とするために、GW（仲介）での対応必要。</li> </ul>	
(4)	異主体連携	—	<p><b>【概要】</b></p>  <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>DB サーバがシステムのボトルネック箇所となる。</li> <li>システム間接続に性能限界あり。</li> <li>DB サーバの管理主体、運用方法の明確化要。</li> </ul>	<p><b>【概要】</b></p>  <p><b>【制約】</b></p> <ul style="list-style-type: none"> <li>情報系及び制御系にクラウドにアクセスするプログラムの導入必要。</li> <li>システム間接続に性能限界あり。</li> <li>セキュリティ保証が必要。</li> <li>クラウド上に公開可能なデータの選定要。</li> </ul>

#	連携パターン	ネットワーク型	DB サーバ型	クラウド型
(5)	サービス事業者連携	—	—	<p>【概要】</p>  <p>【制約】</p> <ul style="list-style-type: none"> <li>• 情報系及び制御系にクラウドにアクセスするプログラムの導入必要。</li> <li>• セキュリティ保証が必要。</li> <li>• クラウド上に公開可能なデータの選定要。</li> </ul>

#### 4.8. 実現方式の評価

本研究にて提案する技術に関して、実現方式及び実施手段の観点から有用性を評価する。以降では本研究にて提案する技術の一部分に対して実施した、定量的な評価及び定性的な評価の内容について述べる。

##### 4.8.1. 実現方式の有用性の定量評価

本研究にて提案する実現方式による技術を用いることで得られる効果を、定量的に評価する。

###### 4.8.1.1. 評価対象

4.6.2.2(1)にて述べた異種ミドルウェア連携における規格間相互変換を構成する「データモデル変換」に関して評価する。本機能では、様々な異種システム間のデータモデル変換に対応し、処理作成のソフトウェア生産性を向上させるために、一連の処理をフレームワーク化し、システム毎の個別データ仕様に対応するための定義や処理関数のみをユーザ

が作成し組み込むこととしている。

ゆえにここでは従来方法と比較して、本機能を活用することによるソフトウェア生産性の向上度合いを評価する。

#### 4.8.1.2. 評価方法

##### (1) 評価方法

従来方式として、異種システム間のデータモデル変換処理をアプリケーションとして全て作り込む場合と、本研究の提案技術によるデータモデル変換機能（ミドルウェアとして提供）を用いる場合とで比較する。ここではユーザが実際に作成する部分のソフトウェア規模が小さくなる程、ソフトウェア生産性は向上する、とする。これにより従来方式と本研究の提案技術による機能を活用する場合とで、ユーザが作成する部分のソフトウェア規模を比較することによりソフトウェア生産性を評価する。

##### (2) ファンクション・ポイント法

(1)にて述べた、異種システム間のデータモデル変換処理におけるユーザが作成する部分のソフトウェア規模を算出するために、ファンクション・ポイント法を用いる。

ファンクション・ポイント法<sup>59)60)</sup>は、ソフトウェア規模を測定する手法の1つであり、開発工数見積り等に利用される。ソフトウェアの機能を基にして、機能の数や複雑さ等からファンクション・ポイントという点数を付けていき、全ての機能のポイントを合計して規模を導き出すものである。実装技術（プログラミング言語、開発支援ツール、等）には左右されない。計算方法を以下に述べる。

まずソフトウェアが持つ機能の数を洗い出す。ソフトウェアが持つ機能（ファンクション）の種別を図 4-17 に示す。情報（データファンクション）として、内部論理ファイル（Internal Logical File：ILF）、外部インタフェースファイル（External Interface File：EIF）とがある。また情報の出し入れ（トランザクションファンクション）として、外部入力（External Input：EI）、外部出力（External Output：EO）、外部照会（External inquiry：EQ）とがある。

洗い出した上記の種別の機能に対して、表 4-17 に従って複雑さから重み付けを行う。重み付けした点数（ファンクション・ポイント）を集計して、そのソフトウェアの規模を算出する。

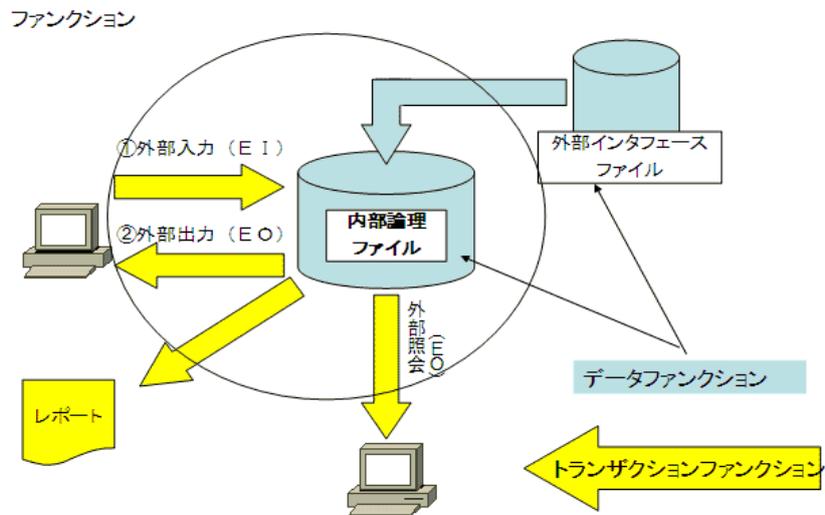


図 4-17 ファンクション・ポイント法：ファンクションの種別（参考文献 57）より引用）

表 4-17 ファンクション・ポイント法：ファンクションの重み（参考文献 60）より引用）

ファンクション型		複雑度		
		低	中	高
データ ファンクション	内部論理ファイル (ILF)	7	10	15
	外部インターフェースファイル (EIF)	5	7	10
トランザクション ファンクション	外部入力 (EI)	3	4	6
	外部出力 (EO)	4	5	7
	外部照会 (EQ)	3	4	6

### 4.8.1.3. 評価結果

#### (1) 比較

4.8.1.2(2)にて述べたファンクション・ポイント法を用いて、異種システム間のデータモデル変換処理におけるユーザが作成する部分のソフトウェア規模を算出した結果を図 4-18 に示す。4.6.2.2(1)にて述べた本研究にて提案するデータモデル変換に関して図 4-9 に示した方式を基にして、従来方式として、異種システム間のデータモデル変換処理をアプリケーションとして全て作り込む場合と、本研究の提案技術によるデータモデル変換機能（ミドルウェアとして提供）を用いる場合とで、それぞれユーザが作成する部分の機能を洗い出して、ファンクション・ポイントを算出、集計している。

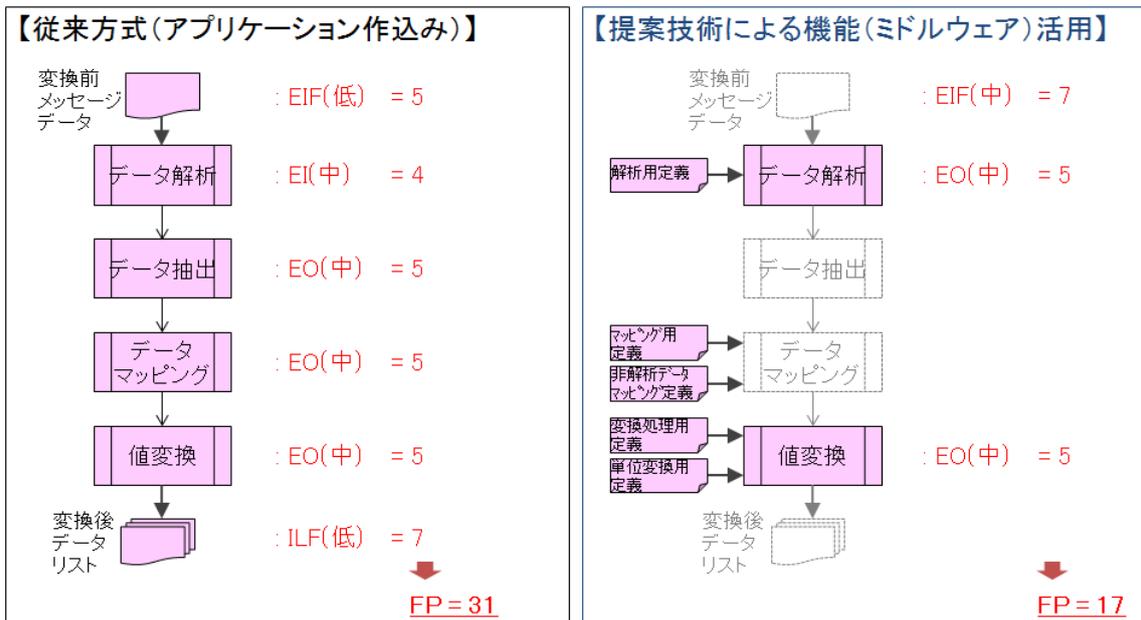


図 4-18 ファンクション・ポイントの比較

$$([\text{従来方式の FP}] - [\text{提案方式の FP}]) / [\text{従来方式の FP}] = (31 - 17) / 31 = 0.45$$

(2) まとめ

図 4-18 に示した結果から上式より、従来方式によるアプリケーション作り込みの場合に比べて、本研究の提案技術による機能を用いる場合では、ソフトウェア規模が 45%削減した。これによりソフトウェア生産性が向上したと言える。

#### 4.8.2. 実現方式の有用性の定性評価

本研究にて提案する実現方式による技術を用いることで得られる効果を、定性的に評価する。

##### 4.8.2.1. 評価対象

4.6.2.2(1)にて述べた異種ミドルウェア連携における規格間相互変換に関して評価する。本機能は、異なる環境にあり準拠規格等が異なるシステムまたは機器が相互接続、連携して動作するために、各システムにて準拠する規格（データモデル、インタフェース、プロトコル）の相互変換を行う。ここでは異なる環境にあり準拠規格等が異なるシステム間の連携動作によりサービスを行うユースケースに、実際に本機能が適用できるかを評価する。

##### 4.8.2.2. 評価結果

###### (1) 提案技術の適用評価

本研究の提案技術である、4.6.2.2(1)にて述べた異種ミドルウェア連携における規格間相互変換は、以下に述べるスマートシティ IT 基盤の機能の1つとして実装し、スマートグリッド環境でのユースケースに適用している<sup>63)64)</sup>。

スマートシティ向け IT 基盤の概要を図 4-19 に示す。本基盤は、社会インフラにおける供給側と需要側のシステムとを相互連携するエネルギーマネジメントシステム等に導入する。本基盤では、多種多様な機器からの情報を容易に収集・蓄積する機能、各種の業務アプリケーションに上記情報を提供する機能、機器のメンテナンスの効率化のための機能、等を提供する。特にエネルギー、水、モビリティ等の社会インフラに対応した設備の構成情報、稼働情報（状態、履歴）、計画情報を汎用データモデルとして管理する。また機器とアプリケーションの間での信頼性を保証した高速な情報伝達、高度なセキュリティを提供する。さらに制御システムと情報システムとの連携や異なる標準規格間の相互接続、等も実施する。

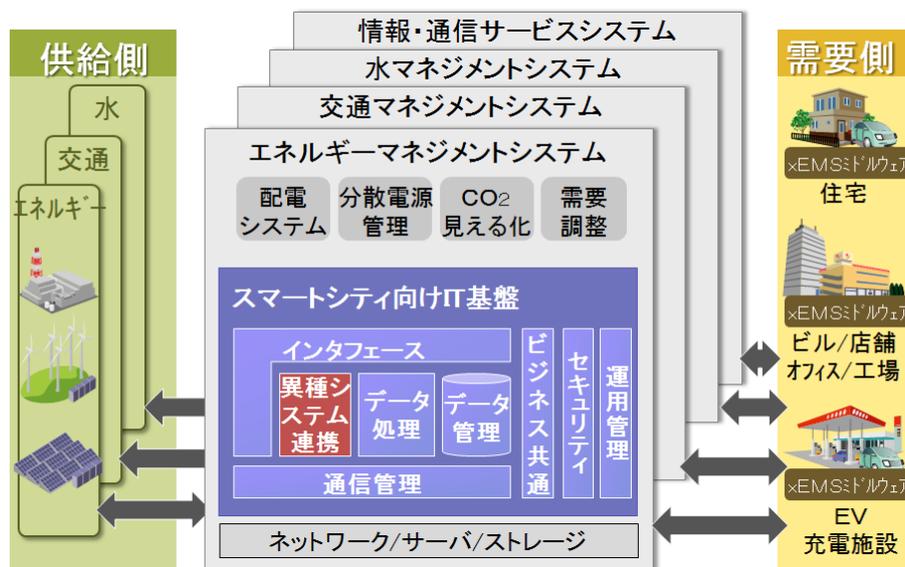


図 4-19 スマートシティ向け IT 基盤の概要（筆者文献 64）より引用

上記のスマートシティ向け IT 基盤の適用事例の1つとして、再生可能エネルギーや Electric Vehicle (EV) を大幅に導入するスマートグリッド環境でのソリューションに取り組んでいる。ユースケースの1つとして、上記環境における、地域全体での省エネ・再生可能エネルギー有効利用のための、異なる事業者の、異なる業務システム間の連携によるデマンドレスポンスの例を挙げる。例えば電力会社の需給調整システムと、EV 会社の EV 管理・充電ステーション管理システムとの連携である。本ユースケースにおけるシステム構成例を図 4-20 に示す。需給調整システムはスマートシティ向け IT 基盤上に構築し、ネットワーク（インターネット、M2M 等）を介して、業務実施のために多数の家庭やビル等の需要家との間でデータ収集や配信を行う。また配電管理システムが配電系統設備の監視制

御を実施するために配電系統設備から収集するデータもスマートシティ向け IT 基盤を介して参照する。EV 管理・充電ステーション管理システムも同様にスマートシティ向け IT 基盤上に構築し、ネットワークを介して、業務実施のために EV や充電ステーションとの間でデータ収集や配信を行う。

再生可能エネルギーを使用する環境にて、電力会社では需給調整システムにより配電系統安定化や需給調整のためのデマンドレスポンスを実施する。このため継続的に家庭、ビル等の需要家設備や配電設備の稼働情報や電力使用状況情報を収集し、需要予測や制御のための計画作成等に使用する。上記により作成された計画情報は需要家側へと配信される。

EV 会社では EV 管理・充電ステーション管理システムが充電ステーションの充電設備の状態管理や充放電制御、個々の EV の状態管理、EV の最適な充電ステーションへの誘導等を実施する。このために同じく継続的に充電ステーション設備や EV 機器の稼働情報等を収集している。

ここで需給調整システムがデマンドレスポンスをより高度に実施するために、EV 管理・充電ステーション管理システムが管理する充電ステーションの稼働情報や EV 充電スケジュール等を定期的に取り得し需要予測やデマンドレスポンス計画作成に活用する。また EV 管理・充電ステーション管理システムも最適な EV の誘導・充電スケジュールを作成するために需給調整システムが管理する需給予測情報や給電上限電力情報等を定期的に取り得する。上記のように事業者を跨いで業務システム間でデータのやり取りを行う。

上記ユースケースにおける基盤の要件の 1 つが、配電系統機器や家庭内機器または EV や充電ステーション設備、等の異なるシステムにおける多種多様な機器と接続すること、である。つまり各々異なる標準規格に準拠する個々のフィールド側の制御システム（需要家設備を制御するシステム、配電系統設備を制御するシステム、等）とセンタ側の他業務システムとの接続、連携が必要となる。

上記要件に対応するために、本研究の提案技術である異種ミドルウェア連携における規格間相互変換を、スマートシティ IT 基盤にて実装した機能を用いた。

また本研究の提案技術である異種ミドルウェア連携における規格間相互変換をスマートシティ IT 基盤の機能にて実装する上で、以下を実現している。

- 接続先システムや標準規格の増加に対する開発負荷の低減
- 変換処理のユーザ定義組込み可能なフレームワーク化によるソフトウェア生産性向上（4.8.1 参照）
- データモデル変換の一例として、IEC61850<sup>52)53)54)</sup>と 4W1H 実世界情報モデル<sup>49)50)</sup>との変換（図 4-21 に示す）

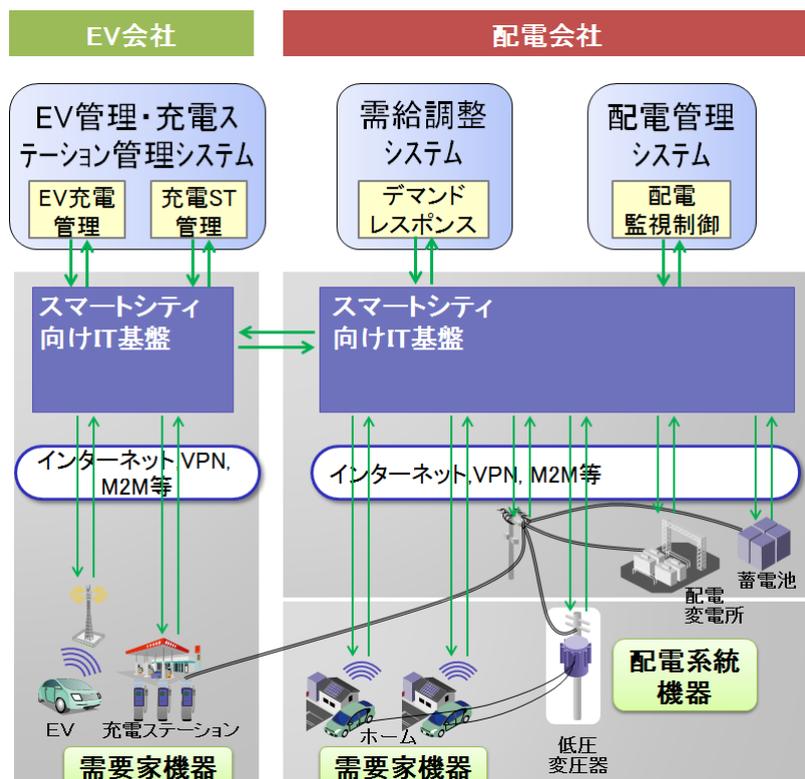


図 4-20 スマートグリッド環境への適用 (筆者文献 64) より引用、一部加筆)

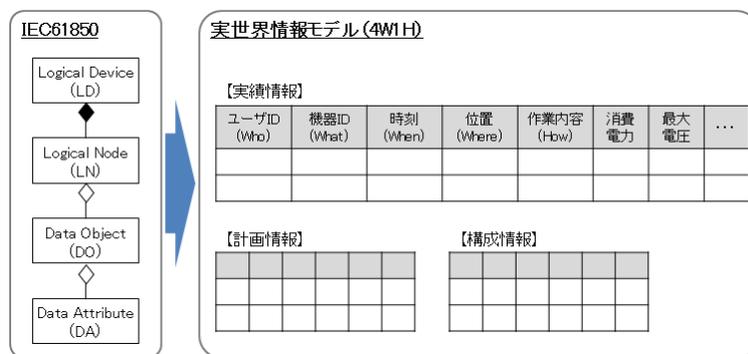


図 4-21 規格間相互変換の例 (筆者文献 64) より引用)

(2) まとめ

4.8.2.2(1)にて述べた結果からに、本研究の提案技術である異種ミドルウェア連携における規格間相互変換は、スマートシティ IT 基盤の一機能として実装し、一例としてスマートグリッド環境での実際のユースケースに適用することで有用性を示した。

上述も含めて本研究にて提案する動的グループ形成・制御、異種ミドルウェア連携の技術の一部分は、製品化し、産業、電力・スマートグリッド等での実システムに実適用済み

である 61)62)64)。<sup>4</sup>

<sup>4</sup>本適用技術に関連して、主に以下の特許を出願済みである。

#	提案技術	登録番号	名称	登録先	備考
1	動的グループ 形成・制御	PZL03158737.2	分散システムおよびコンテキスト対 応ブローカリング方法	中国	
2	異種ミドルウ ェア連携	仲介	P04498262	分散システムにおける情報共有方法 及び情報共有システム	日本
3		仲介	P04808173	分散システム、データ管理サーバ及 びデータ流通方法	日本、中 国、台湾
4		規格間相互接 続	P05809743	分散システムにおける異種システム データ提供方法	日本

## 5. 実問題における適用例と評価

本章では、第3章及び第4章にて述べた、本研究にて提案する異種システム連携技術を適用すると効果が高いと見込まれる実問題における想定ユースケース例と、それらの想定ユースケース例への適用検証を通しての本研究の提案技術の有用性に関する評価結果について述べる。本章にて扱う想定ユースケース例を表5-1にまとめる。各々の想定ユースケース例に本研究にて提案する異種システム連携技術を適用する場合の具体例及び評価結果を以降の節にて述べていく。

表 5-1 想定ユースケース例

分野	ユースケース	概要	評価の目的
社会インフラ (電力)	電力需給調整・スマートグリッド (Demand Response、Virtual Power Plant)	再生可能エネルギー(風力、太陽光)を導入する環境において、需給バランス平準化、エネルギー消費効率化のための、需要家(家庭、ビル、等)の段階的な制御。	動的グループ形成・メンバー代替の事例パターン例示。 (特にグループの範囲、制御の粒度が変わる場合)
航空宇宙	惑星探査	複数の異種の探査機(自律走行車、自律飛翔体、観測設備等)の連携・協調動作により、観測、等のミッションを遂行。	上記と同等の検証のため。

### 5.1. 社会インフラ(電力): 電力需給調整・スマートグリッド

#### 5.1.1. 概要

##### 5.1.1.1. Demand Response の概要

電力システムにおいて、発電設備にて生成された電力は、送電系統や配電系統を経て需要側へと供給されるが、電力の安定供給は供給側の発電量と需要側の負荷量のバランスの上に成り立っている。ここで電力の需給バランスを維持する方法の1つとして、供給側が需要側に電力使用を奨励したり抑制したりするのが Demand Response<sup>65)66)</sup> (以下、DR と記す) である。特に電力供給逼迫時において、DRにより電力需要を削減させたりシフトさせたりすることは、電力の安定供給に有効である。電力供給側の視点に立つと、DRは以下の3つに分類される。

#### ① キャパシティ補償

需要のピーク時に系統の信頼性を向上させるために発動される。DRが発動されると、需要家は一度に数時間程度の需要量の削減を行う。図5-1に概要を示す<sup>67)</sup>。

## ② 瞬時バランス補償

発電機や送電網の事故等で突発的に電力の需給バランスが崩れることに対して需要量の削減を行う。通知後直ちに需要量の削減が必要となる。

## ③ 経済性補償

需要家は各々の裁量で需要量の削減を入札し、供給側にとってより経済的なものが落札される。これにより燃費効率の悪い発電設備の稼働時間の削減が期待できる。

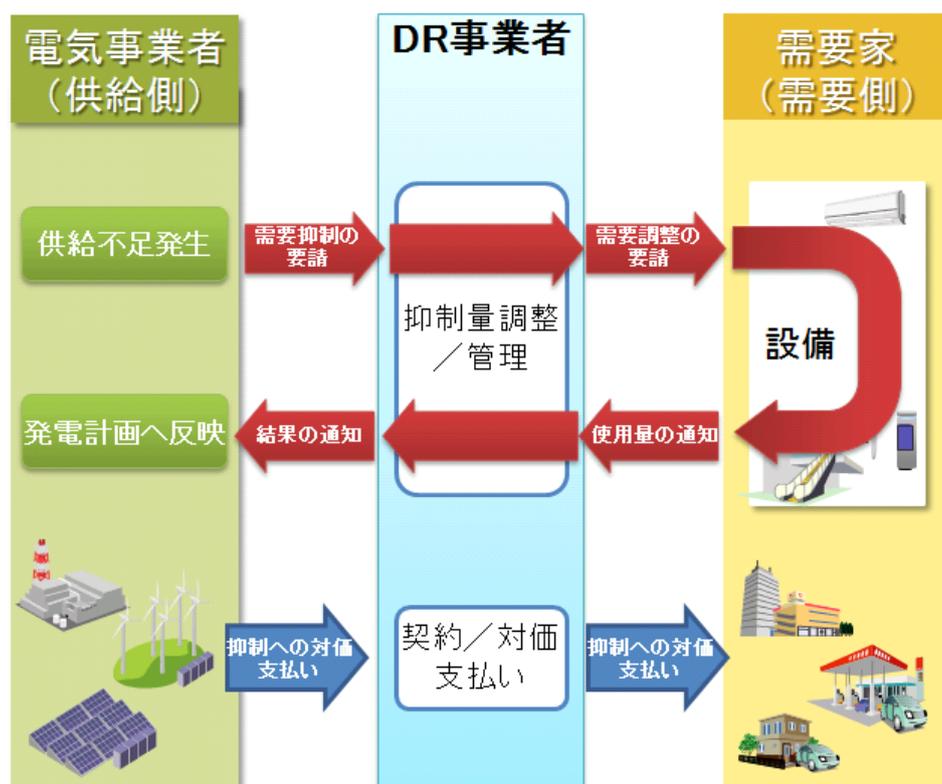


図 5-1 Demand Response (DR)の概要 (参考文献 67)より

上記 DR の 3 つの分類のうち、以下では特に①キャパシティ補償に関して扱う。なお②瞬時バランス補償は、主に電力系統設備側で実施するものである。③経済性補償は、主に電力市場取引にて実施されるものである。

①キャパシティ補償型の DR のビジネス形態を図 5-2 に示す<sup>67)</sup>。電力需給逼迫時に電気事業者から電力調整依頼を受け、需給統合計画事業者が取りまとめる需給統合計画に基づき、アグリゲータが提供する Building Energy Management System (BEMS) 等を活用して、節電対象となる空調機器等のビル設備のピーク電力を抑制する。ここでのアグリゲータとは、需要家個々と電力需要を抑制する契約を締結し、契約した需要家全体で求められる節電量を達成するために、需要家の使用電力の調整を実施する事業者のことをいう。

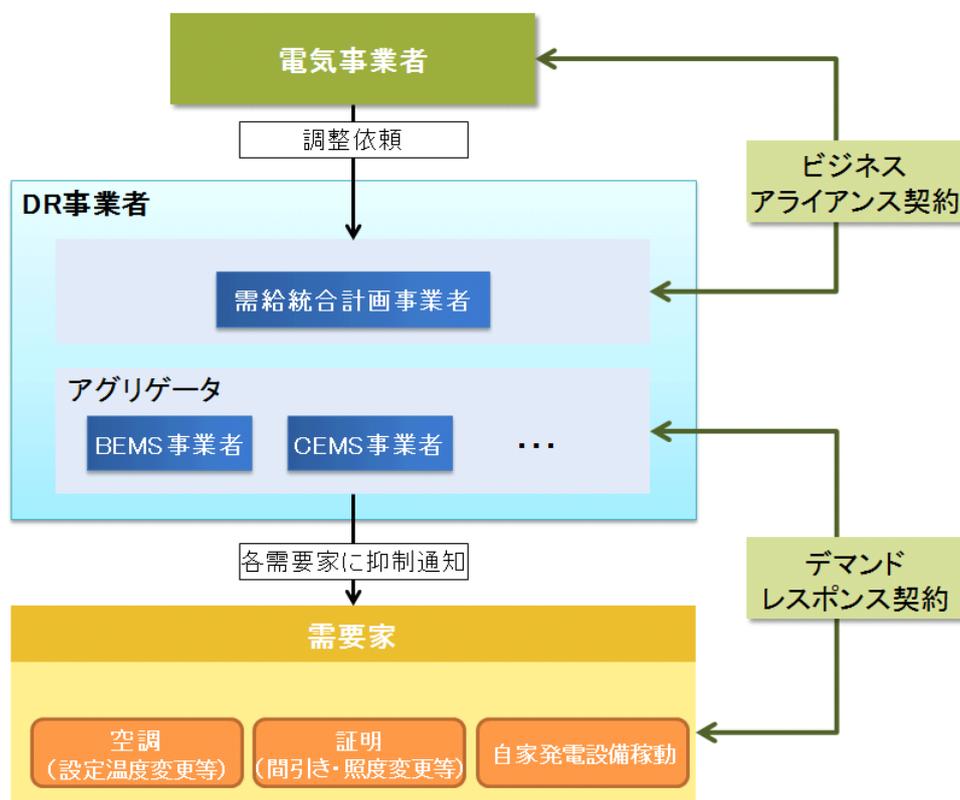


図 5-2 キャパシティ補償型 DR のビジネス形態 (参考文献 67) より)

### 5.1.1.2. Virtual Power Plant の概要

5.1.1.1 にて述べた DR を応用した主要なサービスの 1 つとして Virtual Power Plant (以下、VPP と記す) が挙げられる<sup>68)69)</sup>。図 5-3 に VPP の概要を示す。VPP は、分散型の電源となる需要家設備を、ネットワークを介して集約し制御・運用、管理することで電力供給力を確保するシステムである。

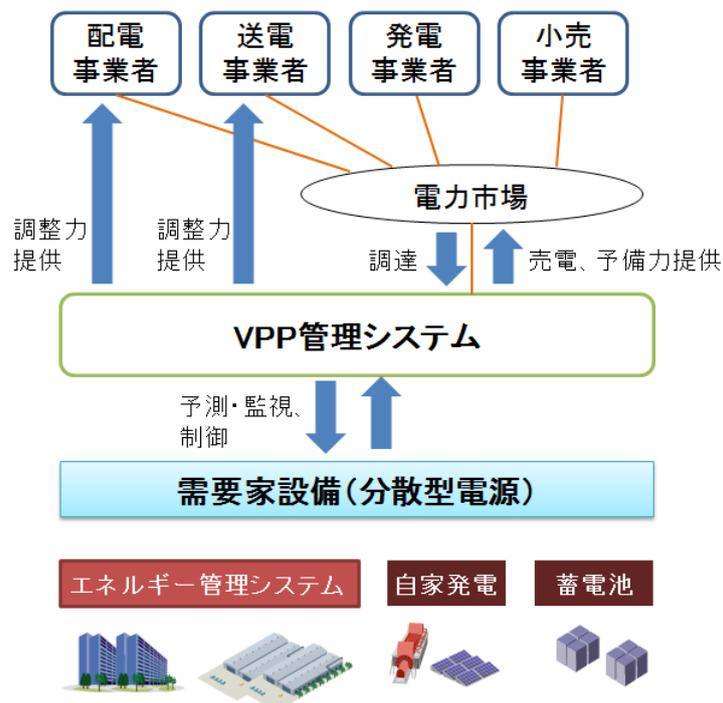


図 5-3 Virtual Power Plant の概要

### 5.1.1.3. Virtual Power Plant の必要性

5. 1. 1. 2 にて述べた VPP に期待される主なユースケースを表 5-2 にまとめる。

表 5-2 Virtual Power Plant のユースケース

#	ユースケース	概要
(1)	予備力 <sup>70)</sup> 提供	VPP を構成するシステム内での分散電源からの放電や需要家設備の電力使用抑制（ネガワット）により、送電事業者の要請に応じて予備力提供を行う。
(2)	系統安定	電力系統網（配電網、送電網）において過負荷や電力量の上限逸脱の発生により同時同量のバランスが崩れて不安定になることを回避するために、構成するシステム内の蓄電池への充電や需要家設備での電力使用増加により系統から余剰の電力を吸収する。
(3)	経済的取引	電力小売事業者を介して電力市場にて取引を行い、取引結果に基づいて VPP を構成するシステム内での分散電源からの放電や需要家設備の電力使用抑制（ネガワット）により調達した電力の売電を行う。

表 5-2 のユースケース(1)予備力提供に関して、電力システムを安定稼働させるためには電力の需要（消費）と供給（発電）のバランスを常に保つ必要がある。このため電力会社は、毎日、電力需要の量とパターン（何時ごろに、どれくらい電気が使われるか）を予

測し、それに合わせて発電所毎の出力パターン（何時ごろに、どの発電所をどれくらいの出力で動かすか）を計画する。この際、突発的な事象（設備故障による急な発電量低下や、急激な気温変化による需要変化、等）に対応できるように、全ての発電所をフル出力にせず、幾つかの発電所では出力に余裕のある状態にしておき、いつでも起動できる発電所を決めて待機させておく。このように予備力とは、予想される最大電力需要に対して、さらにどれくらい発電能力の余力があるかを表すものである<sup>69)</sup>。

ここで火力発電所等の従来からの大型発電設備を稼働し出力させるためには、運用等のコストと時間がかかり、急な電力供給量低下に対してすぐに対応できない場合がある。特に風力、太陽光発電等の再生可能エネルギーを導入する電力システムである場合、電力供給量が低下する事象はより発生しやすい。また需要を満たすための計画に対して電力供給量を達成できなかった場合、発電事業者にはペナルティ料金が課せられて、発電事業者の経営を圧迫することとなる。

上記に対して VPP が従来からの大型発電設備のようなコストや時間をかけずに予備力を提供できることは、再生可能エネルギーを導入する電力システムにおける電力の安定供給や発電事業者の経営にとって有用となる。

表 5-2 のユースケース(2)系統安定に関して、例えば地域の配電系統等の限られた範囲内で電力需給のバランスを維持し、電力システム全体への影響波及を回避するのに有用である。

表 5-2 のユースケース(3)経済的取引に関して、電力自由化、電力市場・取引の活性化を担う 1つのプレイヤーとなり得る。

#### 5.1.1.4. Virtual Power Plant の構成

5.1.1.2、5.1.1.3 にて述べた VPP を実現するための構成を図 5-4 に示す。初期の VPP は主にビル、工場等の大口の需要家に対する DR 実施、もしくは系統につながる比較的規模の大きいに限られた数の自家発電機や蓄電池を対象として構成していた。

一方、今後は、一般家庭等の小口だが巨多の需要家に対する DR 実施、小規模の自家発電機や蓄電池も対象として範囲を拡張して構成していくものと考えられる。また個々の需要家は常に VPP の構成要素として稼働するのではなく、VPP を運営する DR 事業者との契約内容や各需要家の保有する設備の稼働状況等に応じて、柔軟に VPP システムに加入及び離脱するものとなる。

## 【従来】



## 【今後】

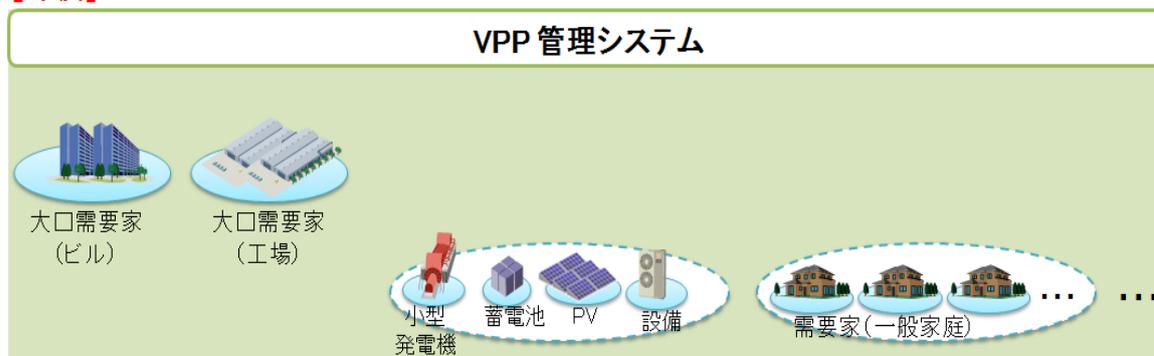


図 5-4 Virtual Power Plant の構成

特に上述のような今後の VPP を実現するためには、以下の要件を満たす必要がある。

- DR 対象となる需要家設備は状況に応じて稼働状態や電力使用状態が変動し得る。このため電力供給力確保のために制御対象とする、必要な需要家設備またはエネルギー管理システムの組合せを状況に応じて変更する必要がある。
- 需要家及び設備の種別により電力使用の規模や特質が異なるため、個々の性質に応じた状態監視と制御が必要となる。
- 再生可能エネルギーを使用する系統である場合、電力需給状態が不規則に変動する可能性が高く、VPP の管理する範囲内で完結して対応できない場合もあり、電力市場を介しての他事業者・他システムとの連携も必要となる。

### 5.1.2. 本研究の適用方法

#### 5.1.2.1. 動的グループ形成・制御の適用

5.1.1.4 にて述べたように、規模が異なる、巨多の需要家及び設備を対象として VPP を構成するために、3.2 及び 4.6.1 にて述べた、本研究にて提案する動的グループ形成・制御の

技術を適用する。なお需要家の規模の違いによって構成するグループを分けて多段の構成とする。システム構成を図 5-5 に示す。

図 5-5 に示すように、ビルには Building Energy Management System (BEMS)<sup>71)72)</sup>、工場には Factory Energy Management System (FEMS)<sup>72)73)</sup> が導入され、ビル内及び工場内の設備による電力使用及び発電装置による発電、蓄電池の放電・充電を一元管理、制御するものとする。小規模の発電装置や蓄電池等を保有、運営する地域には Area Energy Management System (AEMS)<sup>74)75)</sup> が導入され、地域内の発電装置、蓄電池、電力負荷等の制御を行い、地域内の電力の使用及び電力系統への電力提供等を一元管理、制御するものとする。また個々の一般家庭には家庭内の設備による電力使用を管理、制御する Home Energy Management System (HEMS)<sup>71)72)</sup> を導入するものとする。さらに地域毎に複数の一般家庭の HEMS をまとめて管理するのに AEMS を導入するものとする。これらの BEMS、FEMS、AEMS を管理し、連携させることで VPP を実現するための中核となる VPP 管理システムを導入する。VPP 管理システム及び BEMS、FEMS、AEMS 及び HEMS はそれぞれネットワークを介して相互接続できるものとする。

上記のシステム構成において、VPP 管理システムにコンダクタ、モニタを割り当て、BEMS、FEMS、AEMS1 (地域)、AEMS2 (一般家庭) にそれぞれノードを割り当て、これらをメンバとしたグループを形成する。また AEMS1 (地域) にコンダクタ、モニタを割り当て、地域内の各設備にノードを割り当て、これらをメンバとしたサブグループを形成する。同様に AEMS2 (一般家庭) にコンダクタ、モニタを割り当て、各家庭の HEMS にノードを割り当て、これらをメンバとしたサブグループを形成する。

上述のように複数、多段のグループを構成することで、規模の異なる、巨多の設備やシステムに対して、同程度の規模のもの毎に範囲を区切って個々に管理することで、個々の需要家設備の変動に対する影響を局所化して、全体としても効率的な管理を実現できる。また扱う電力需給量の違いやメンバとしての入替の単位の違いに対応する。

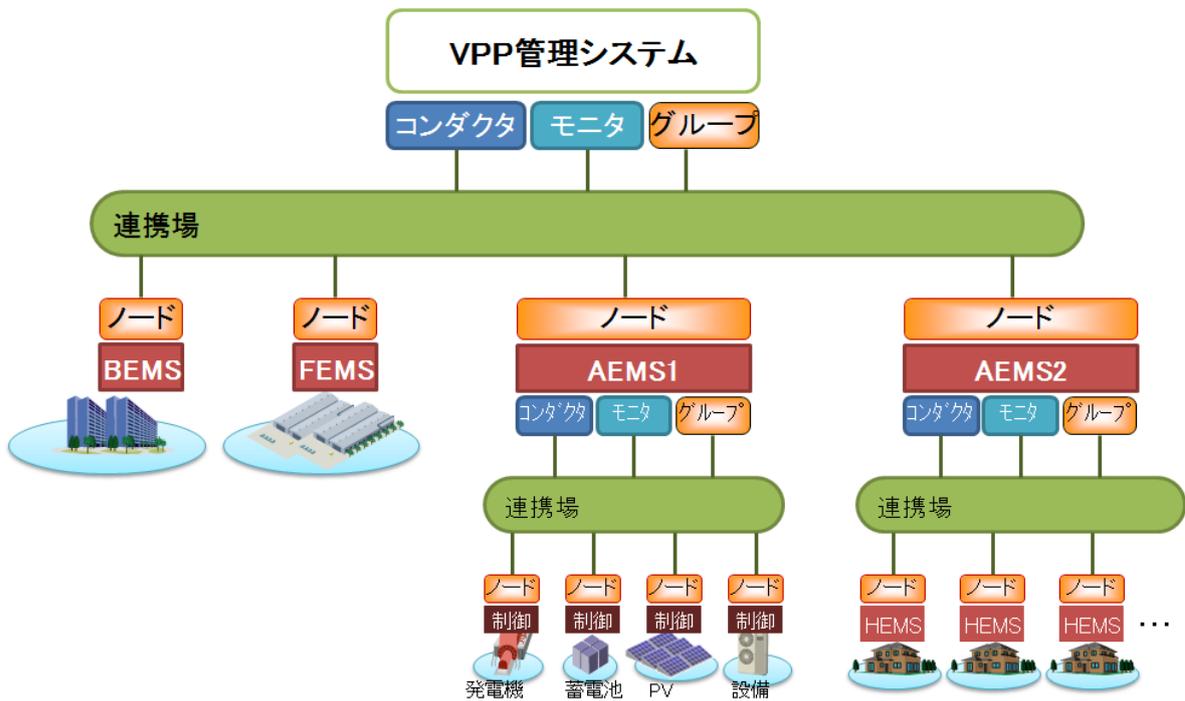


図 5-5 システム構成 (グループ構成)

#### 5.1.2.2. 異種ミドルウェア連携の適用

5.1.1.4にて述べたように、規模が異なる、巨多の需要家及び設備を対象としてVPPを構成するために、3.3及び4.6.2にて述べた、本研究にて提案する異種ミドルウェア連携の技術を適用する。

ここで図 5-6 に示すように電力システム及び需要家設備にはシステム階層や設備の種別毎に多種多様な標準規格が乱立しており、各々異なるデータモデル及びインタフェースが用いられている。このためVPPを構成するために巨多の需要家設備を対象とする場合、各々の標準規格に準拠した、多数の異なるデータモデル及びインタフェースを有する設備を扱う必要がある。この場合、全体でVPPとして動作させるべく個々の設備を管理及び制御するために、VPP管理システムは巨多の設備の個々のデータモデル及びインタフェースに対応することが不可欠である。VPP管理システムにて扱うデータモデル及びインタフェースと設備の個々のデータモデル及びインタフェースとの相互変換を行う必要があり、これらの処理を1つずつアプリケーションとして作り込むことは莫大なコストがかかる。



携を介して共通のデータモデルに変換して AEMS サーバ 1 上に格納し、各ノードが該データを参照、更新することで稼働する。各家庭の HEMS から送受信されるデータは異種ミドルウェア連携を介して共通のデータモデルに変換して AEMS サーバ 2 上に格納し、各ノードが該データを参照、更新することで稼働する。

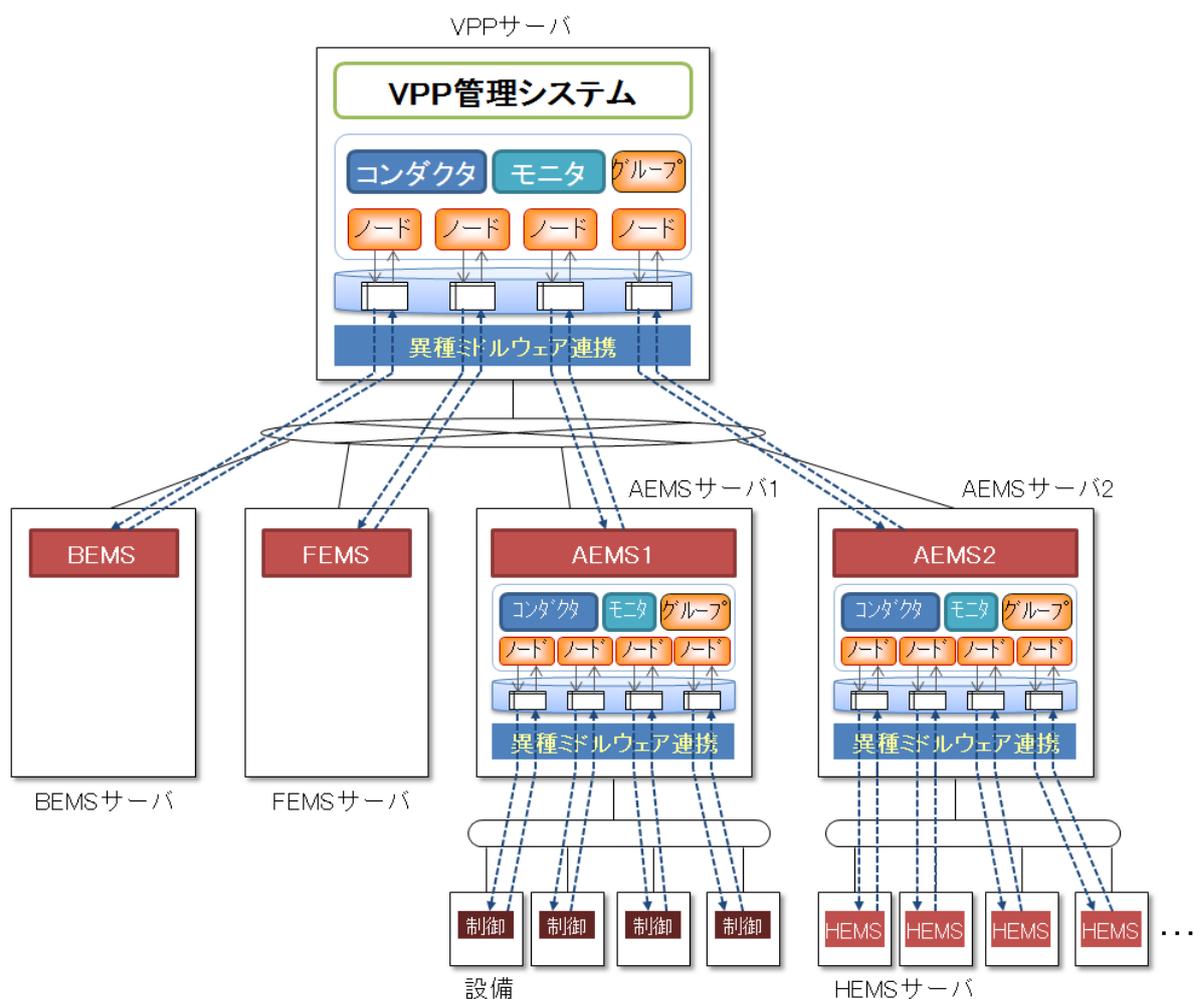


図 5-7 システム構成及び異種ミドルウェア連携の適用

### 5.1.3. ユースケース適用

5.1.1、5.1.2 にて述べた VPP のユースケース例に本研究にて提案する異種システム連携技術を適用する場合のシナリオ及び動作内容について本節で述べる。

#### 5.1.3.1. シナリオ

今回 VPP のユースケースシナリオとして、表 5-2 に挙げたユースケースの中で特に、(1) 予備力提供、(2) 系統安定、を扱う。以降では各々の詳細について述べる。

## (1) 予備力提供

VPPによる予備力提供のシナリオの概要を図 5-8 に示し、シナリオにおける各ステップを以下に述べる。なお以下の①～⑨は図中の①～⑨に対応する。

- ① 電力需給実績 BEMS、FEMS は、それぞれビル内、工場内における電力需給の実績を集計して、定期的に VPP 管理システムに伝達する。  
AEMS1（地域）は、定期的に地域内の各設備の稼働実績を収集して、それらより地域内での電力需給実績を算出し、VPP 管理システムに伝達する。  
AEMS2（一般家庭）は、定期的に個々の HEMS から各々の電力使用実績を収集し、それらを集計して、VPP 管理システムに伝達する。
- ② DR 計画作成 VPP 管理システムは、①の BEMS、FEMS、AEMS1、AEMS2 から収集した電力需給実績を基に今後の電力需給予測を行い、予備力提供のためにシステム内の DR 実施の配分を計画した DR 計画を作成する。
- ③ DR 依頼 VPP 管理システムは、②にて作成した DR 計画を BEMS、FEMS、AEMS1、AEMS2 に対して送信する。
- ④ 設備稼働計画立案 BEMS、FEMS 及び AEMS1（地域）は各々、③の DR 依頼を受け取り、本（BEMS、FEMS、AEMS1）DR 計画の各システム分を達成するための設備稼働計画を作成する。  
ここで各々にて依頼された DR 計画が実施可能であるか否かの判定を行う。
- ④ DR 配分計画立案（AEMS2） AEMS2（一般家庭）は、③の DR 依頼を受け取り、本 DR 計画を達成するための傘下の HEMS に対する DR 配分計画を作成する。
- ⑤ 制御指示 BEMS、FEMS 及び AEMS1（地域）は各々、④で作成した設備稼働計画（BEMS、FEMS、AEMS1）に基づき、システム内の各設備に対して制御指示を実施する。
- ⑤ DR 依頼（AEMS2） AEMS2（一般家庭）は、④で作成した DR 配分計画に基づき、各 HEMS に対して DR 計画を送信する。ここで各 HEMS からの DR 回答結果に基づき VPP 管理システムから依頼された DR 計画が実施可能であるか否かの判定を行う。
- ⑥ DR 回答 BEMS、FEMS、AEMS1（地域）及び AEMS2（一般家庭）は、④設備稼働計画立案、DR 配分計画立案の結果に基づき、VPP 管理システムに対して DR 回答を送信する。DR 回答として、DR 実施可否、DR 実施可能である場合の調達可能な電力量の見込みを含める。

- ⑦ 設備動作 ⑤の制御指示、DR 依頼に従って、需要家設備が動作する。  
例えば、ビルの設備の使用抑制を行い、ネガワットを提供する。工場では発電設備による発電、蓄電池等による放電を行う。地域では発電設備による発電、蓄電池による放電、電力負荷の使用抑制によるネガワット提供を行う。一般家庭では使用抑制を行い、ネガワットを提供する。
- ⑧ 稼働実績／電力使用実績 AEMS1（地域）は、各設備の⑦の設備動作の結果としての稼働実績を収集する。  
AEMS2（一般家庭）は、各 HEMS から⑦の設備動作の結果としての電力使用実績を収集する。
- ⑨ 電力需給実績 BEMS、FEMS、AEMS1（地域）及び AEMS2（一般家庭）は、⑦の設備動作、⑧の稼働実績及び電力使用実績収集の結果として、電力需給実績を算出し、VPP 管理システムに伝達する。  
VPP 管理システムでは、上記実績と計画を比較し、予備力提供の達成可否を評価する。

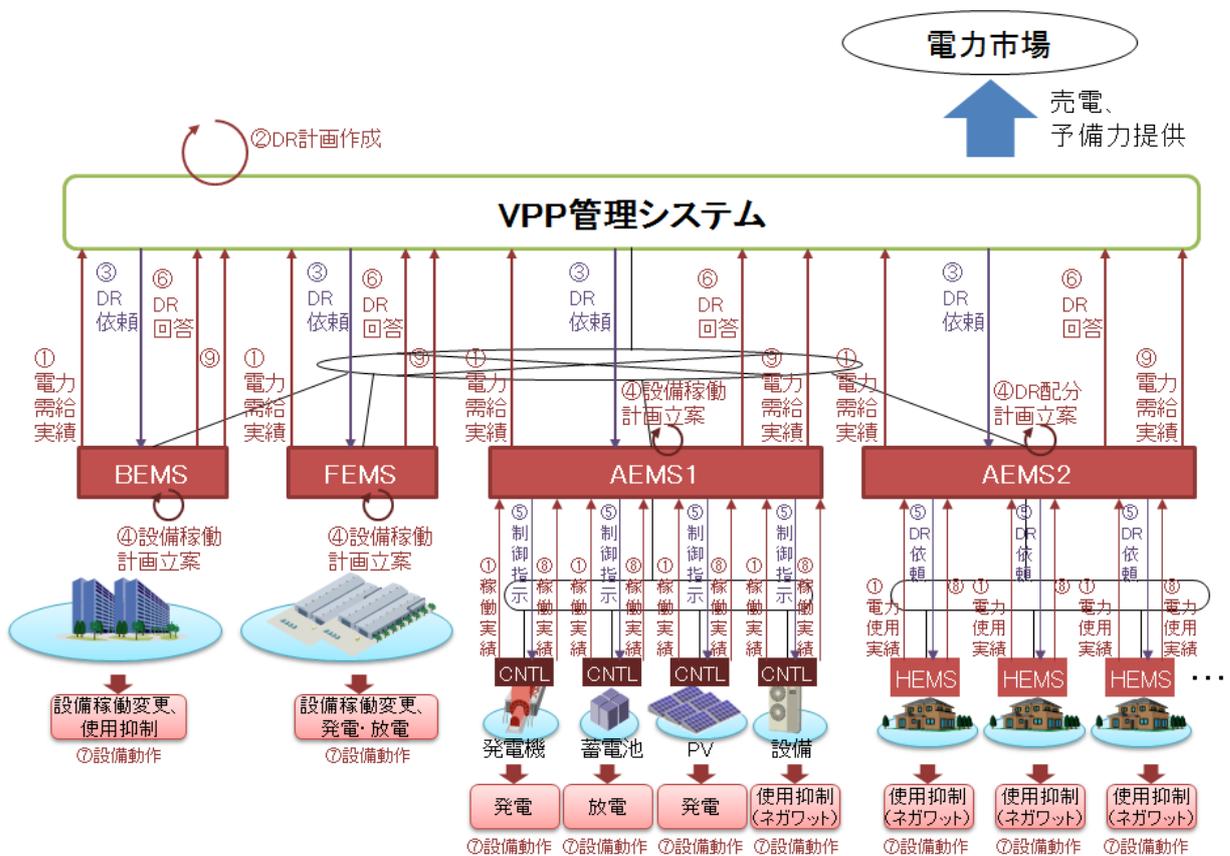


図 5-8 Virtual Power Plant のシナリオ (予備力提供)

## (2) 系統安定

VPP による系統安定のシナリオの概要を図 5-9 に示し、シナリオにおける各ステップを以下に述べる。なお以下の①～⑨は図中の①～⑨に対応する。

- ① 電力需給実績 BEMS、FEMS は、それぞれビル内、工場内における電力需給の実績を集計して、定期的に VPP 管理システムに伝達する。  
AEMS1（地域）は、定期的に地域内の各設備の稼働実績を収集して、それらより地域内での電力需給実績を算出し、VPP 管理システムに伝達する。  
AEMS2（一般家庭）は、定期的に個々の HEMS から各々の電力使用実績を収集し、それらを集計して、VPP 管理システムに伝達する。
- ② DR 計画作成 VPP 管理システムは、①の BEMS、FEMS、AEMS1、AEMS2 から収集した電力需給実績を基に今後の電力需給予測を行い、系統安定のためにシステム内の DR 実施の配分を計画した DR 計画を作成する。
- ③ DR 依頼 VPP 管理システムは、②にて作成した DR 計画を BEMS、FEMS、AEMS1、AEMS2 に対して送信する。
- ④ 設備稼働計画立案 BEMS、FEMS 及び AEMS1（地域）は各々、③の DR 依頼を受け取り、本（BEMS、FEMS、AEMS1）DR 計画の各システム分を達成するための設備稼働計画を作成する。  
ここで各々にて依頼された DR 計画が実施可能であるか否かの判定を行う。
- ④ DR 配分計画立案（AEMS2） AEMS2（一般家庭）は、③の DR 依頼を受け取り、本 DR 計画を達成するための傘下の HEMS に対する DR 配分計画を作成する。
- ⑤ 制御指示 BEMS、FEMS 及び AEMS1（地域）は各々、④で作成した設備稼働計画（BEMS、FEMS、AEMS1）に基づき、システム内の各設備に対して制御指示を実施する。
- ⑤ DR 依頼（AEMS2） AEMS2（一般家庭）は、④で作成した DR 配分計画に基づき、各 HEMS に対して DR 計画を送信する。ここで各 HEMS からの DR 回答結果に基づき VPP 管理システムから依頼された DR 計画が実施可能であるか否かの判定を行う。
- ⑥ DR 回答 BEMS、FEMS、AEMS1（地域）及び AEMS2（一般家庭）は、④設備稼働計画立案、DR 配分計画立案の結果に基づき、VPP 管理システムに対して DR 回答を送信する。DR 回答として、DR 実施可否、DR 実施可能である場合の吸収可能な電力量の見込みを含める。

⑦ 設備動作

⑤の制御指示、DR 依頼に従って、需要家設備が動作する。

例えば、ビルの設備の使用を行い、電力を消費する。工場では発電設備による発電の抑制、蓄電池等による充電を行う。地域では発電設備による発電の抑制、蓄電池による充電、電力負荷の使用による電力消費を行う。一般家庭では使用による電力消費を行う。

⑧ 稼働実績／電力使用実績

AEMS1（地域）は、各設備の⑦の設備動作の結果としての稼働実績を収集する。

AEMS2（一般家庭）は、各 HEMS から⑦の設備動作の結果としての電力使用実績を収集する。

⑨ 電力需給実績

BEMS、FEMS、AEMS1（地域）及び AEMS2（一般家庭）は、⑦の設備動作、⑧の稼働実績及び電力使用実績収集の結果として、電力需給実績を算出し、VPP 管理システムに伝達する。

VPP 管理システムでは、上記実績と計画を比較し、系統安定の達成可否を評価する。

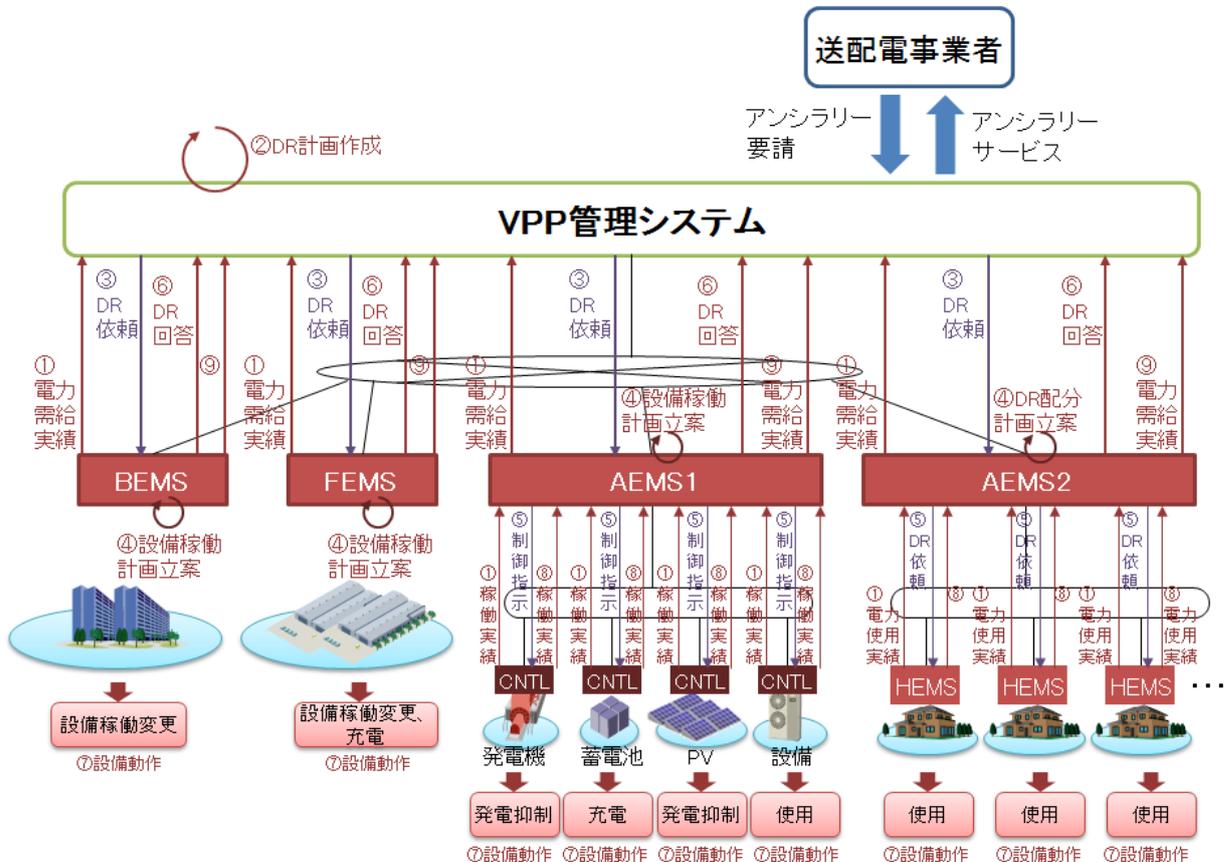


図 5-9 Virtual Power Plant のシナリオ (系統安定)

### 5.1.3.2. 動作

本研究にて提案する異種システム連携技術を適用する場合の、5.1.3.1にて述べたVPPのユースケースシナリオを実行する場合の動作内容について述べる。

#### (1) 予備力提供

図5-7に示したシステム構成において、5.1.2にて述べた本研究の適用方法に従って、5.1.3.1(1)にて述べたVPPによる予備力提供のシナリオを実行する場合の処理の流れを図5-10に示す。これらの処理の流れは4.6.1.2にて述べたグループ動作の方式に従う。

VPP管理システムのコンダクタ及びモニタが予備力提供のためのサービスシナリオを読み込む。コンダクタは上記サービスシナリオに基づきサービス実行に必要なノード（BEMS、FEMS、AEMS1、AEMS2）を検索、選出し、グループ化を行う。以降では、5.1.3.1(1)にて述べたシナリオに従って、コンダクタ及びノード間で処理を行う。AEMS1（地域）及びAEMS2（一般家庭）では、ノードとしてVPP管理システムのコンダクタの指示に従って動作する上で、上記指示を達成するためにコンダクタとして各々のサブグループにおけるノード（設備、HEMS）に対して指示を行う。

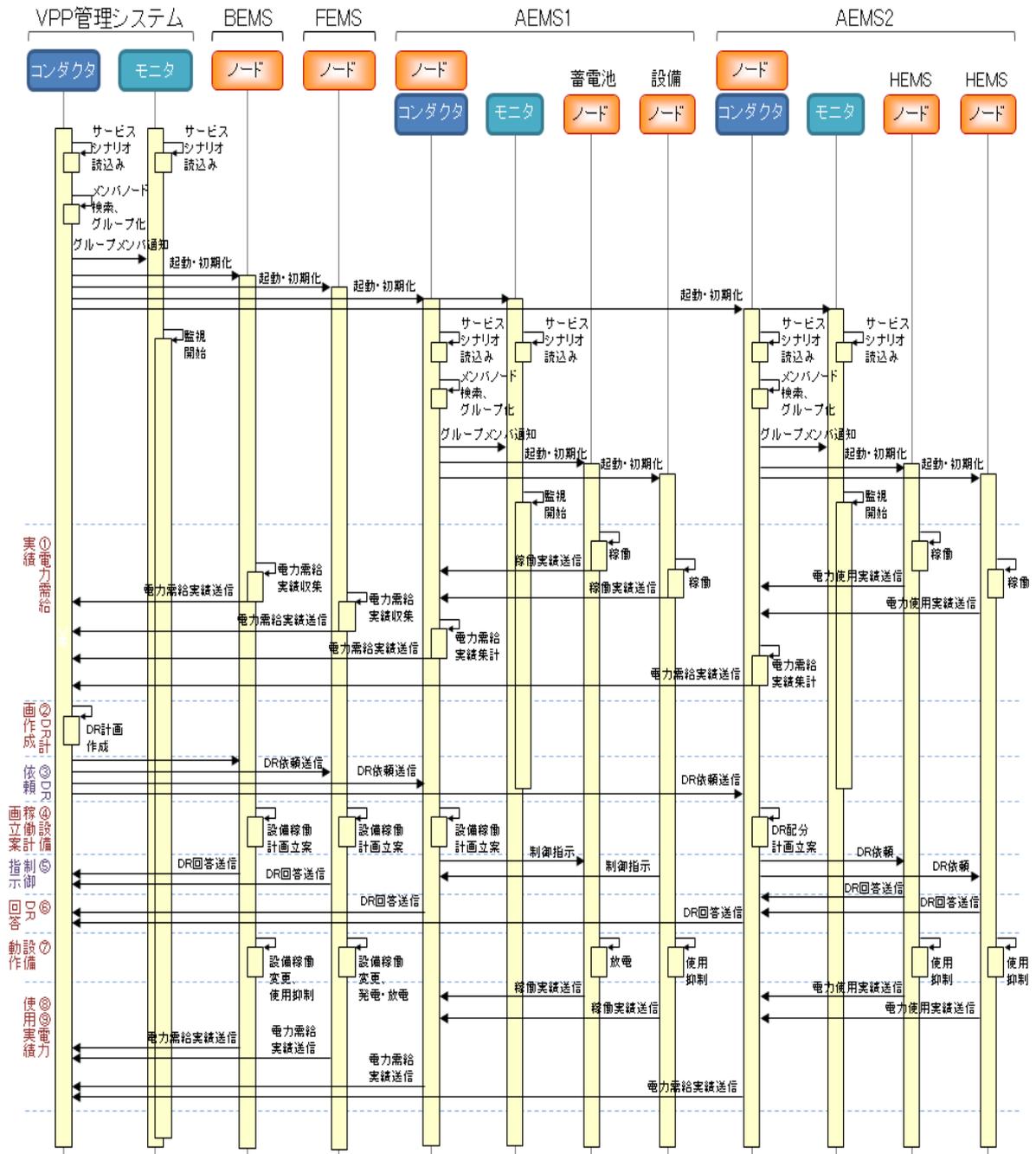


図 5-10 処理の流れ (予備力提供)

(2) 系統安定

図 5-7 に示したシステム構成において、5.1.2 にて述べた本研究の適用方法に従って、5.1.3.1(2)にて述べた VPP による系統安定のシナリオを実行する場合の処理の流れを図 5-11 に示す。これらの処理の流れは 4.6.1.2 にて述べたグループ動作の方式に従う。

VPP 管理システムのコンダクタ及びモニタが系統安定のためのサービスシナリオを読み

込む。コンダクタは上記サービスシナリオに基づきサービス実行に必要なノード（BEMS、FEMS、AEMS1、AEMS2）を検索、選出し、グループ化を行う。以降では、5.1.3.1(2)にて述べたシナリオに従って、コンダクタ及びノード間で処理を行う。AEMS1（地域）及びAEMS2（一般家庭）では、ノードとしてVPP管理システムのコンダクタの指示に従って動作する上で、上記指示を達成するためにコンダクタとして各々のサブグループにおけるノード（設備、HEMS）に対して指示を行う。

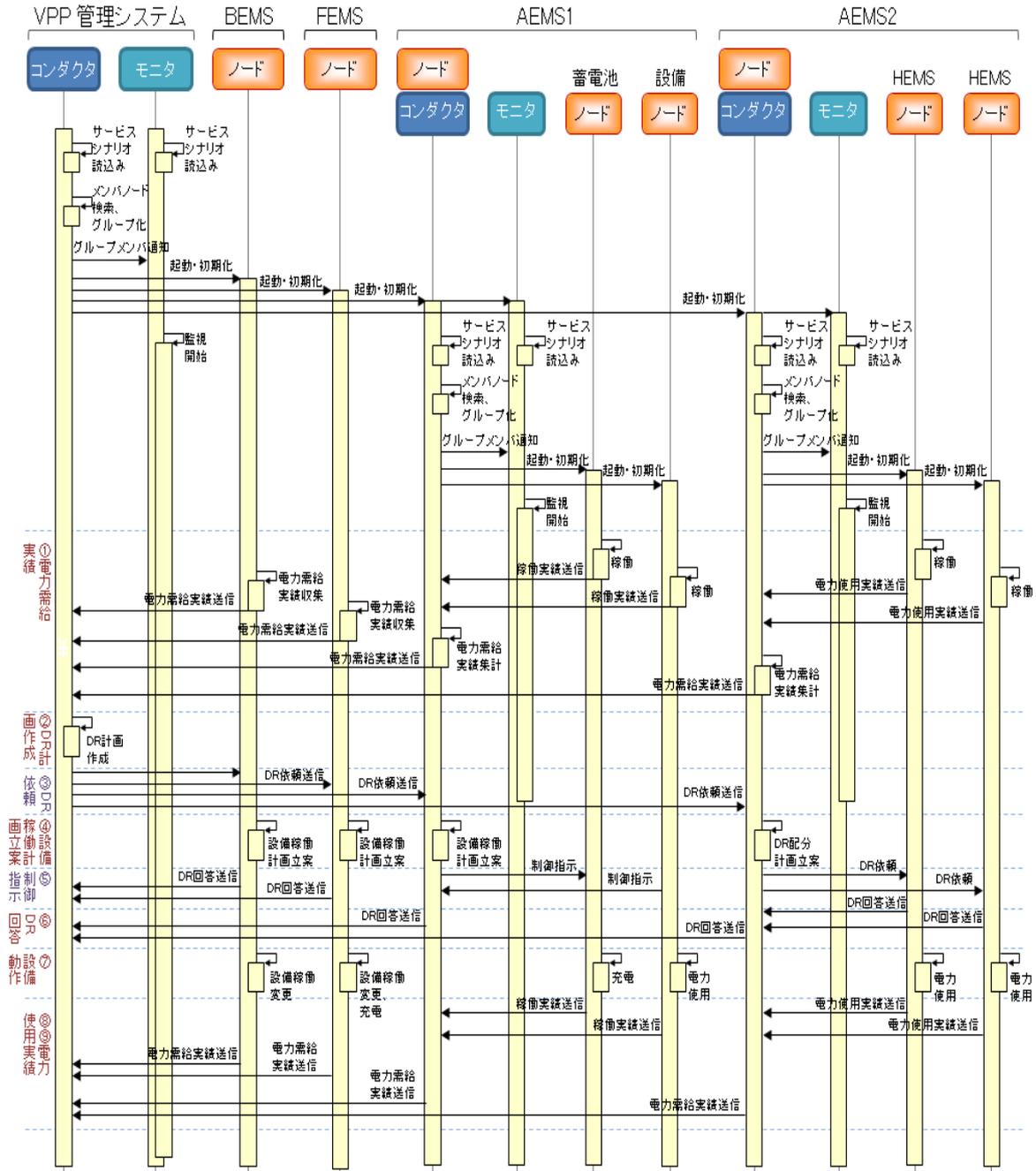


図 5-11 処理の流れ（系統安定）

(3) 状況に応じたグループ制御

5.1.3.2(1)及び(2)にて述べた、ユースケースシナリオ（予備力提供、系統安定）を実施するための処理の流れにおいて、状況変化等に関わらず、安定的に動作してサービス提供を継続するための、本研究にて提案するグループメンバ管理の実施例を表 5-3 にまとめる。シナリオ中の各ステップにおいて、VPP 管理システム、AEMS1（地域）、AEMS2（一般家庭）にて判定 KPI に基づいて判断、実施する例をまとめる。

表 5-3 グループメンバ管理の実施例

項目	グループメンバ管理の実施例		
	VPP 管理システム	AEMS1（地域）	AEMS2（一般家庭）
判定 KPI	電力需給量（予測値）	電力需給量（予測値）	電力需給量（予測値）
DR 依頼時 ：③④	各ノード（BEMS、FEMS、AEMS）に DR 依頼し、回答結果から、DR 計画（予備力調達もしくは電力消費）による電力需給量を達成できない場合、利用可能な他のノードを検索し、グループメンバに追加する。	各ノード（地域内の設備）に制御指示し、稼働実績から、VPP 管理システムより受領した DR 計画（予備力調達もしくは電力消費）による電力需給量を達成できない場合、利用可能な他のノード（設備）を検索し、グループメンバに追加する。	各ノード（HEMS）に DR 依頼し、回答結果から、VPP 管理システムより受領した DR 計画（予備力調達もしくは電力消費）による電力需給量を達成できない場合、利用可能な他のノード（HEMS）を検索し、グループメンバに追加する。
		新たにノード検索しても DR 計画による電力需給量達成不可の見込みであれば、VPP 管理システムに DR 実施不可と回答する。	新たにノード検索しても DR 計画による電力需給量達成不可の見込みであれば、VPP 管理システムに DR 実施不可と回答する。
DR 実行時 ：⑦⑧⑨	各ノードの電力需給実績を監視し、DR 計画による電力需給量に対して達成が困難なノードを検出した場合、該ノードをメンバから削除するとともに、利用可能な他のノードを検索し、グループメンバに追加する。	各ノード（設備）の稼働実績を監視し、DR 計画による電力需給量に対して達成が困難なノードを検出した場合、該ノードをメンバから削除するとともに、利用可能な他のノードを検索し、グループメンバに追加する。	各ノード（HEMS）の電力使用実績を監視し、DR 計画による電力需給量に対して達成が困難なノードを検出した場合、該ノードをメンバから削除するとともに、利用可能な他のノードを検索し、グループメンバに追加する。

項目	グループメンバ管理の実施例		
	VPP 管理システム	AEMS1 (地域)	AEMS2 (一般家庭)
		新たにノード検索しても DR 計画による電力需給量達成不可の見込みであれば、VPP 管理システムに DR 実施不可と報告する。	新たにノード検索しても DR 計画による電力需給量達成不可の見込みであれば、VPP 管理システムに DR 実施不可と報告する。

#### 5.1.4. 評価

5.1.3 までに述べたように、想定ユースケースとして VPP を挙げて、本研究にて提案する異種システム連携技術を適用検証した。

初期の VPP は主にビル、工場等の大口の需要家、もしくは系統につながる比較的規模の大きい設備を対象とし、限られた範囲内で構成していた。一方、今後は、一般家庭等の小口だが巨多の需要家に対する DR 実施、小規模の自家発電機や蓄電池も対象として範囲を拡張して構成していくものと考えられる。また個々の需要家及び設備は、VPP を運営する DR 事業者との契約内容や各需要家の保有する設備の稼働状況等に応じて、柔軟に VPP システムに加入及び離脱するものとなる。このため VPP を構成する需要家設備の管理及び制御はますます複雑化していく。

上述のような規模の異なる、巨多の需要家設備やシステムに対して 5.1.2.1 にて述べたように、本研究にて提案する動的グループ形成・制御を適用し、複数、多段のグループを構成し、同程度の規模のもの毎に範囲を区切って個々に管理することで、個々の需要家設備の変動に対する影響を局所化して、全体としても効率的な管理を可能とした。さらに扱う電力需給量の違いやメンバとしての入替の単位の違いに対応した。これにより規模や電力需給量が異なり、さらに一定して稼働するとは限らない、巨多の需要家設備を対象として構成する VPP システムの安定動作を可能とする。

また VPP を構成するために巨多の需要家設備を対象とし、全体で VPP として動作させるべく個々の設備を管理及び制御するために、各々の設備が準拠する多数の標準規格による異なるデータモデル及びインタフェースとの相互変換を行うことが不可欠であった。これらの処理を1つずつアプリケーションとして作り込むことは莫大なコストがかかる。

そこで 5.1.2.2 にて述べたように、本研究にて提案する異種ミドルウェア連携を適用し、VPP にて扱う共通データモデル及び共通インタフェースと、各々の設備が準拠する多数の標準規格による異なるデータモデル及びインタフェースとの相互変換の実現に伴うコストを削減させる。

以上のことから、社会インフラ（電力）におけるユースケースの1つとして挙げた VPP への適用検証を通して、本研究にて提案する異種システム連携技術の有用性を示すことができる。

## 5.2. 航空宇宙：惑星探査

### 5.2.1. 概要

#### 5.2.1.1. 無人探査機による惑星探査の概要

一例として、宇宙航空研究開発機構（JAXA）では惑星探査に関して、以下に述べることを提案している<sup>76)</sup>。

なぜ地球以外の惑星に生命存在に適した海洋・大気が存在しないのか、惑星の気候変動の究極的な原因は何なのか、惑星の磁気はどのようにして発生し消滅するのか、地殻やマントルを駆動するエンジンは惑星によってどのように変わるのか、等、現在の環境と惑星誕生以来の歴史を解明することが惑星科学の目標である。地球型惑星の進化過程を解明するためには、表面の物質分布や岩石・鉱物組成をリモートセンシングや「その場」観測、そしてサンプルリターンにより調べるとともに、惑星の内部構造を解明することが非常に重要である。惑星の内部構造を精密に知ることはリモートセンシングだけでは不可能であり、明確に意図した測器を装備しての着陸探査が必要である<sup>76)</sup>。

地球外の惑星は当分の間、多くの人類が出かけて活動する場とはならないと考えられ、当面、惑星探査の大部分は無人探査機を用いた科学探査となる。

#### 5.2.1.2. 無人探査機を用いた惑星探査の必要性

5.2.1.1にて述べた無人探査機を用いた惑星探査の主なミッション<sup>76)</sup>を表 5-4 にまとめる。

表 5-4 無人探査機を用いた惑星探査のミッション

#	ミッション	概要
(1)	「その場」観測	惑星上に探査機や観測機器等を投入、配置し、「その場」にある大気の観測や地表物質の分析、等を行う。
(2)	内部構造探査	惑星上の多数の地点に配置する地震計、電波源、熱流量計等の観測機器による多点ネットワークを構築し、惑星の3次元的構造を計測する。
(3)	サンプルリターン	惑星に着陸した探査機が地表より物質を採取して離陸し、地球へと持ち帰る。

表 5-4 のミッション(1)「その場」観測に関して、地球からもしくは惑星を周回する人工衛星からのリモートセンシングでは得られない、惑星上のより多種でより精密な観測データを長期間に渡り得ることができる。

表 5-4 のミッション(2)内部構造探査に関して、これまでの惑星探査ではあまり行われていないが、惑星の進化過程の解明につながり科学的価値が高い。ただし惑星の複数箇所の地中に観測機器を打ち込む必要があり、技術的難易度が高い。

表 5-4 のミッション(3) サンプルリターンに関して、地球外の惑星にある物質を地球への持ち帰ることは科学的な意義が極めて大きい。ただし地球までの航行も含めてミッション完了までに要する時間が膨大であり、有用な結果が得られるかどうかは最後まで分からない。

以上からミッション(2)(3)に関しては宇宙機技術自体の課題も多く含むことから、本研究の有用性を検証するためには、以降では特にミッション(1)について机上評価にて扱うものとする。

### 5.2.1.3. 無人探査機を用いた惑星探査の構成

5.2.1.2 にて述べた惑星探査のミッション(1)「その場」観測を実現するための構成を図 5-12 に示す。惑星探査の黎明期の頃には惑星上に着陸した 1 台の探査機（自律走行車、等）による「その場」観測が実施されていた。

一方、今後は図 5-12 に示すように、探査機の高機能化、複数投入により、惑星上に着陸した複数の探査機の連携動作により、より観測精度が高く、よりミッション実行の信頼性の高い「その場」観測を実施していくものと考えられる。ここでは複数の観測機器を搭載した複数の自律走行車、複数の観測機器を搭載した複数の自律飛翔体、複数の観測機器を搭載し、惑星上のある地点に設置される複数の観測設備、という異なる種類の複数の探査機で構成されるものとする。またこれらの探査機を管理、制御し、地球との通信を行う基地局も含まれるものとする。

上述のような構成において今後の惑星探査（「その場」観測）を実現するためには、以下の要件を満たす必要がある。

- 「その場」観測のために、惑星上の指定された地点において、指定された期間、指定された 1 つ以上の対象の観測を行い、観測データを基地局に伝送することを、タスクと定義する。
- タスクにて指定される観測に必要な観測機器を有し、指定された地点に到達可能な、1 つ以上の探査機の組合せによりタスクを実行する。
- 個々の探査機は電池残量（太陽光発電を想定）の低下や故障発生等により、稼働不能（タスク実行不可）となる場合がある。このため個々の探査機の状態を考慮して、タスクに用いる探査機の選出やタスク実行を行う必要がある。

また上述のような惑星探査（「その場」観測）を実施するためには、惑星上での探査機の長期運用が必要となる。探査機を出来るだけ長く使用し続けるためには、点検、搭載ソフトウェアの更新等の保守を実施していくことが不可欠となる。

宇宙探査は、世界の英知を結集する協力の場である一方、各国の技術力の競争の場でもある<sup>77)</sup>。ゆえに「その場」観測のためのタスク実行は、国を跨いで協調実施するべきであ

る。一方、各探査機の保守は、管理及び技術情報保護の観点で国毎に実施するべきである。

本節では、惑星上に複数の国（図 5-12 では A、B、C 国）が保有する探査機が存在すると  
して、それらの連携による「その場」観測のタスク実行、もしくは国毎の探査機の保守を  
実施するものとする。

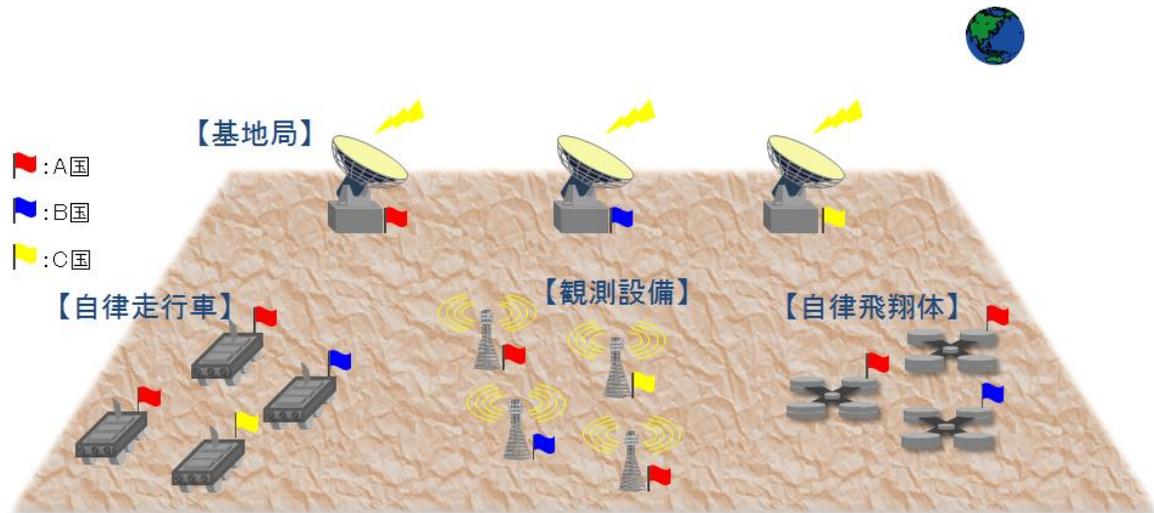


図 5-12 無人探査機を用いた惑星探査の構成

## 5.2.2. 本研究の適用方法

### 5.2.2.1. 動的グループ形成・制御の適用

5.2.1.3 にて述べたように、異なる種類の複数の無人探査機による惑星探査（「その場」観測）及び保守を実施するために、3.2 及び 4.6.1 にて述べた、本研究にて提案する動的グループ形成・制御の技術を適用する。なお実施する「その場」観測のタスク毎、保守のそれぞれに対してグループを構成するものとする。システム構成を図 5-13 に示す。

#### (1) 「その場」観測

図 5-13 に示すように、「その場」観測のタスク毎にタスク管理を稼働させ、タスク実行に必要な 1 つ以上の探査機への指示及び監視、各探査機の観測結果の収集及び集計とタスク実行結果の地球への送信を行う。

上記のシステム構成において、個々のタスク管理にそれぞれコンダクタ、モニタを割り当て、タスク実行に必要な個々の探査機にそれぞれノードを割り当て、これらをメンバとしたグループを国を跨いで形成する。

上述のように実行すべきタスク毎に必要なかつ利用可能な探査機をメンバとして選出しグループを構成することで、惑星上で実行すべきタスクの内容の変更や後からの追加に対して柔軟に対応できる。

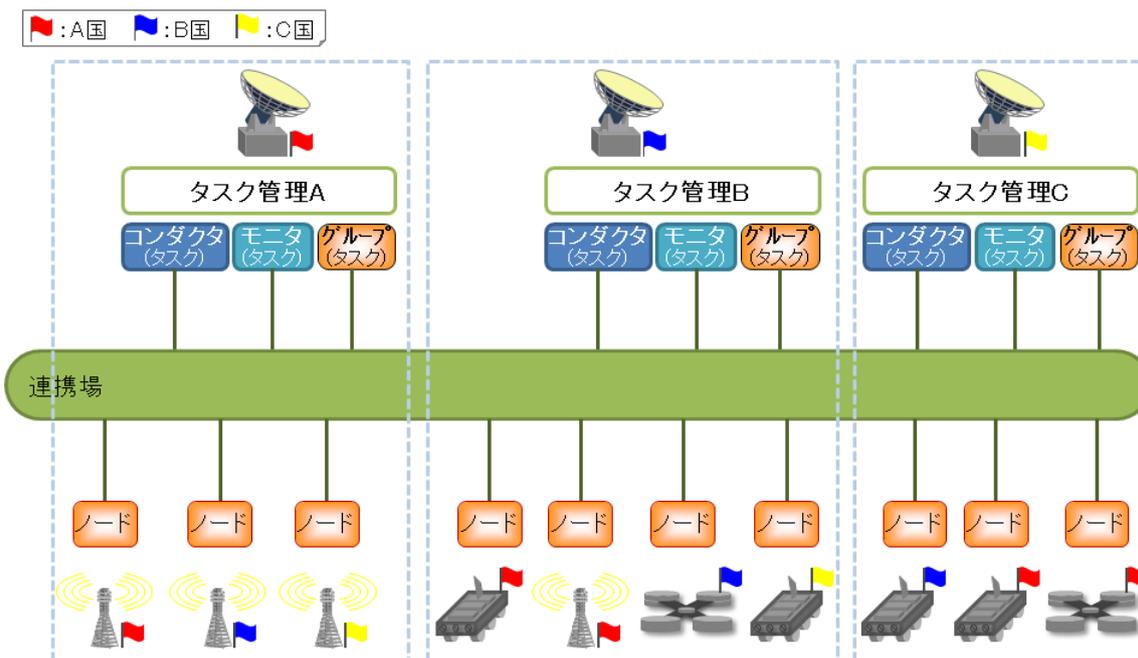


図 5-13 システム構成（グループ構成）：「その場」観測

(2) 保守

図 5-14 に示すように、国毎に保守を実施するために、保守管理を稼働させ、保守対象である各国の保有する 1 つ以上の探査機への指示及び監視、保守結果の集計、等を行う。

グループは国毎に形成する。上記のシステム構成において、個々の保守管理にそれぞれコンダクタ、モニタを割り当て、保守対象である個々の探査機にそれぞれノードを割り当て、これらをメンバとしたグループとする。

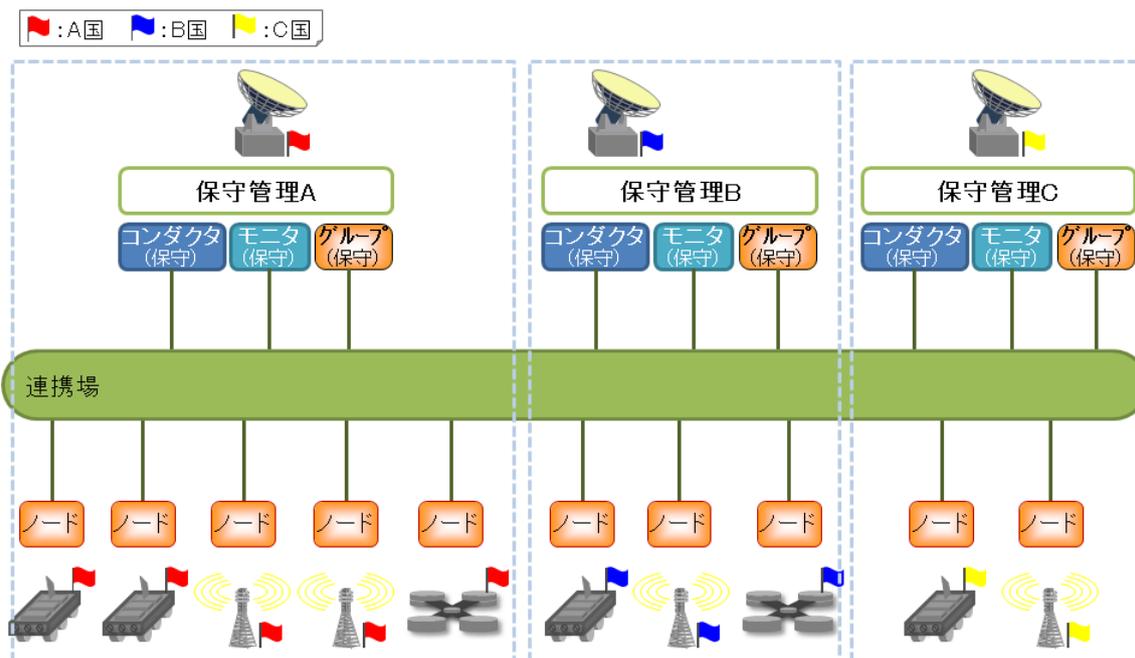


図 5-14 システム構成 (グループ構成) : 保守

### 5.2.2.2. 異種ミドルウェア連携の適用

5.2.1.3にて述べたように、異なる種類の複数の無人探査機による惑星探査（「その場」観測）及び保守を実施するために、3.3及び4.6.2にて述べた、本研究にて提案する異種ミドルウェア連携の技術を適用する。これにより各々異なるデータモデル及びインタフェースを有する場合でも、複数の探査機を1つのタスクにて扱うことを容易とする。

図 5-15 に示すように、タスク管理及び保守管理が稼働する基地局サーバ（国毎に設置）と、無線ネットワークを介して相互接続する各々の探査機で構成するものとする。

搭載するコンピュータのスペックから新たにソフトウェアを導入することが困難な探査機も含める可能性があるため、各々の探査機にマッピングするノードは、惑星上で最も高スペックのコンピュータを有すると考えられる基地局サーバに導入、稼働させるものとする。

各々の探査機から送受信されるデータは異種ミドルウェア連携を介して共通のデータモデルに変換して基地局サーバ上に格納し、各ノードが該データを参照、更新することで稼働する。

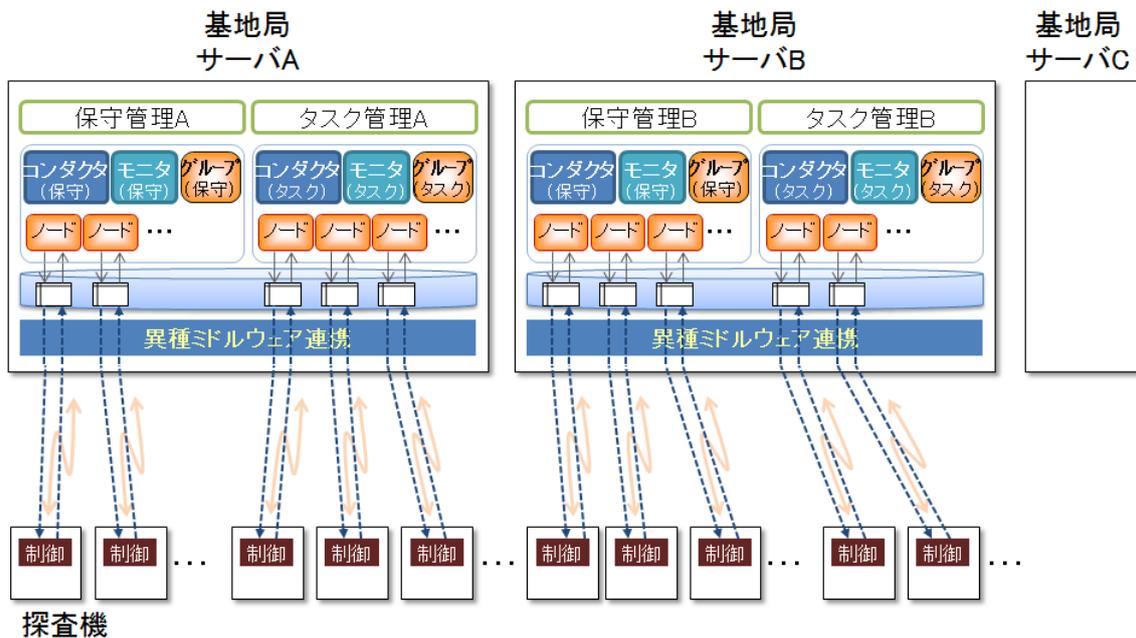


図 5-15 システム構成及び異種ミドルウェア連携の適用

### 5.2.3. ユースケース適用

5.2.1、5.2.2にて述べた惑星探査におけるミッション(1)「その場」観測及び探査機の保守に、本研究にて提案する異種システム連携技術を適用する場合のシナリオ及び動作内容を本節で述べる。

#### 5.2.3.1. シナリオ

##### (1) 「その場」観測

惑星探査における「その場」観測のシナリオの概要を図 5-16 に示し、シナリオにおける各ステップを以下に述べる。なお以下の①～⑧は図中の①～⑧に対応する。

- ① タスク実行計画作成  
タスク管理は、タスク実行に必要な1つ以上の探査機（自律走行車、自律飛翔体、観測設備）を選出し、それらを用いてのタスク実行計画を作成する。ここでタスク毎に個別に計画作成及び実行を行う。
- ② 移動指示  
タスク管理は、①にて作成したタスク実行計画に従って、移動可能な探査機（自律走行車、自律飛翔体）に対して、指定したタスク実行の地点までの移動指示を送信する。
- ③ 動作  
移動可能な探査機（自律走行車、自律飛翔体）は、②の移動指示を受け取り、指定されたタスク実行の地点まで自律的に移動する。

- ④ 移動完了報告 移動可能な探査機（自律走行車、自律飛翔体）は、③の動作の結果としてタスク実行の地点まで到達すると、タスク管理に対して移動完了報告を送信する。
- 一定時間経過しても移動完了報告が送信されない場合、タスク管理は該当探査機の移動が失敗したと判断して、該当探査機を除外してタスクを継続させるか、もしくは代替可能な他の探査機を選出して改めて移動指示を送信する。
- ⑤ センシング指示 タスク管理は、タスク実行の地点に存在する探査機（自律走行車、自律飛翔体、観測設備）に対して、指定する1つ以上の対象に対するセンシング（観測）実行の指示を送信する。
- ⑥ 動作 タスク実行の地点に存在する探査機（自律走行車、自律飛翔体、観測設備）は、⑤のセンシング指示を受け取り、指定された対象に対するセンシングを指定された期間で実行する。
- ⑦ センシング結果伝送 タスク実行の地点に存在する探査機（自律走行車、自律飛翔体、観測設備）は、⑥のセンシング実行の結果として得られたセンシングデータをタスク管理（基地局）に送信する。
- ⑧ センシング結果集計、送信 タスク管理は、⑦の個々の探査機から送信されたセンシングデータを集計して、まとめて地球に対して送信する。
- ⑤～⑧の処理は指定された期間、繰り返し実行する。

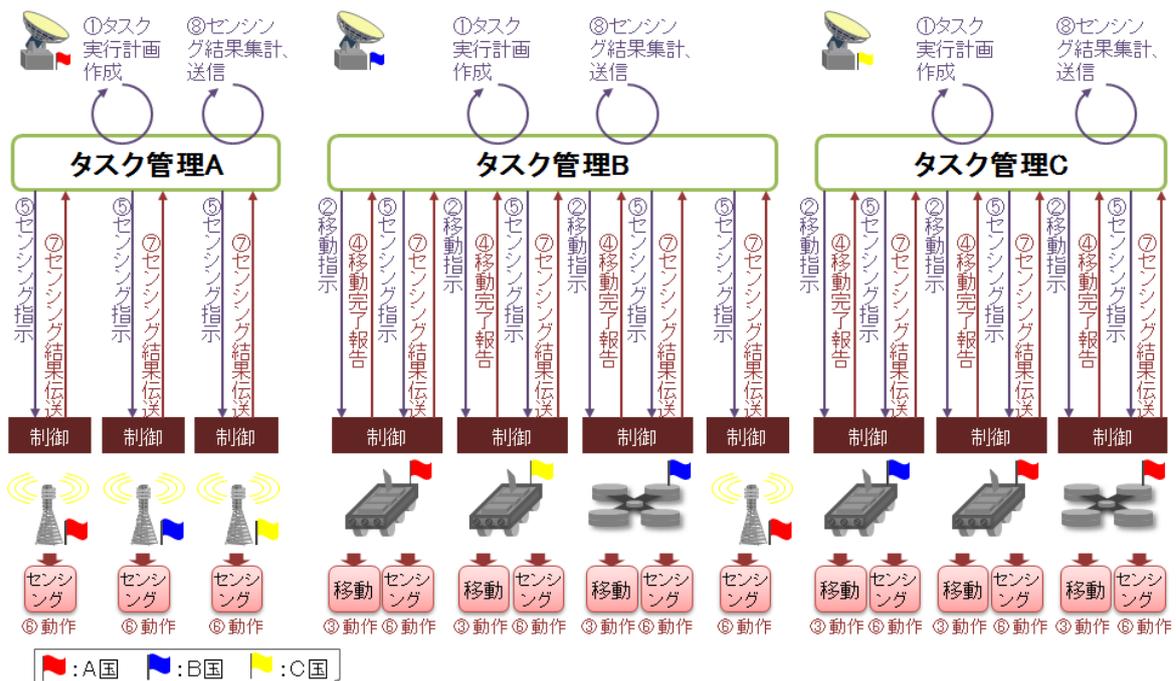


図 5-16 惑星探査のシナリオ（「その場」観測）

(2) 保守

惑星探査における保守のシナリオの概要を図 5-17 に示し、シナリオにおける各ステップを以下に述べる。なお以下の①～⑤は図中の①～⑤に対応する。

- ① 検査項目選定 国毎の保守管理は、自国の保有する探査機（自律走行車、自律飛行体、観測設備）を選出し、それらに対する保守計画を作成する。ここでは保守の一環として探査機の状態診断のために検査項目の選定を行う。
- ② ログ収集指示 国毎の保守管理は、自国の保有する探査機（自律走行車、自律飛行体、観測設備）に対して、①にて選定した検査項目に該当する自機のログデータ収集の指示を送信する。
- ③ 動作 探査機（自律走行車、自律飛行体、観測設備）は、②のログ収集指示を受け取り、指定された検査項目に該当するログデータの収集を実行する。
- ④ 収集ログ伝送 探査機（自律走行車、自律飛行体、観測設備）は、③の実行結果として得られたログデータを自国の保守管理（基地局）に送信する。

⑤ 診断

国毎の保守管理は、④の自国の個々の探査機から送信されたログデータを用いて、各探査機の状態診断を実行する。診断した結果はまとめて地球に対して送信する。

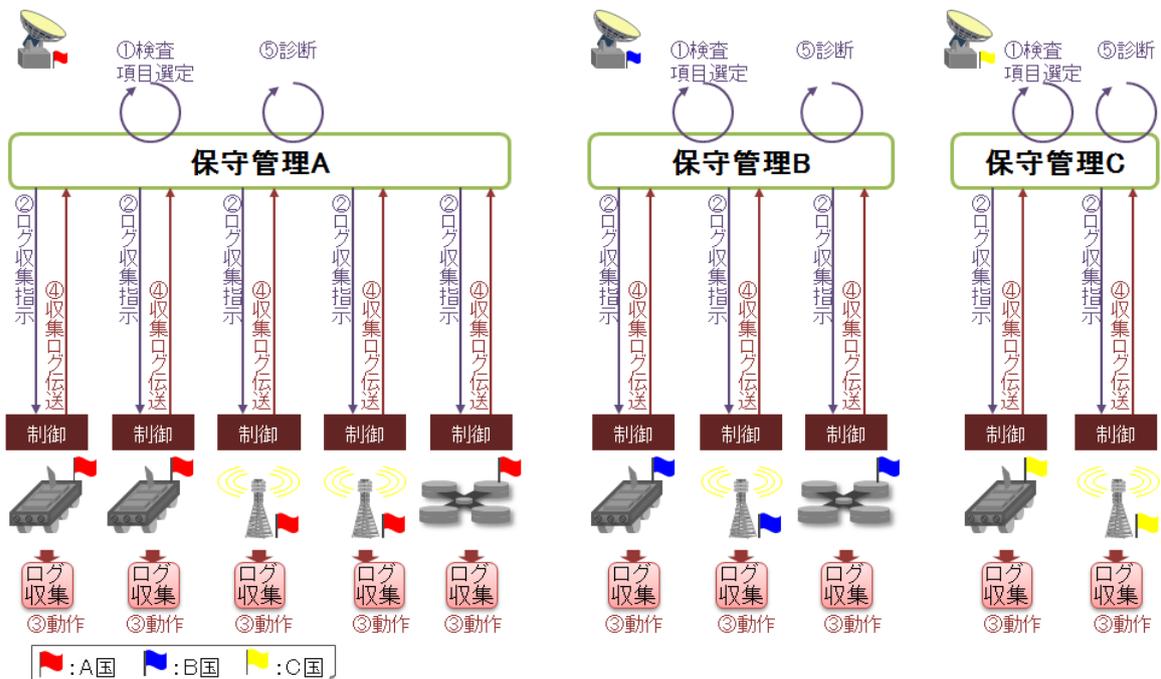


図 5-17 惑星探査のシナリオ（保守）

5.2.3.2. 動作

(1) 「その場」観測

図 5-15 に示したシステム構成において、5.2.2 にて述べた本研究の適用方法に従って、5.2.3.1(1)にて述べた惑星探査における「その場」観測のシナリオを実行する場合の処理の流れを図 5-18 に示す。これらの処理の流れは4.6.1.2にて述べたグループ動作の方式に従う。

タスク管理のコンダクタ及びモニタが、惑星上のある地点、ある期間での、ある観測対象に対する「その場」観測のためのサービスシナリオを読み込む。コンダクタは上記サービスシナリオに基づきサービス実行に必要なノード（自律走行車、自律飛行体、観測設備）を検索、選出し、グループ化を行う。以降では、5.2.3.1(1)にて述べたシナリオに従って、コンダクタ及びノード間で処理を行う。

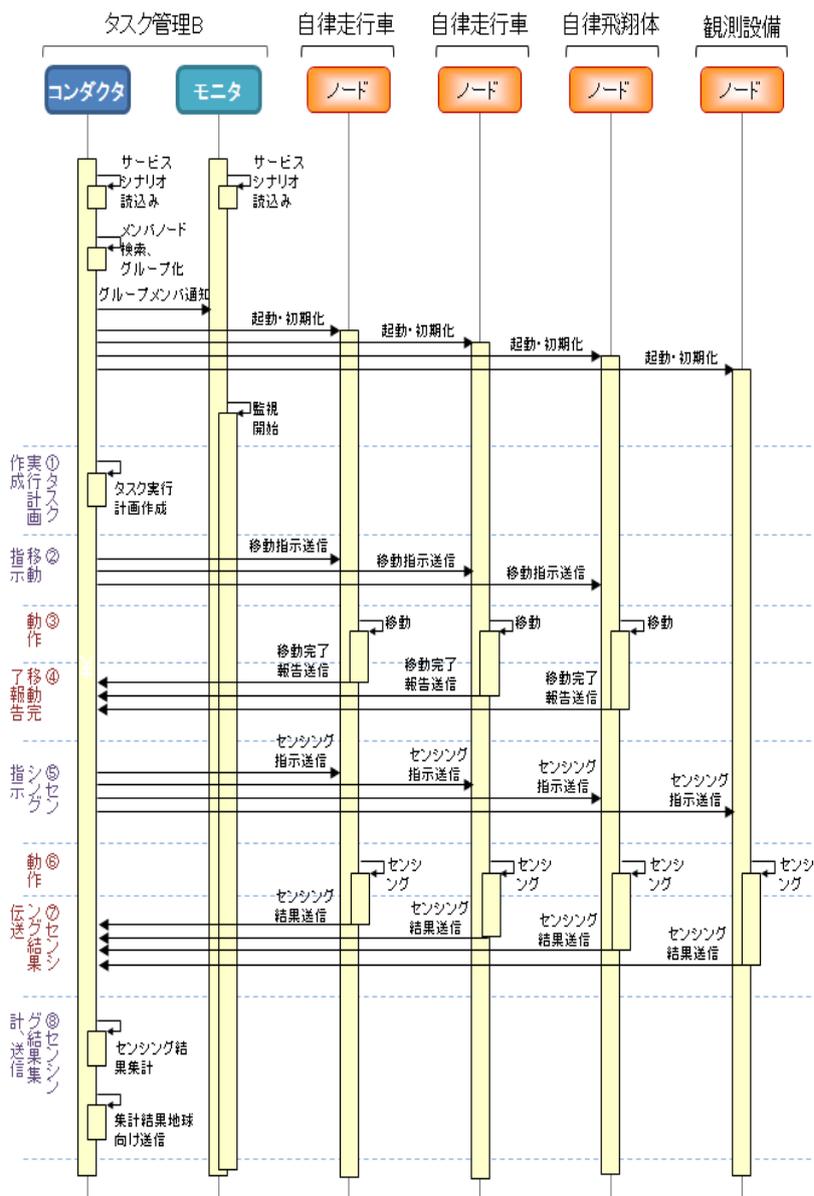


図 5-18 処理の流れ（「その場」観測）

(2) 保守

図 5-15 に示したシステム構成において、5.2.2 にて述べた本研究の適用方法に従って、5.2.3.1(2)にて述べた惑星探査における保守のシナリオを実行する場合の処理の流れを図 5-19 に示す。これらの処理の流れは 4.6.1.2 にて述べたグループ動作の方式に従う。

国毎の保守管理のコンダクタ及びモニタが、惑星上で自国が保有する探査機の保守のためのサービスシナリオを読み込む。コンダクタは上記サービスシナリオに基づきサービス（保守）実行に必要なノード（自律走行車、自律飛行体、観測設備）を検索、選出し、グループ化を行う。以降では、5.2.3.1(2)にて述べたシナリオに従って、コンダクタ及びノ

ード間で処理を行う。

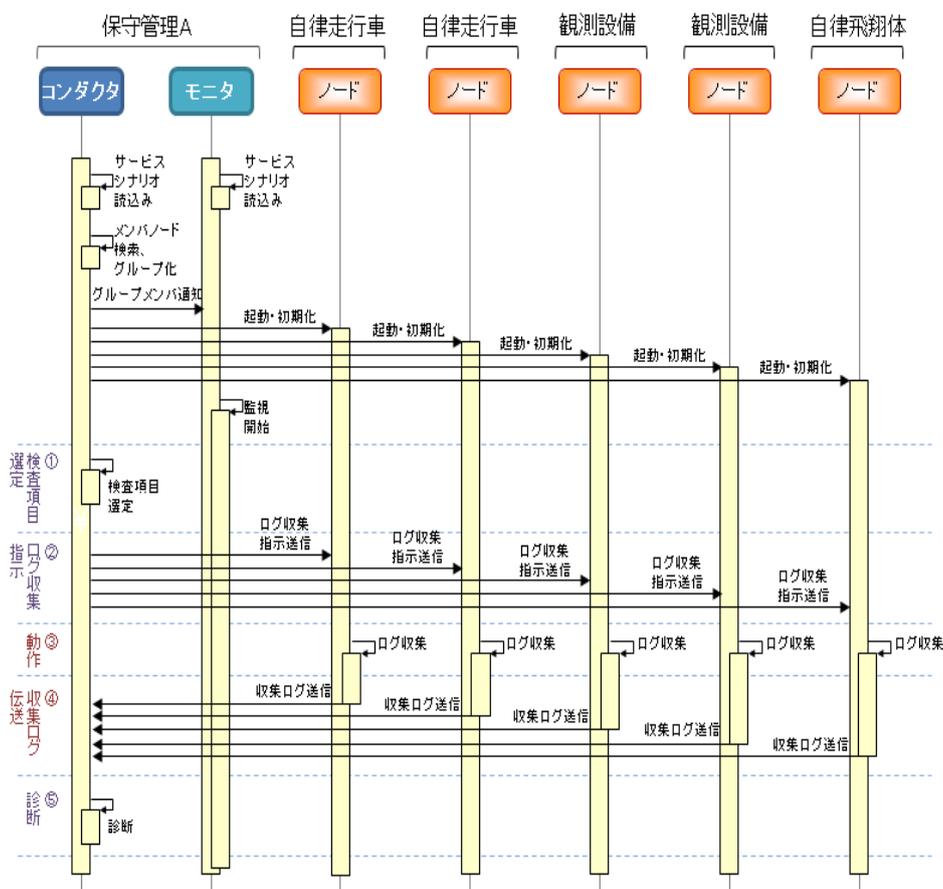


図 5-19 処理の流れ (保守)

### (3) 状況に応じたグループ制御

5.2.3.1(1)にて述べた、ユースケースシナリオを実施するための処理の流れにおいて、状況変化等に関わらず、安定的に動作してサービス提供を継続するための、本研究にて提案するグループメンバ管理の実施例を表 5-5 にまとめる。シナリオ中の各ステップにおいて、タスク管理にて判定 KPI に基づいて判断、実施する例をまとめる。

表 5-5 グループメンバ管理の実施例

項目	グループメンバ管理の実施例
	タスク管理
判定 KPI	タスク達成可否

項目	グループメンバ管理の実施例
	タスク管理
タスク実行 計画作成時 ：①	惑星上にある探査機の中から、タスク実行に必要な観測機器を有し、観測地点への到達が可能であるものを検索し、グループメンバとして追加する。
移動時 ：③	自律走行車、自律飛翔体の位置、観測地点への到達状況を監視し、タスク実行計画による観測地点への達成が困難なノード（自律走行車、自律飛翔体）を検出した場合、該ノードをメンバから削除するとともに、利用可能な他のノードを検索し、グループメンバに追加する。
センシング 時 ：⑥	センシング実行中に、センシング実行不能となるノード（自律走行車、自律飛翔体、観測装置）を検出した場合、該ノードをメンバから削除するとともに、利用可能な他のノードを検索し、グループメンバに追加する。追加したノードが観測地点にはいない場合、まず観測地点まで移動させて観測開始させる。

## 5.2.4. 換装・新旧入替

### 5.2.4.1. 概要と要件

5.2.1にて述べたような惑星探査は長期間に渡り継続的に実施される<sup>78)79)</sup>。その間に探査機の経年劣化・老朽化が生じるため、探査機の換装が必要となる。そのため探査機の惑星への順次投入による老朽化した旧機から新機への入替を実施していくことが想定される。また順次投入していく複数の探査機（種別の異なるものも含む）を組合せていき、惑星探査システムを拡張、長期維持していくことも想定される。探査機の換装・新旧入替のイメージを図 5-20 に示す。

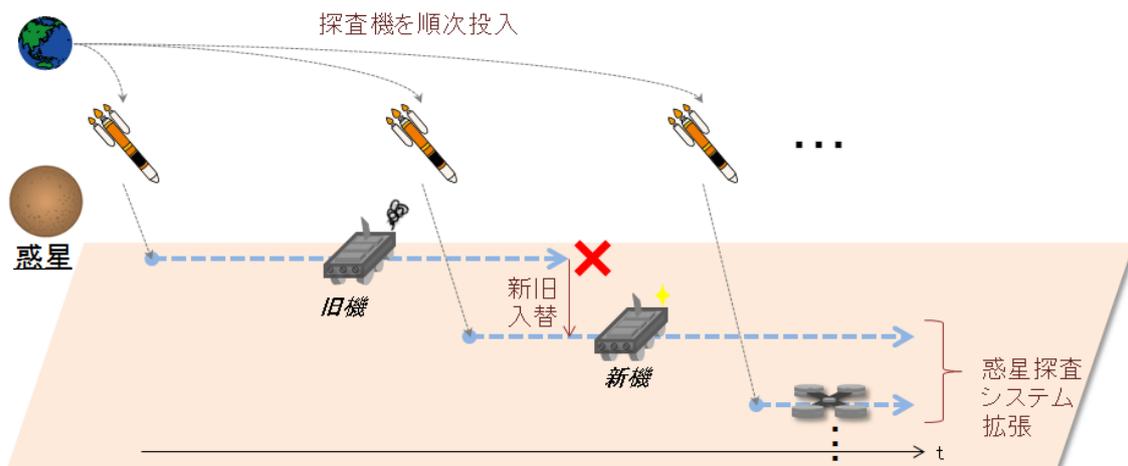


図 5-20 探査機の換装・新旧入替のイメージ

上述の探査機の換装・新旧入替を実施する際の要件を以下に挙げる。

- ① 惑星投入時期の違いに起因する探査機の新旧の仕様の違いを、惑星探査システムに意識させない。
- ② 探査機の新旧入替の際に、無停止で惑星探査システムのミッション・タスクを継続する。
- ③ 惑星探査システムを段階的に拡張する（段階的に機能・サービスを拡充）。

#### 5.2.4.2. 本研究の適用方法

5.2.4.1にて述べた探査機の換装・新旧入替を実施する際の3つの要件に対して、本研究の提案技術を用いることで、①アプリケーションの変更不要で、新旧仕様の差異を吸収する、②探査機の新旧並行稼働による検証とサービス無停止での切替を行う、③新機を追加、活用してサービスを拡張していく、ことを実現する。以降で詳細を述べる。

##### (1) 新旧仕様の差異吸収

3.3及び4.6.2にて述べた、本研究にて提案する異種ミドルウェア連携の技術を適用する。概要を図5-21に示す。

新旧の探査機それぞれに対してアダプタを用意し、ノードとのマッピングを行う。アダプタでは、データモデル、インタフェースに関する探査機側の個別仕様と、ノード側の共通仕様（SD0準拠）との対応付けを行う。

上記によりアプリケーションが参照するのはノード（SD0）の共通仕様によるデータモデル、インタフェースであるため、探査機の新旧の仕様の差異をアプリケーションは意識する必要はない。

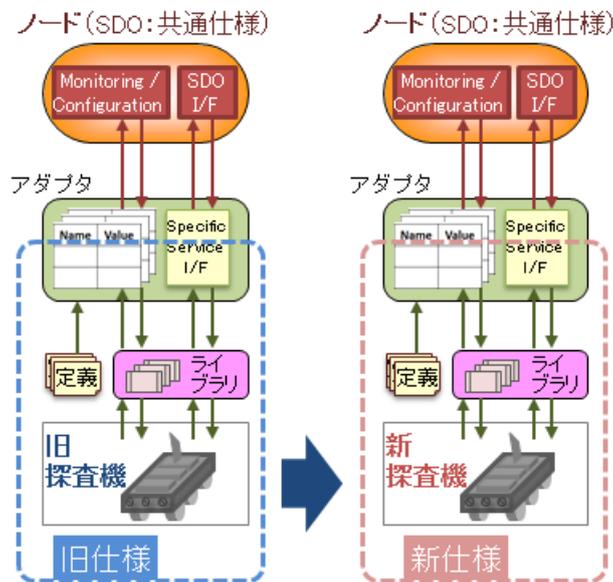


図 5-21 新旧仕様の差異吸収

(2) 新旧並行稼働・切替

3.2 及び 4.6.1 にて述べた、本研究にて提案する動的グループ形成・制御の技術を適用する。概要を図 5-22 に示す。

グループ内に新旧の探査機にマッピングされるノードでサブグループを形成する。サブグループ内で一定期間、新旧のノードを並行稼働した後に、旧探査機のノードを停止させる。並行稼働期間中には新旧のノードの出力比較等により新探査機のテストを実施する。また並行稼働期間中はコンダクタが新旧のノードからの出力を選択することで、グループによるサービス実行が支障なく行えるようにする。

上記により新旧並行稼働による新探査機の検証とサービス無停止での探査機の新旧切替を実施することができる。

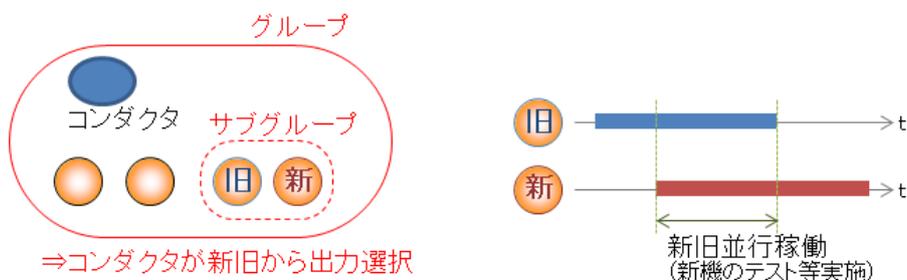


図 5-22 新旧並行稼働・切替

(3) サービス拡張

3.2 及び 4.6.1 にて述べた、本研究にて提案する動的グループ形成・制御の技術を適用する。概要を図 5-23 に示す。

新探査機のノードを加えた新グループを形成する。これまでのグループによるサービスだけでなく、新グループを用いての新サービス（タスク拡張）を実施する。



図 5-23 サービス拡張

### 5.2.5. 評価

5.2.4 までに述べたように、想定ユースケースとして惑星探査における「その場」観測、及び探査機長期維持のための保守を挙げて、本研究にて提案する異種システム連携技術を適用検証した。

惑星探査の黎明期の頃には惑星上に着陸した 1 台の探査機（自律走行車、等）による「その場」観測が実施されていた。一方、今後は探査機の高機能化、複数投入により、惑星上に着陸した複数の探査機（自律走行車、自律飛翔体、観測設備、等）の連携動作により、より複雑でより観測精度が高くなるが、よりタスク実行の信頼性の高い「その場」観測を実施していくものと考えられる。個々の探査機は電池残量（太陽光発電を想定）の低下や故障発生等により、稼働不能（タスク実行不可）となる場合がある。このため複雑化するタスクを確実に達成するためには、個々の探査機の状態も考慮してのより高度なタスク実行管理を行う必要がある。一方、管理及び技術情報保護の観点で、各探査機の保守は国毎に実施するべきである。

上述のような「その場」観測のための異なる複数の探査機を用いるタスク実行及び国毎の探査機の保守に対して 5.2.2.1 にて述べたように、本研究にて提案する動的グループ形成・制御を適用し、「その場」観測のためのタスク実行の単位や保守の管理体系に基づくグループを構成し、タスク実行に必要な能力を有する探査機をメンバとして、各探査機の状態に応じてメンバの入替等を実施することでタスク達成までの継続的な実行に対応した。また実行すべきタスクの内容の変更や後からの追加に対して柔軟に対応することを可能とする。さらに 5.2.4.2 にて述べたように、新旧の探査機にマッピングしたノード及びサブグループの管理により、サービス無停止での新旧入替、段階的なサービス拡張を可能とする。

また複数の異なる種別の探査機を対象とし、全体でタスク実行のために動作させるべく個々の探査機を管理及び制御するために、各々の探査機が有する異なるデータモデル及びインタフェースとの相互変換を行うことが不可欠であった。これらの処理を1つずつアプリケーションとして作り込むことも莫大なコストがかかった。

そこで5.2.2.2にて述べたように、本研究にて提案する異種ミドルウェア連携を適用し、タスク管理にて扱う共通データモデル及び共通インタフェースと、各々の探査機が有する異なるデータモデル及びインタフェースとの相互変換の実現に伴うコストを削減させる。さらに5.2.4.2にて述べたように、長期に渡る運用の中で、探査機の仕様の変化を吸収し、新旧移行に伴うコストを削減させる。

以上のことから、航空宇宙におけるユースケースの1つとして挙げた惑星探査における「その場」観測及び探査機の保守への適用検証を通して、本研究にて提案する異種システム連携技術の有用性を示すことができる。

## 6. 考察

本章では、第5章にて述べた実問題における想定ユースケース例及び評価結果を踏まえて、第3章及び第4章にて述べた、本研究にて提案する異種システム連携技術を、社会インフラ分野、航空宇宙分野に適用することにより期待できる効果について述べる。また第5章にて述べた以外のその他の実問題や、その他の分野への、本研究にて提案する異種システム連携技術の応用可能性についても述べる。

### 6.1. 本研究の提案技術の適用により期待できる効果

#### 6.1.1. 社会インフラ分野における効果

本研究にて提案する異種システム連携技術を、社会インフラ分野に適用することにより、以下に述べるような効果が期待できる。

例えば5.1にて述べた電力におけるVirtual Power Plantへの適用のように、従来からの既存設備だけでなく、様々な需要家やサービス事業者の設備やシステムの柔軟な連携による、新しいサービスの提供や後からのサービスの追加・変更が容易となる。ここで異種のシステムにおける異なる標準規格・プロトコルに準拠する様々な設備やアプリケーションを、状況に応じて柔軟に組合せ、入替しながら、それらを用いることによるサービス提供を継続的に維持するという、多種多様性及び信頼性を保証する。また社会インフラにおける設備やアプリケーションプログラムは、ある範囲毎に段階的に新旧入替が実施されていく中で、上記のように様々な変化に追従しながらの、サービス提供の長期維持に寄与できるという、拡張性も保証できる。

#### 6.1.2. 航空宇宙分野における効果

本研究にて提案する異種システム連携技術を、航空宇宙分野に適用することにより、以下に述べるような効果が期待できる。

例えば5.2にて述べた無人探査機を用いた惑星探査への適用のように、惑星上に着陸した複数の探査機の連携動作による、より複雑でより観測精度が高くなるが、より信頼性の高いミッション遂行を容易にすると考える。ここで惑星上の環境変化や電池残量（太陽光発電を想定）の低下や故障発生等による1つ以上の探査機の稼働不能化、等の状況変化に柔軟に対応してミッション遂行継続を容易化するはずである。

また惑星探査システムを構築する上で、地球から惑星への輸送で1回あたり宇宙往還機に搭載できる無人探査機の数に限りがあり、輸送に時間が要する。このため惑星探査システムを構成する無人探査機は新旧含めて異なる種別、能力のものが混在している可能性がある。このような場合でも構成する個々の無人探査機の性能、状況に応じて、確実度をもってミッションを遂行するための惑星探査システムを構成、運用することができる。さらに上記の惑星探査システムにて、ミッションやタスクを無停止のままで探査機の新旧入替

を実施していくこともできる。

## 6.2. 本研究の提案技術の応用可能性

### 6.2.1. 社会インフラ分野における応用可能性

本研究にて提案する異種システム連携技術を応用すれば、災害発生等が原因で社会インフラシステムの一部が動作不能となった場合でも、インフラ事業者等を跨いで残存した利用可能な設備やシステムを連携させることで、インフラサービスの安定的な提供もしくは早期復旧の実現に貢献できると考える。

また本研究にて提案する技術と、IoT、ビッグデータ分析等の技術を組み合わせることにより、社会全体における様々な問題を解決するための、状況に応じた多種多様なデータの収集と、データ分析結果に基づく現実世界（サービス、制御システム）への反映を行う社会システムの実現に応用できると考える。

さらに2016年度から始まっている「第5期科学技術基本計画」<sup>81)</sup>にて掲げる、「必要なもの・サービスを、必要な人に、必要な時に、必要なだけ提供し、社会の様々なきめ細かに対応でき、あらゆる人が質の高いサービスを受けられ、年齢、性別、地域、言語といった様々な違いを乗り越え、生き生きと快適に暮らすことのできる社会」である「超スマート社会（Society 5.0）」<sup>82)</sup>を実現する上での下支えとなる技術として、本研究にて提案する異種システム連携技術は応用可能であると考えられる。

### 6.2.2. 航空宇宙分野における応用可能性

宇宙空間もしくは地球外の他の惑星上に、探査・観測、等の大規模なシステムを構築する場合、5.2.4、6.1.2でも述べたように、地球から惑星への輸送で1回あたり宇宙往還機に搭載できる機材・設備の数に限りがあり、輸送にも時間が要するため、上記のようなシステムは段階的な構築を行うこととなる。5.2.4でも述べたように、システムが全て完成するまでに長い時間を要するため、完成まで待っていると初期の機材・設備の劣化が起り得る。このため構築途中であってもシステムは稼働し、その時点で動作可能な機能・サービスを実行させておくことが望ましい。稼働中の機材・設備の劣化と新しい機材・設備の段階的な投入により、システムの構成要素の入替・拡張が断続的に実施されていくと考えられる。

上記にて、本研究にて提案する異種システム連携技術を応用すれば、5.2.4にて述べた惑星探査システム以外にも、宇宙環境で利用される様々な宇宙システムを対象として、長期間に及ぶ段階的なシステム構築において、その時点で実行可能な機能・サービスを提供するとともに、長期視点で段階的に機能・サービスを拡張していくことも可能であると考えられる。

### 6.2.3. 他分野への応用可能性

6.2.1、6.2.2にて述べた社会インフラ分野、航空宇宙分野以外の分野への、本研究にて提案する異種システム連携技術の応用を検討する。ここでは一例として産業分野への応用を検討する。以下に応用可能であると考えられる2つの想定ユースケース例を挙げる。

#### (1) 生産プロセスのコンカレント化

図 6-1 に生産プロセスのコンカレント化の概要を示す。従来は1つの主体内で商品企画から設計、生産準備、生産までの工程を一貫して行っていたのに対して、生産工程を他の主体へ分散、同時実行を行う。これにより生産プロセスにおける手戻りの削減、開発期間の短縮が期待できる。またサプライチェーンに関わる複数主体（部品生産～組立て）を跨いでのプロセス全体の効率化にもつながる。

ここでは本研究の提案技術を応用して、各主体の生産ライン・設備の稼働状態及び稼働計画や部品在庫状態、等を監視し、生産計画に対して、主体を跨いでの生産ライン・設備の割り当てを行う。また生産工程間もしくは主体間での密なプロセス状況の共有を行い、状況に応じた生産ライン・設備の再割り当てや再計画を行う。

なお制約または前提として、製品構造やプロセス間・主体間の相関の捉え直し、業務工程の再設計が必要となる。

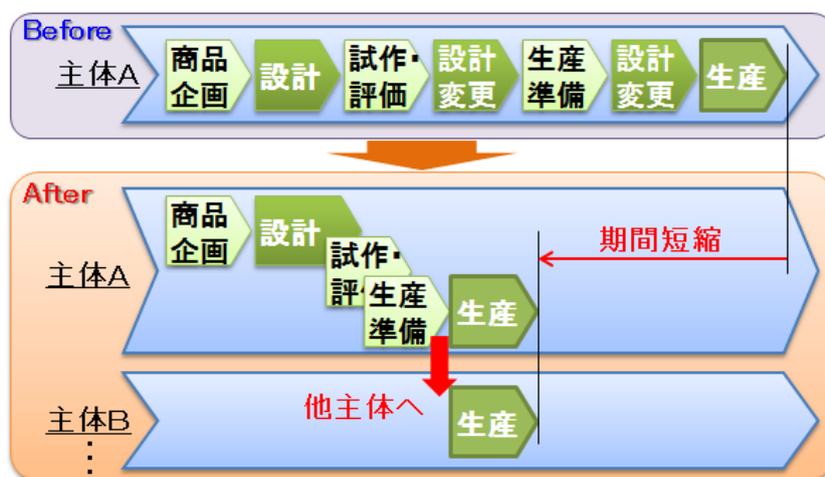


図 6-1 生産プロセスのコンカレント化

#### (2) プロアクティブな設備保守

図 6-2 にプロアクティブな設備保守の概要を示す。特に製造業における設備保全（予防保全）の例として、設備・システムの状態に応じて、必要な時に必要なレベルでのみ保守を実施することでの作業の効率化につながることを期待できる。特に障害発生に伴う製造工程におけるシステムダウンタイムの軽減のためには、予防保全が推奨されている。また

保守員のスキル・経験に関わらず、一定レベル以上の保守実績を保証することが期待できる。

ここでは本研究の提案技術を応用して、主体を跨いで設備からのデータを集約・加工・分析し、システム全体の状態や影響範囲を把握する。また主体を跨いで関連する範囲内での予兆・異常事象の検知を行う。検出した予兆・異常の程度に応じての保守員割当てや作業指示を行う。ここでは多主体で人間系も含めた保守システムを構築することとなる。また保守員間で現場知識・経験の共有、有効利用化にもつなげる。

なお制約または前提として、製品仕様変更、ライン再構成が頻度高く発生すると、システム稼動不安定化による障害が発生し得る。

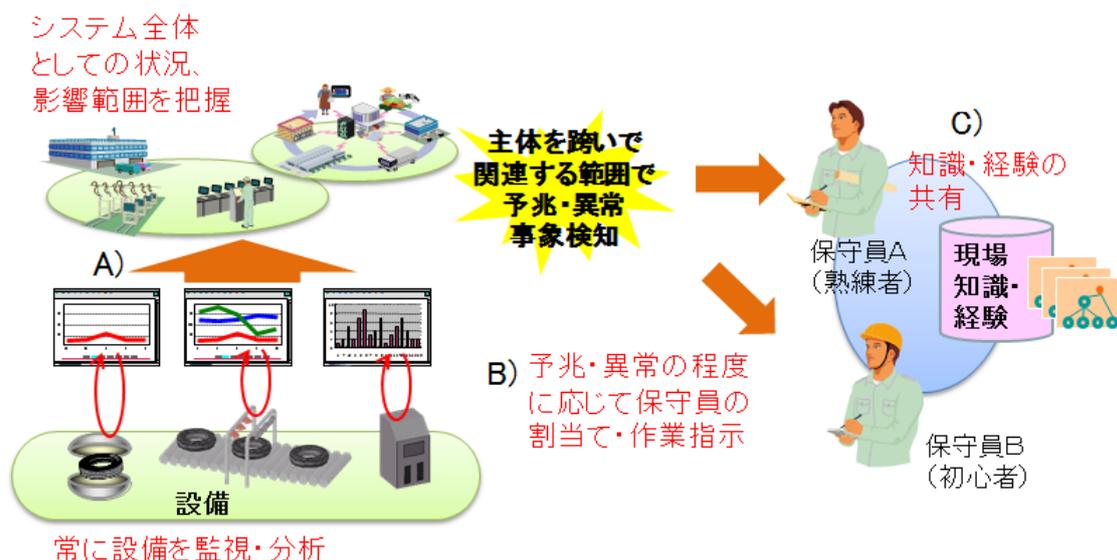


図 6-2 プロアクティブな設備保守

## 7. 結論

本章では、第1章から第6章までに述べてきた本研究による提案内容をまとめるとともに、本研究をさらに深度化させていく上での今後の課題について述べる。

### 7.1. 研究のまとめ

本研究では、社会インフラシステムの信頼性を保証し、どのような状況においてもサービス提供を維持させるために、目的、導入時期等の異なる異種のシステム同士を、各システム的设计当初の想定・前提に関わらず、状況に応じて、柔軟に連携させるためのシステムアーキテクチャ及び基盤技術としての異種システム連携技術について提案した。

第2章では、本研究に対する従来研究及び関連研究として特に、これからの社会インフラを支える主要なシステム技術の1つである、System of Systems (SoS)やCyber Physical System (CPS)の概要、潮流について述べた。特にSoSの性質として、Connectivity (接続性)、Belongings (所属)、Autonomy (自律性)、Diversity (多様性)、Emergence (創発)の5つが挙げられていた。

本研究による異種システム連携はSoSの1つの形態であると言える。上記のSoSの5つの性質のうち、Connectivity (接続性)、Belongings (所属)、Autonomy (自律性)は対象とするシステムやサービスの前提となり得るとした。特にDiversity (多様性)の観点で、社会インフラに関して異なる多数の事業主体を跨いでの、状況に応じた様々な目的を達成するための、多主体多目的協調型のシステムとして本研究による異種システム連携を捉えた。またEmergence (創発)の観点で、非常時対応も含めて状況に関わらず社会インフラとしてのサービスを継続するために、社会インフラシステムを構成する制御系及び情報系の柔軟な連携による、制御/情報統合型のシステムとしても本研究による異種システム連携を捉えた。一方、CPSに関しては、CPSは、従来からの企業等の特定の範囲内だけでの制御系や情報系からシステムの裾野を大きく広げる概念及び技術であり、本研究による異種システム連携を具体的なシステムにて実現していく際に、実現形態の1つとなり得ると考えた。

なお従来研究・関連研究では、Connectivity (接続性)の観点では、例えば自律分散プロトコルADS-netにより、データフィールドというデータ共有の場を形成し、システムを構成する計算機間でデータの受け渡しを可能としたが、制御系と情報系のように異なるシステムにおいては、既に実績を持って導入されているプロトコル、基盤・ミドルウェアや運用形態等が異なるだけでなく、データ構造・粒度、処理周期・タイミング等も異なり、システム間で連携するためにはこれらの差異の解決が必要であった。Autonomy (自律性)の観点では、多様な機器の連携によるサービスを提供する上で、例えば超分散オブジェクト (Super Distributed Objects : SDO) モデルのような共通モデルを用いて、個々の機器の仕様の違いを意識せず一様に扱い、アプリケーションを共通的に作成可能であったが、

対象とするシステムの裾野が拡大する程に、多種多様で大量の機器を一様に扱うことは、個々の状態の把握も必要となり、管理の負荷が増大してしまった。Diversity（多様性）の観点では、スマートシティ実証等で、主に同じ種別で目的も類似する複数の主体間の連携が達成できていたが、異なる種別で目的も異なる複数の主体間の連携となると、主体間の違いを調整した上で全体的な効率化が必要であった。Emergence（創発）の観点では、例えば SoS の要素となるシステムの追加・削除における相互運用性確保のため、要素となるシステムの突発的な振舞いを事前把握するための振舞いモデルの検討が行われていたが、実システムへの適用検討はこれからであった。

第3章では、上記の第2章にて述べた従来研究及び関連研究では対応できていない問題を解決するために本研究にて提案する、異種システム連携技術について述べた。

上記の SoS の5つの性質のうちの Diversity（多様性）、Emergence（創発）の観点から、複数の異なる種別のシステムの連携であっても、全体効率化や様々な新たなサービス提供を実現するために、サービスと実際のインフラシステム、さらにそれらを相互接続し仲介するシステムビュー、の3つの階層から成るアーキテクチャを有する異種システム連携コンセプトを提案した。

Autonomy（自律性）の観点では、上述の異種システム連携コンセプトを具体化する上で、システムの裾野が拡大され、多種多様で大量の機器を、各機器の状態やサービスに対する要求内容に応じて柔軟に扱うための、動的グループ形成・制御の技術を提案した。

Connectivity（接続性）の観点では、上述の異種システム連携コンセプトを具体化する上で、実際のシステム間で仕様や動作状況の違いを吸収するために、個々のシステムの物理構成やデータの配置等に基づき既に稼働している異なるミドルウェアの間での連携を行う、異種ミドルウェア連携の技術を提案した。

また本研究の提案技術を導入するシステムの運用、動作保証、既存システム保護等の、その他の必要な要素技術についても述べた。

第4章では、第3章にて述べた、本研究にて提案する異種システム連携技術を実装し、実際のシステムを構築するための方式について述べた。

ここでは本研究に基づくシステムを構成する上で想定するソフトウェア構成及びハードウェア構成について述べた。ただしソフトウェア構成において実現すべき機能は全てが新規ではなく、個々の機能及び技術の中には既存のものを有効活用できるもあったため、特に本研究の特長となる、「動的グループ形成・制御」、「異種ミドルウェア連携」の実現方式について具体的に述べた。また上記の「異種ミドルウェア連携」の一部について、実現方式及び実施手段の観点からの有用性を定量的及び定性的に評価した。

第5章では、第3章及び第4章にて述べた、本研究にて提案する異種システム連携技術を適用すると効果が高いと見込まれる実問題における想定ユースケース例と、それらの想定ユースケース適用を通しての本研究の提案技術の有用性に関する評価結果について述べた。特に社会インフラ（電力）分野のユースケースとして、Virtual Power Plant（VPP）の

例を挙げて、規模の異なる、巨多の需要家設備やシステムを含む環境においても Virtual Power Plant システムの安定動作が可能であるとし、本研究の提案技術の有効性を示した。また航空宇宙の分野におけるユースケースとして、無人探査機を用いた惑星探査における「その場」観測及び探査機の保守の例を挙げて、複数の異なる種別の探査機による目的毎の連携、上記連携による状況に応じたミッション遂行が可能であるとし、本研究の提案技術の有効性を示した。

第 6 章では、第 5 章にて述べた実問題における適用例及び評価結果を踏まえて、第 3 章及び第 4 章にて述べた、本研究にて提案する異種システム連携技術を、社会インフラ分野、航空宇宙分野に適用することにより期待できる効果について述べた。また第 5 章にて述べた以外のその他の実問題や、その他の分野として産業分野への、本研究にて提案する異種システム連携技術の応用可能性についても述べた。

## 7.2. 今後の課題

本研究をさらに深度化させていく上での今後の課題について以下に述べる。

### (1) SoS システム全体の最適化

本研究にて提案する異種システム連携技術を適用することで、目的、導入時期等の異なる異種のシステム同士を、各システムの設計当初の想定・前提に関わらず、状況に応じて柔軟に連携させ、どのような状況においてもサービス提供を維持させることを可能とした。ただしこれは異種のシステムを組合せた SoS システムとしては、個々のサービス目的のために部分的に最適化したに過ぎないと言える。あるサービス維持のためにシステム間を柔軟に連携させるも、SoS システム全体で見るとムラやムダが生じている可能性もある。

ゆえに異種のシステムを組合せた SoS システムが複数のサービスを提供する場合に、個々のシステムやリソースをより有効に活用し、各々のサービスの信頼性を高めるためにも、SoS システム全体としての最適化を実現する必要がある。

### (2) システムの成長・高度化

本研究にて提案する異種システム連携技術を適用することで、異種システム同士を柔軟に連携させ、どのような状況においてもサービス提供を維持させることを可能とした。ただし上記は状況変化が発生してからの受動的な対応となり、一時的にもサービス品質の低下が生じる可能性がある。

ゆえにサービス品質をより高く維持していくためには、変化が発生してから対応するだけでなく、変化を予測して、先行して異種システム同士を柔軟に連携させ、システムを成長させ、高度化させていくことも必要である。

### (3) 開発環境の整備

本研究にて提案する異種システム連携技術を適用することによる、システム、サービスアプリケーション等を容易に開発し、展開させていくためのソフトウェア開発環境の整備が必要である。

## 謝辞

本研究を遂行し学位論文をまとめるに当たり、実に多くの方々よりご支援、ご指導を賜りました。ここに深く感謝の意を表します。

本研究の全般に渡り、懇切丁寧なご指導と格別なるご高配を賜りました、東京大学大学院工学系研究科航空宇宙工学専攻 堀浩一 教授に心より感謝の意を表します。最初に学位取得のご相談に伺ってから5年もの長きに渡り、研究としての体系化から論文のまとめ方まで実に多岐に渡りご指導を賜りました。

本論文の審査委員をお引き受けいただきました、東京大学大学院新領域創成科学研究科先端エネルギー工学専攻 横山明彦 教授、東京大学大学院工学系研究科先端学際工学専攻 岩崎晃 教授、東京大学大学院工学系研究科航空宇宙工学専攻 鈴木真二 教授、矢入健久 准教授には、本研究に対する有益かつ的確なご指摘とご助言をいただきました。ここに深く感謝の意を表します。

本研究は（株）日立製作所 研究開発グループ テクノロジーイノベーション統括本部において、多くの方々のご指導とご協力をいただいて成し遂げた成果が基となっています。副統括本部長兼横浜研究所所長 鮫嶋茂稔 博士には上司として、研究開発とは何たるか、どう取り組むべきかという基礎から粘り強くご指導いただき、本研究に取り組む機会を与えていただきました。河野克己 博士（現 大阪工業大学教授）、加藤博光 博士（現 同社戦略企画本部）には、本論文をまとめるにあたり強い動機付けを与えていただいたとともに、多大なるご指導と励ましをいただきました。ここに深く感謝の意と敬意を表します。

同社スマート情報システム統括本部 水野善弘氏、河田洋平氏、システムイノベーションセンタ 矢野浩仁 博士、同社制御プラットフォーム統括本部 足達芳昭氏、外岡秀樹氏、吉川秀之氏（現 同社 IT ビジネスサービス本部）には、本研究の提案技術を基盤ソフトウェアとして実現するにあたり、ご指導、ご協力いただきました。深く感謝致します。

ここで名前を挙げなかった多くの方々にも様々な形でご支援、ご指導いただき、本研究を推進することができました。ここに深く感謝致します。

最後に、本研究の推進と論文執筆に当たり、日々の生活の中で前向きに協力してくれ、時には温かく励まし精神的な支えとなってくれた妻 圓、長男 一帆、長女 奈々帆に心より感謝します。

## 参考文献

- 1) 河野克己, 平澤茂樹: 社会インフラの高度化に向けたシステム技術開発の動向, 電気学会論文誌 C, Vol.132, No.2, 183/186 (2012)
- 2) Mo M Jamshidi, Theme of the IEEE SMC 2005, Waikoloa, Hawaii, USA, (2005)  
<http://ieeesmc2005.unm.edu/>
- 3) Tariq Samad, Thomas Parisini: Systems of Systems, IEEE Control Systems Society, The Impact of Control Technology, 175/183 (2011)
- 4) Mark W. Maier: Architecting Principles for Systems-of-Systems, Systems Engineering, 1-4, 267/284 (1999)
- 5) 貝原俊也, 下原勝憲: System of Systems コンセプトと超スマート社会, 計測と制御, Vol.55, No.4, 284/287 (2016)
- 6) John Boardman, Brian Sauser: System of systems-The meaning of OF, Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering, 118/123 (2006)
- 7) Alex Gorod, Brian Sauser, John Boardman :System-of-Systems Engineering Management: A Review of Modern History and Path Forward, IEEE Systems, 2008
- 8) 土屋隆司: センサネットワーク研究の最新動向とその鉄道への適用可能性, 鉄道総研報告, Vol.23, No.8, 53/58 (2009)
- 9) 高田広章, 宿口雅弘, 南角茂樹, 岸田昌巳: リアルタイム OS と組み込み技術の基礎 実践  $\mu$ ITRON プログラミング, CQ 出版 (2003)
- 10) Mark Weiser: Some Computer Science Problems in Ubiquitous Computing, Communications of the ACM, Vol. 36, No. 7, 74/83 (1993)
- 11) 山田茂樹, 上岡英史: ユビキタスコンピューティングネットワークの研究動向, NII Journal, No.5, 41/47 (2003)
- 12) Gregory D. Abowd, Elizabeth D. Mynatt: Charting Past, Present and Future Research in Ubiquitous Computing, ACM Transactions on Computer-Human Interaction, Vol.7, Issue 1, 29/58 (2000)
- 13) Motohisa Funabashi, Koichi Homma, Toshiro Sasaki: Goal and Research Architecture of the Yaoyorozu Project Designing Ubiquitous Information Society in 2010, SICE Annual Conference 2004, 2578/2583 (2005)
- 14) 情報処理学会: 特集 ユビキタスコンピューティングの世界を実現する革新的ネットワーク技術, 情報処理, Vol.43, No.6, (2002)

- 15) 独立行政法人科学技術振興機構 研究開発戦略センター：戦略プロポーザル CPS (Cyber Physical Systems) 基盤技術の研究開発とその社会への導入に関する提案, (2012)  
<http://www.jst.go.jp/crds/pdf/2012/SP/CRDS-FY2012-SP-05.pdf>
- 16) 経済産業省 産業構造審議会 商務流通情報分科会 情報経済小委員会：中間取りまとめ～C P Sによるデータ駆動型社会の到来を見据えた変革～, (2015)  
[http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/report\\_001.html](http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/report_001.html)
- 17) John A. Stankovic : Research Directions for the Internet of Things, IEEE Internet of Things Journal, Vol.1, No.1, 3/9 (2014)
- 18) 日立製作所, 日経エレクトロニクス：稼ぐビッグデータ・IoT 技術 徹底解説, 日経 BP 社, (2014)
- 19) Yunchuan Sun, Houbing Song, Antonio J. Jara, Rongfang Bie: Internet of Things and Big Data Analytics for Smart and Connected Communities, IEEE Access, Vol.4, 766/773 (2016)
- 20) Industrie 4.0 Working Group : Recommendations for implementing the strategic initiative INDUSTRIE 4.0, (2013)  
<http://www.plattform-i40.de/finalreport2013>
- 21) 望月智治, ユリア・クリューガー：【ドイツ】Industrie 4.0 : 製造業に革新を, ジェトロセンサー, 2013年9月号, (2013)
- 22) Peter C. Evans, Marco Annunziata: Industrial Internet: Pushing the Boundaries of Minds and Machines, imagination at work, (2012)
- 23) 鮫嶋茂稔, 河野克己：オープン自律分散FAシステム, システム/制御/情報, Vol.43, No.1, 42/47 (1999)
- 24) Hiroshi Wataya, Katsumi Kawano, Keijirou Hayashi : The Cooperating Autonomous Decentralized System Architecture, Proc. of ISADS 95, 40/47 (1995)
- 25) (財)製造科学技術センター FA オープン推進協議会：自律分散プロトコル仕様書, (1999)
- 26) 鮫嶋茂稔, 河野克己, 足達芳昭, 新誠一：時空間型データフィールドアーキテクチャと情報制御システムの稼動中拡張, 計測自動制御学会産業論文, Vol.1, No.2, 9/14 (2002)
- 27) Object Management Group (ADSS DSIG) : White Paper for ADSS, (1998)
- 28) Hitachi, Ltd. : ADSS: Autonomous Decentralized Service System - An Agent-based Software Architecture for Large Scale and Ever Changing Information Market Places -, Hitachi Review Vol. 49, No.3, 128/128 (2000)
- 29) Object Management Group: Data Distribution Service for Real-time Systems Version 1.2, OMG Available Specification, formal/07-01-01 (2007)
- 30) 鮫嶋茂稔, 河野克己, 新誠一：環境適応サービスを狙いとした超分散オブジェクトモデルと自律プラグアンドプレイ方式, 電気学会論文誌. C, 電子・情報・システム部門

- 誌, Vol.124, No.1, 64/72 (2004)
- 31) Hidenori Yamamoto, Shigetoshi Sameshima, Katusmi Kawano : Service reconfiguration using Super Distributed Objects(SDO) in context-aware service systems, The 2nd IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems (WSTFEUS2004), 163/165 (2004)
  - 32) Hidenori Yamamoto, Shigetoshi Sameshima, Platform-independent domain management using Super Distributed Objects (SDO) in context-aware service systems, The 2005 International Symposium on Applications and the Internet (SAINT2005), 196/199 (2005)
  - 33) Object Management Group : Platform Independent Model (PIM) and Platform Specific Model (PSM) for Super Distributed Objects (SDO) Specification, format/04-11-01 (2004)
  - 34) 野村総合研究所 : ソーシャルコンピューティングのオープン化によりサービスの連携が加速, (2008)  
<http://www.nri.com/jp/news/2008/081119.html>
  - 35) 総務省 : 情報通信白書 平成 27 年度, (2015)  
<http://www.soumu.go.jp/johotsusintokei/whitepaper/h27.html>
  - 36) 愛須英之, 大槻知史, 前川智則, 松澤茂雄, 飯野穰, 荻原一徳 : 横浜スマートシティプロジェクトにおけるビル群協調型デマンドレスポンス実証実験 –大規模な社会実証実験を通して得られた知見–, 情報処理学会デジタルプラクティス, Vol.5, No.3, 173/179 (2014)
  - 37) Dominik Wachholder, Chris Stary: Enabling emergent behavior in systems-of-systems through bigraph-based modeling, Proceedings of 10th System of Systems Engineering Conference (SoSE), 334/339 (2015)
  - 38) 水野善弘, 矢野浩仁, 大河内一弥, 真下祐一 : 社会インフラを支える IT 基盤, 日立評論, Vol.93, No.12, 838/843, (2011)
  - 39) 吉川義人, 佐藤敦俊, 平澤茂樹, 高橋真人, 山本真由子 : 日立が考えるスマートシティ, 日立評論, Vol.93, No.12, 794/800 (2011)
  - 40) 日立製作所, 他 : 平成 23 年度調査委託事業「ハワイにおける日米共同世界最先端の離島型スマートグリッド実証事業」に係わる事前調査委託業務報告書, (2011.9)
  - 41) Milind Naphade, Guruduth Banavar, Colin Harrison, Jurij Paraszczak, Robert Morris: Smarter Cities and Their Innovation Challenges, IEEE Computer, Vol.44, No.6, 32/39 (2011)
  - 42) George W. Arnold : Challenges and Opportunities in Smart Grid: A Position Article, Proceedings of The IEEE, Vol. 99, No. 6, 922/927 (2011)
  - 43) Koichiro Iijima, Takashi Fukumoto, Akitoshi Shimura, Hiromitsu Kato, Shigeki

- Hirasawa: Systems Development Technology for Public Infrastructure, Hitachi Review Vol. 61, No. 3, 159/166 (2012)
- 44) 山本秀典, 鮫嶋茂稔, 関口隆昭, 加藤博光, 由良淳一, 高汐一紀: 環境適応サービスを狙いとしたミドルウェア相互接続の一方式, 電気学会論文誌. C, 電子・情報・システム部門誌, Vol.128, No. 8, 1327/1333 (2008)
  - 45) 丸山不二夫: グリッドと SOA からみる Web サービス標準技術 SOA の中核技術としての BPEL 入門(1) BPEL はどのようにサービスを結合するか?, 情報処理, Vol.48, No. 2, 191/199 (2007)
  - 46) OASIS Standard, Web Services Business Process Execution Language Version 2.0, (2007)  
<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>
  - 47) CEN-CENELEC-ETSI Smart Grid Coordination Group, CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture, (2012)  
[https://ec.europa.eu/energy/sites/ener/files/documents/xpert\\_group1\\_reference\\_architecture.pdf](https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf)
  - 48) 独立行政法人情報処理推進機構 (IPA) , 非機能要求グレード, (2014)  
<http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>
  - 49) 財団法人 日本情報処理開発協会: 平成 22 年度地域エネルギーマネジメントシステム開発事業費補助金 (地域エネルギーマネジメントシステムに関する標準化等調査事業) 共通仕様書 第 1.0 版, (2011)
  - 50) 財団法人 日本情報処理開発協会: 平成 22 年度地域エネルギーマネジメントシステム開発事業費補助金 (地域エネルギーマネジメントシステムに関する標準化等調査事業) 共通仕様書 サンプル集 第 1.0 版, (2011)
  - 51) 山岡和雄, 大谷哲夫: IEC TC57 標準化動向その 1 (全般, IEC61968 シリーズ, 61970 シリーズ), 平成 21 年電気学会電子・情報・システム部門大会講演論文集, 541/546, (2009)
  - 52) 田中立二, 芹澤善積: IEC TC57 標準化動向その 2 (IEC61850 シリーズ, 62351 シリーズ), 平成 21 年電気学会電子・情報・システム部門大会講演論文集, 547/552, (2009)
  - 53) International Electrotechnical Commission: INTERNATIONAL STANDARD IEC61850-7-3 Communication networks and systems for power utility automation - Part 7-3: Basic communication structure - Common data classes, Edition 2.0, (2010)
  - 54) International Electrotechnical Commission: INTERNATIONAL STANDARD IEC61850-7-4 Communication networks and systems for power utility automation - Part 7-4: Basic communication structure - Compatible logical node classes and data object classes, Edition 2.0, (2010)

- 55) International Electrotechnical Commission : IEC61970-403 Energy management system application program interface (EMS-API) - Part 403: Generic data access, Edition 1.0, (2008)
- 56) IEC : IEC61970-404 Energy management system application program interface (EMS-API) - Part 404: High Speed Data Access (HSDA), Edition 1.0, (2007)
- 57) International Electrotechnical Commission : IEC61968-1 Application integration at electric utilities - System interfaces for distribution management - Part 1: Interface architecture and general requirements, Edition 1.0, (2003)
- 58) International Electrotechnical Commission : IEC61968-3 Application integration at electric utilities - System interfaces for distribution management - Part 3: Interface for network operations, Edition 1.0, (2004)
- 59) 清水恵子 : 第 15 回 提案依頼書 (RFP) について (3) —ファンクション・ポイント法 (FP 法) , (2006)  
<http://itpro.nikkeibp.co.jp/article/govtech/20060223/230478/>
- 60) Capers Jones : ソフトウェア開発の定量化手法 生産性と品質の向上をめざして 第 3 版, 構造計画研究所, (2010)
- 61) 山本秀典, 加藤博光, 鮫嶋茂稔 : 情報制御システムにおける異種システム間相互接続のためのミドルウェア連携, 計測自動制御学会論文集, Vol. 49, No. 8, 780/786 (2013)
- 62) Hidenori Yamamoto, Hiromitsu Kato : Platform for Cooperating among Field Systems, Control Systems and Enterprise Systems, 2011 Tenth International Symposium on Autonomous Decentralized Systems (ISADS2011), 164/171 (2011)
- 63) 河田洋平, 矢野浩仁, 水野善弘, 寺田博文 : スマートシティ向け情報管理基盤におけるデータアクセス制御方式の提案, 研究報告マルチメディア通信と分散処理 (DPS) , Vol. 2012-DPS-152 , No. 28, 1/8 (2012)
- 64) 山本秀典, 矢野浩仁, 河田洋平, 水野善弘 : スマートシティ向け IT 基盤における異種システム連携機能, 情報処理学会デジタルプラクティス, Vol. 5, No. 3, 205/212 (2014)
- 65) 経済産業省 資源エネルギー庁 総合資源エネルギー調査会 : 長期エネルギー需給見通し小委員会 (第 5 回会合) 資料 5 デマンドレスポンスについて, (2015)  
[http://www.enecho.meti.go.jp/committee/council/basic\\_policy\\_subcommittee/mitoshi/005/pdf/005\\_09.pdf](http://www.enecho.meti.go.jp/committee/council/basic_policy_subcommittee/mitoshi/005/pdf/005_09.pdf)
- 66) 田路龍太郎 : グローバルスタンダード最前線 OpenADR の標準化動向, NTT 技術ジャーナル, Vol. 25, No. 10, 38/41 (2013)
- 67) 多田昌雄, 露崎正雄 : 社会インフラを支える電力需要調整サービスソリューション, 日立評論, Vol. 95, No. 4, 288/293 (2013)

- 68) 松尾尚志, 田子大作: VPP (Virtual Power Plant) で実現する電力の需給管理, UNISYS TECHNOLOGY REVIEW, 第 123 号, 13/22 (2015)
- 69) 川井秀之, 高山雅行: 仮想発電所 (Virtual Power Plant) 海外事例と IBM ソリューション 電力小売事業や地域エネルギー会社のビジネスをスマートに実現するソリューション, IBM ProVISION, No. 86, 66/69 (2015)
- 70) 日経テクノロジー online, 電力用語辞典,  
<http://techon.nikkeibp.co.jp/article/WORD/20121106/249620/>
- 71) 資源エネルギー庁 省エネルギー・新エネルギー部: 次世代エネルギー・社会システム協議会 (第 16 回) 資料 1 次世代エネルギー・社会システム実証事業 ～進捗状況と成果等～, (2014)  
[http://www.meti.go.jp/committee/summary/0004633/016\\_haifu.html](http://www.meti.go.jp/committee/summary/0004633/016_haifu.html)
- 72) 経済産業省 総合資源エネルギー調査会 省エネルギー・新エネルギー分科会 省エネルギー小委員会: 情報技術等の将来技術及びデータの活用, (2015)  
[http://www.meti.go.jp/committee/sougouenergy/shoene\\_shinene/sho\\_ene/pdf/report\\_01\\_01\\_06.pdf](http://www.meti.go.jp/committee/sougouenergy/shoene_shinene/sho_ene/pdf/report_01_01_06.pdf)
- 73) Koichiro Iijima, Takashi Fukumoto, Michiki Nakano: Proposal of Symbiosis ADS concept and negotiation support methods for cooperative resource allocation, 7th International Conference on System of Systems Engineering (SoSE), (2012)
- 74) 上谷勝洋: AEMS (エムス) エリアエネルギーマネジメントシステムの紹介, TSC21 オープンセミナー2007, (2007)
- 75) 峯元長, 宮本麻子, 土肥真梨子, 坂東淳子, 平澤茂樹, 佐野豊, 松村茂: 都市開発事業での協創によるサービスデザイン適用とビジョンデザイン活用, 日立評論, Vol. 97, No. 11, 674/678 (2015)
- 76) JAXA 月惑星探査推進チーム・太陽系探査ロードマップ検討小委員会: 報告書(暫定版) 「太陽系探査科学の進むべき方向 ―我々の地球をよりよく理解するために―」, (2007)
- 77) 文部科学省 国際宇宙ステーション・国際宇宙探査小委員会: 国際宇宙ステーション・国際宇宙探査小委員会 第 2 次とりまとめ ～宇宙探査新時代の幕開けと我が国の挑戦～, (2015)
- 78) 国際宇宙探査協働グループ: 国際宇宙探査ロードマップ, (2013)
- 79) 国際宇宙探査協働グループ: 宇宙探査のもたらすベネフィット, (2013)
- 80) 特定非営利活動法人ものづくり APS 推進機構: 製造オペレーションマネジメント入門 ～ISA-95 が製造業を変える!～, (2015)
- 81) 内閣府: 第 5 期科学技術基本計画, (2016)  
<http://www8.cao.go.jp/cstp/kihonkeikaku/5honbun.pdf>

- 82) 小倉博行, 馬奈木俊介, 石野正彦: 経済・社会・環境が持続可能なスマートシティ構築・運用のための評価手法の研究 ～ 超スマート社会の実現に向けて～, 電子情報通信学会技術研究報告, Vol. 115, No. 463, 29/34 (2016)
- 83) 武和秀仁, 加藤博光, 益子英昭, 石川雅一: 情報制御プラットフォームソリューションー安心・安全コンポーネントで動的な連携機能を実現ー, 日立評論, Vol. 90, No. 8, 682/691 (2008)
- 84) 計測自動制御学会: 特集「産業オートメーションにおける情報連携に向けた最新動向」, 計測と制御, Vol. 49, No. 3 (2010)

## 付録. 関連研究発表文献リスト

### 国内学術論文誌(査読有り)

- 1) 山本秀典, 鮫嶋茂稔, 関口隆昭, 加藤博光, 由良淳一, 高汐一紀: 環境適応サービスを狙いとしたミドルウェア相互接続の一方式, 電気学会論文誌. C, 電子・情報・システム部門誌, Vol.128, No. 8, 1327/1333 (2008)
- 2) 山本秀典, 加藤博光, 鮫嶋茂稔: 情報制御システムにおける異種システム間相互接続のためのミドルウェア連携, 計測自動制御学会論文集, Vol.49, No. 8, 780/786 (2013)
- 3) 山本秀典, 矢野浩仁, 河田洋平, 水野善弘: スマートシティ向け IT 基盤における異種システム連携機能, 情報処理学会デジタルプラクティス, Vol. 5, No. 3, 205/212 (2014)

### 国外学術論文誌(査読有り)

本研究に関連する該当文献無し。

### 国内講演(査読有り)

本研究に関連する該当文献無し。

### 国外講演(査読有り)

- 4) Hidenori Yamamoto, Shigetoshi Sameshima, Katusmi Kawano : Service reconfiguration using Super Distributed Objects(SDO) in context-aware service systems, The 2nd IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing Systems (WSTFEUS2004), 163/165 (2004)
- 5) Hidenori Yamamoto, Shigetoshi Sameshima, Platform-independent domain management using Super Distributed Objects (SDO) in context-aware service systems, The 2005 International Symposium on Applications and the Internet (SAINT2005), 196/199 (2005)
- 6) Hidenori Yamamoto, Hiromitsu Kato : Platform for Cooperating among Field Systems, Control Systems and Enterprise Systems, 2011 Tenth International Symposium on Autonomous Decentralized Systems (ISADS2011), 164/171 (2011)

### 国内講演(査読無し)

- 7) 山本秀典, 矢野浩仁, 河田洋平, 水野善弘: “スマートシティ向け IT 基盤における異種システム連携機能”, 情報処理学会第 77 回全国大会 デジタルプラクティスライブ (2015)

### **国外講演(査読無し)**

本研究に関連する該当文献無し。

### **表彰**

- 8) 山本秀典, 加藤博光, 鮫嶋茂稔, 矢野浩仁, 河田洋平, 水野善弘: “情報系への現場データ提供を容易化する制御系ミドルウェア技術の開発”, 2016 年度計測自動制御学会 学会賞 (技術賞) (2016)