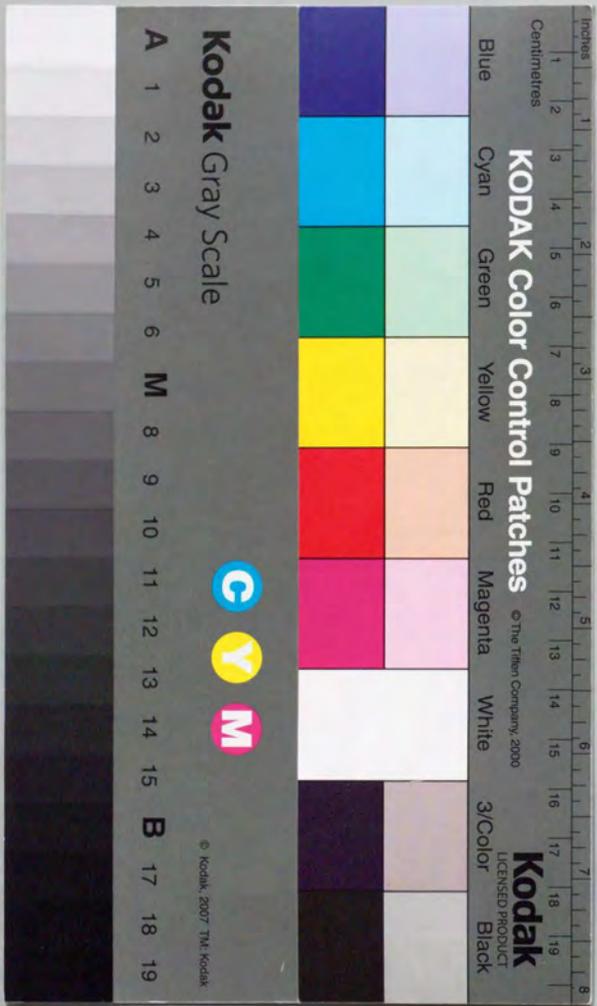


自己修復機械設計方法論

梅田 靖



1991年度博士論文

自己修復機械設計方法論

指導教官 吉川弘之教授

東京大学大学院工学系研究科

精密機械工学専攻

博士課程 97052

梅田靖

概要

近年、一般に機械システムの社会的重要性が高まっており、さらには、機械システムが故障した場合の社会的、経済的影響が重要な問題になりつつある。この問題に対し、従来技術は機械システムの要素としての高信頼化を目的としてきたが、このような手法は技術的、経済的に飽和状態に達しつつあると考えられる。そこで本研究では見方を変え、要素として故障が発生することを前提としつつシステム全体としては要素の故障に耐えるような「柔らかく壊れる」機械システムの設計方法論を提案することを目的とする。

本論文では、この柔らかく壊れる機械を機能的な見地から「自己修復機械」と呼び、故障診断、修復という事後保全作業の方法を機械システム上で自動化することにより実現する。このために、知識工学的手法を利用し保全作業を自動化するための知的判断部の構成方法、および、保全の自動化に適した機械構造の設計方法という二つの側面から議論を行ない、さらに、これらを統合化した計算機・メカトロニクス機械システムとして自己修復機械を構築する。

保全作業の特徴は、故障診断や修復計画といった知的判断作業と点検、清掃、部品交換作業といった実際の動作が組み合わされて実現されることである。さらにこのとき、作業者は様々な分野のさまざまな知識を利用して保全作業を行なわなければならない(多様性)、保全作業を行なうためには対象系内の保全を行なうべき箇所とその構造、性質の同定をまず行なわなければならないという意味で制御技術に比較して不定性が高く、さらに、対象系において同一の故障が繰り返し発生するとは限らず、従来的一般技術で行なわれている経験・帰納という方法が単純に適用できない(非繰り返し性)という特徴がある。

本研究では、「機能保全」、および、「機能トレードオフ」という保全戦略に基づき、自己修復機械を部品交換の自動化によらず実現する。機能保全とは、属性的には

故障しても必要最低限の機能を発現することを目的とした保全の戦略である。従来一般に行なわれている修復は、故障した機械を正常な属性的状態に戻すことを目的としており、「属性保全」と呼ぶことができ、機能保全と属性保全はその修復の目的が異なるという点で区別することができる。さらに故障が発生した緊急時においては、必ず発現しなければならない機能と必ずしも発現しなくて良い機能があると考えることができる。この必要機能を選択する過程を「機能トレードオフ」と呼び、修復計画時にこの過程を含めることにより、選択された機能に対して機能保全を行ない、柔軟に修復を実現することが可能になる。この柔軟かく壊れる機械を実現するために、本論文では制御型自己修復機械、機能冗長型自己修復機械という二つのタイプの自己修復機械を提案する。

機械部は、状態監視機能と柔軟な修復実行機能を持たなければならない。このとき本研究では、部品交換のみによらない修復の実行方法を考える。その実現方法として、制御型自己修復機械は、対象系上のアクチュエータ制御により修復を実行する。従来の制御技術との相違は、後述する知的判断部により、故障の状況、機械部の修復実行能力に応じて制御するアクチュエータを動的に決定、すなわち、制御ループを動的に決定することにある。これにより、対象系が故障により構造変化を引き起こした場合やアクチュエータが操作不能になった場合にも、有効に修復を実行することが可能になる。単一の機械システムで状態監視機能、制御型の修復実行機能を実現するためには、適当なセンサ、アクチュエータの配置を考察し、また、機能保全を実現容易にする機械構造を考察しなければならない。本論文では、センサ、アクチュエータの配置、および、機能保全を可能にする機械構造という三項目に関して制御型自己修復機械の機械部の設計支援方法を議論する。

機能冗長型自己修復機械は、機械がその構造を動的に変更することにより、各部品の潜在機能を利用して冗長性を得るというもので、従来の冗長系のように並列的に予備部品を付加することなく多くの冗長性を得ることができるという長所を持つ。これにより、制御型自己修復機械よりも広い範囲の故障に対応可能になると考えられる。この機能冗長型の機械部を設計するために、本論文では、概念設計段階において機能冗長系を対象機械に付加する設計手法を提案する。

知的判断部は、保全作業の自動化を実現するために、保全の特徴である不定性、非線り返し性、および、多様性に対応可能な形で、故障診断の自動化、および、機能保全を実現する修復計画の自動化を実現しなければならない。そこで本研究では、機械

のモデリング手法、故障診断の自動化手法、および、修復計画の自動化手法という三点について議論を行なう。

まず機能保全を実現するために、機械のモデリング手法として、機械の機能を明示的に取り扱い可能にする FBS(Function-Behavior-State) ダイアグラムを提案する。FBS ダイアグラムにおいて、人間の主観的な判断に基づく「機能」は、物理的な知識に基づき記述可能な挙動との対応付けにより記述される。FBS ダイアグラムにより主観的機能の記述を可能に、また、挙動、状態レベルでの計算機による無矛盾性の管理が実現可能になる。ここでは、この FBS ダイアグラムの有効性を実験的な設計支援システム「FBS モデラー」のインプリメンテーションにより検証する。

故障診断を自動化するために、上記のモデリング手法を用い、対象モデルに基づき定性物理を用いた故障診断手法を提案する。まず、因果的に関連する複数の故障に対応し、また、構造変更を引き起こす故障を診断するために、単一故障発生過程仮説を提案し、この仮説に基づく故障診断手法を提案する。ここでは、対象系上で発生し得る物理現象を対象モデルと独立に明示的に記述する。この仮説と物理現象に関する知識を用いることにより、本故障診断手法は保全の特徴である不定性、非線り返し性に有効に対処可能になるが、多様性に対応するために有効に知識獲得を行なう方法は今後の課題である。さらに、定性物理を用いた手法を実際の機器に適用するために、計算機上の記号的物理世界と現実世界を柔軟に対応付けるファジィ定性物理とファジィ定性物理を用いた故障診断手法を提案する。

修復計画作業を自動化するために、上記のモデリング手法を用いた対象モデルに基づく修復計画手法を提案する。本手法において、修復実行時の対象挙動をシミュレーションすることにより、修復実行時の副次的影響を評価し、副次的影響に対する修復計画を導出することが可能になる。これにより、故障に応じた柔軟な修復計画が可能になる。

さらに、制御型自己修復機械、および、機能冗長型自己修復機械のそれぞれの実験機を構築し、実験を行なうことにより、本研究で提案した手法の有効性を検証する。この結果、制御型自己修復機械に関しては、一部の故障に対し有効に機能保全を実現可能であるが、重要な部品の故障には対処不可能であることが明らかになった。この点に関して機能冗長型自己修復が有効である。機能冗長型自己修復機械は機械の故障に対する強靱性を得るために、機械各部に性能には直接には貢献しないという意味での無駄を作り込むという設計思想を提案するものであり、その基本的な有効性が検証

された。機能トレードオフを動的に実行すること、複数の機能冗長系を同時利用すること、さらには、故障発生時に機能冗長系をシステムが動的に発見、利用することによりさらに柔軟な機能保全を実現可能になると考えられる。

目次

1	序論	1
1.1	はじめに	3
1.2	目的と方法	5
1.3	本論文の構成	7
2	保全作業自動化の必要性	9
2.1	保全作業自動化の必要性と実際	11
2.1.1	保全作業の実際	11
2.1.2	故障診断作業の実際	12
2.1.3	修復作業の実際	15
2.1.4	保全作業自動化のための課題	16
2.2	保全の理論と問題点	20
2.2.1	保全の理論	20
2.2.2	保全性設計	20
2.2.3	保全の自動化手法	21
2.2.4	保全の理論の課題	26
2.3	本研究の位置付け	28
3	自己修復機械実現方法論	31
3.1	自己修復機械の枠組	33
3.1.1	自己修復機械に必要な機能	33
3.1.2	本研究における自己修復機械の実現方法	33
3.1.3	機能保全と柔らかく壊れる機械	35
3.1.4	他の手法との比較	37
3.2	知的判断部の実現方法	40
3.3	機械部の実現方法	42
3.3.1	制御型自己修復機械	43
3.3.2	機能冗長型自己修復機械	45
3.4	第3章のまとめ	49
4	自己修復機械のモデリング	51
4.1	機械のモデリング手法	53
4.1.1	属性のモデリング法	53

4.1.2	挙動のモデリング法	53
4.1.3	機能のモデリング法	54
4.2	定性物理	61
4.2.1	定性物理の基本的手法	61
4.2.2	定性物理の研究動向と問題点	62
4.3	FBS ダイアグラム	64
4.3.1	FBS ダイアグラムの提案	64
4.3.2	機械の階層	69
4.3.3	FBS ダイアグラムの特徴	75
4.3.4	FBS ダイアグラム上での設計	76
4.4	インプリメンテーション	78
4.4.1	FBS ダイアグラムの実現方法	78
4.4.2	FBS モデラーと実行例	84
4.5	FBS ダイアグラムの考察	91
4.5.1	インプリメンテーションに関する考察	91
4.5.2	機械の階層的表現	93
4.5.3	モデリング手法としての FBS ダイアグラム	94
4.6	自己修復のための機械のモデリング	95
4.7	第4章のまとめ	97
5	故障診断の自動化	99
5.1	従来の故障診断手法	101
5.2	単一故障発生過程仮説に基づく故障診断手法	104
5.2.1	単一故障発生過程仮説	104
5.2.2	知識表現	105
5.2.3	挙動シミュレーションの方法	113
5.2.4	故障診断の方法	119
5.2.5	インプリメンテーション	123
5.2.6	考察	127
5.3	ファジィ定性物理を用いた故障診断手法	133
5.3.1	ファジィ定性物理	133
5.3.2	知識表現	138
5.3.3	劣化シミュレーション	139
5.3.4	故障診断の方法	144
5.3.5	インプリメンテーション	145
5.3.6	考察	147
5.4	第5章のまとめ	150
6	修復計画の自動化	151
6.1	従来の修復計画手法と問題点	153
6.2	制御型自己修復機械の修復計画	157
6.2.1	知識表現	157
6.2.2	修復計画の方法	157
6.2.3	考察	162

6.3	機能冗長型自己修復機械の修復計画	165
6.3.1	知識表現	165
6.3.2	修復計画の方法	167
6.3.3	考察	168
6.4	第6章のまとめ	171
7	制御型自己修復機械設計方法論	173
7.1	制御型自己修復機械の設計方法	175
7.1.1	制御型自己修復機械設計の要求項目	175
7.1.2	機械部の設計方法	175
7.2	制御型自己修復機械における修復の実現	182
7.3	実験機の設計	187
7.3.1	実験機の構成	187
7.3.2	知識の記述	190
7.3.3	推論部と機械部の接続	192
7.4	制御型自己修復実験	195
7.4.1	実行例1	195
7.4.2	実行例2	198
7.4.3	実行例3	198
7.5	考察	201
7.5.1	実験の特徴	201
7.5.2	制御型自己修復機械の有効性	201
7.5.3	制御型自己修復機械と制御技術の相違	202
7.5.4	故障診断 / 修復計画システムの有効性と課題	204
7.5.5	推論部と機械部の接続	206
7.5.6	機械部の設計方法	207
7.6	第7章のまとめ	208
8	機能冗長型自己修復機械設計方法論	209
8.1	機能冗長型自己修復機械の設計方法	211
8.1.1	機能冗長型自己修復機械設計の要求項目	211
8.1.2	機能冗長設計の方法	211
8.1.3	FBS モデラーを用いた機能冗長設計支援	215
8.2	機能冗長型自己修復機械における修復の実現	224
8.3	実験機の設計	226
8.3.1	機能冗長系の設計	226
8.3.2	実験機の構成	229
8.3.3	知識の記述	231
8.3.4	推論部と機械部の接続	234
8.4	機能冗長型自己修復実験	237
8.4.1	実行例1	237
8.4.2	実行例2	237
8.5	考察	241
8.5.1	機能冗長系の評価	241

8.5.2	故障診断/修復計画システムと機能冗長	242
8.5.3	機能冗長設計の方法	242
8.5.4	機能冗長型自己修復機械の拡張	243
8.6	第8章のまとめ	245
9	結論	247
9.1	結論	249
9.2	本研究の課題	251
9.3	今後の自己修復機械	252
	謝辞	253
	参考文献	257
	発表論文	267

図一覧

2.1	保全の分類 [JIS87]	11
2.2	化学プラントの事故原因 [Ooshima91]	13
2.3	修復のタスク [Sato90]	16
2.4	機械の属性、機能表現	17
2.5	並列冗長系の例	21
2.6	故障診断システムの主なアーキテクチャ [Nishida91]	24
2.7	設計作業と保全作業	29
3.1	属性保全型自己修復機械と機能トレードオフ型自己修復機械	36
3.2	自己修復機械の基本構成	38
3.3	知的判断部のアルゴリズム	41
3.4	熱間圧延工場の概略 [Sato90]	45
3.5	機能冗長な自動車の例	46
3.6	部品冗長な自動車の例	46
3.7	機能冗長による機能の再配置	47
4.1	定性物理のスコープ [Nishida88]	55
4.2	押し出し機の機能構造 [Rodenacker71]	56
4.3	プロセスの記述例 [Forbus84]	62
4.4	文鎖の状態記述	65
4.5	視点の違いによる挙動表現の違い	67
4.6	保持機能の例	68
4.7	機能、挙動、状態間の関係	69
4.8	FBS ダイアグラム	70
4.9	ブザー	71
4.10	ブザーのFBSダイアグラム上での表現例	72
4.11	階層構造構築の例(1)	73
4.12	電氣的視点から見たブザー	73
4.13	階層構造構築の例(2)	74
4.14	階層構造構築の例(3)	74
4.15	複雑な機能階層構造の例	75
4.16	機能プロトタイプの記述例	79
4.17	挙動プロトタイプの記述例	82
4.18	フィジカル・フィーチャーの記述例	83

4.19	FBS モデラーの構成	84
4.20	FBS モデラーの基本的アルゴリズム	86
4.21	FBS モデラーの実行例 (1)	87
4.22	FBS モデラーの実行例 (2)	88
4.23	FBS モデラーの実行例 (3)	89
5.1	簡単な機械の例	102
5.2	ショートした電気回路	105
5.3	故障発生過程の例	106
5.4	部品、関係クラスの階層	107
5.5	ギアの定義	107
5.6	ギア・ギア噛み合い関係の定義	108
5.7	「ギア割れ」現象	111
5.8	対象モデルの例	112
5.9	パラメータモデルの例	120
5.10	故障診断システムの構成	124
5.11	故障診断システムのアルゴリズム	124
5.12	診断対象の複写機	125
5.13	複写機のパラメータモデル	126
5.14	故障診断システムの実行例	127
5.15	実際の故障状態	128
5.16	ファジィ量空間の例	135
5.17	劣化のメカニズム [Takada77]	140
5.18	故障診断の枠組	140
5.19	劣化曲線の近似	141
5.20	劣化要因パラメータのメンバーシップ関数の例	142
5.21	ファジィ故障診断システムの構成	146
5.22	ファジィ故障診断システムのアルゴリズム	146
5.23	ファジィ故障診断の実行例	147
6.1	修復のタスク (再掲)[Sato90]	153
6.2	故障診断 / 修復計画の構成	156
6.3	パラメータモデルの例	160
6.4	修復計画の例	162
6.5	修復操作がループを描く例	164
6.6	機能冗長の適用によるモデルの変化	166
6.7	機能冗長修復計画のアルゴリズム	167
6.8	故障モデルと Alternative Model	168
6.9	機能冗長修復計画の例	169
7.1	制御型自己修復機械に対する FBS ダイアグラムの例	177
7.2	パラメータモデルの構造	181
7.3	修復計画の例 (再掲)	182
7.4	二つのアクチュエータを操作する場合の操作空間	183

7.5	制御型修復計画、実行のアルゴリズム	184
7.6	修復実行の主なパターン	186
7.7	制御型自己修復実験機の内部構造	189
7.8	複写機の動作	190
7.9	制御型自己修復機械システムの構成	191
7.10	制御型自己修復機械システムのアルゴリズム	191
7.11	現在の正常状態を表すランドマークの生成	192
7.12	実験機の対象モデル (挙動・状態レベル)	193
7.13	実験機の対象モデル (機能レベル)	194
7.14	センサデータ定性化知識の例	194
7.15	アクチュエータ操作知識の例	194
7.16	故障診断 / 修復計画システムの実行画面 (実行例 1)	196
7.17	PC9801 の実行画面 (実行例 1)	197
7.18	修復前後の出力画像	197
7.19	故障診断 / 修復計画システムの実行画面 (実行例 2)	199
7.20	故障診断 / 修復計画システムの実行画面 (実行例 3)	200
8.1	複写機の FBS モデルの一部	213
8.2	機能冗長設計支援システムの構成	216
8.3	機能冗長設計支援システムのアルゴリズム	217
8.4	自動車のエンジン前部	220
8.5	入力された自動車の FBS モデル	221
8.6	機能冗長候補の導出	222
8.7	機能冗長設計の例	223
8.8	機能冗長型修復計画、実行のアルゴリズム	225
8.9	複写機のドラム周辺部	226
8.10	メインチャージャの機能を転写チャージャで代替させる機能冗長系	229
8.11	機能冗長を付加したドラム周辺部	230
8.12	機能冗長型自己修復実験機の外観	231
8.13	機能冗長実験機の対象モデル (挙動・状態レベル)	233
8.14	機能冗長実験機の対象モデル (機能レベル)	234
8.15	機能冗長系候補 (A)	235
8.16	機能冗長系候補 (B)	236
8.17	機能冗長型推論システムの実行画面 (実行例 1)	238
8.18	機能冗長系 (A) を利用した出力画像	239
8.19	機能冗長型自己修復実験機のランプ配置	239

1.1 製造業における保全費用 (1979年) [Anonymous82] 12

1.2 故障解析の各種方法 [Shiomi91] 13

1.3 故障診断に用いられる知識 (診断方法 1) [Umeda89b] 14

1.4 故障に対する修復処置の例 [Sato90] 15

1.5 制御技術と比較した保全の不定性 [Yoshikawa83] 19

1.6 代表的な劣化現象とその典型的な検出法 [Akizuki91] 25

2.1 高信頼化/自己修復機械の各手法 38

2.2 部品冗長と機能冗長 48

3.1 機能プロトタイプの定義 78

3.2 挙動プロトタイプの定義 81

4.1 パラメータの定義 108

4.2 パラメータ間の関係の定義 109

4.3 拘束条件式の種類 109

4.4 現象の発生条件の記述 110

4.5 現象の影響の記述 111

4.6 各影響述語の作用 115

4.7 状態遷移則 [Kurumatani89] 117

4.8 故障モデル間でのパラメータ値の分散 123

4.9 劣化の特性値への影響の例 143

4.10 劣化の現象発生度合への影響の例 143

5.1 制御型自己修復実験機の諸元 188

5.2 記述した現象知識 193

6.1 機能冗長型自己修復実験機の諸元 232

表一覧

2.1 製造業における保全費用 (1979年) [Anonymous82] 12

2.2 故障解析の各種方法 [Shiomi91] 13

2.3 故障診断に用いられる知識 (診断方法 1) [Umeda89b] 14

2.4 故障に対する修復処置の例 [Sato90] 15

2.5 制御技術と比較した保全の不定性 [Yoshikawa83] 19

2.6 代表的な劣化現象とその典型的な検出法 [Akizuki91] 25

3.1 高信頼化/自己修復機械の各手法 38

3.2 部品冗長と機能冗長 48

4.1 機能プロトタイプの定義 78

4.2 挙動プロトタイプの定義 81

5.1 パラメータの定義 108

5.2 パラメータ間の関係の定義 109

5.3 拘束条件式の種類 109

5.4 現象の発生条件の記述 110

5.5 現象の影響の記述 111

5.6 各影響述語の作用 115

5.7 状態遷移則 [Kurumatani89] 117

5.8 故障モデル間でのパラメータ値の分散 123

5.9 劣化の特性値への影響の例 143

5.10 劣化の現象発生度合への影響の例 143

6.1 制御型自己修復実験機の諸元 188

6.2 記述した現象知識 193

7.1 機能冗長型自己修復実験機の諸元 232

表一覽

1	序論	1
2	第1章	2
3	第2章	3
4	第3章	4
5	第4章	5
6	第5章	6
7	第6章	7
8	第7章	8
9	第8章	9
10	第9章	10
11	第10章	11
12	第11章	12
13	第12章	13
14	第13章	14
15	第14章	15
16	第15章	16
17	第16章	17
18	第17章	18
19	第18章	19
20	第19章	20
21	第20章	21
22	第21章	22
23	第22章	23
24	第23章	24
25	第24章	25
26	第25章	26
27	第26章	27
28	第27章	28
29	第28章	29
30	第29章	30
31	第30章	31
32	第31章	32
33	第32章	33
34	第33章	34
35	第34章	35
36	第35章	36
37	第36章	37
38	第37章	38
39	第38章	39
40	第39章	40
41	第40章	41
42	第41章	42
43	第42章	43
44	第43章	44
45	第44章	45
46	第45章	46
47	第46章	47
48	第47章	48
49	第48章	49
50	第49章	50
51	第50章	51
52	第51章	52
53	第52章	53
54	第53章	54
55	第54章	55
56	第55章	56
57	第56章	57
58	第57章	58
59	第58章	59
60	第59章	60
61	第60章	61
62	第61章	62
63	第62章	63
64	第63章	64
65	第64章	65
66	第65章	66
67	第66章	67
68	第67章	68
69	第68章	69
70	第69章	70
71	第70章	71
72	第71章	72
73	第72章	73
74	第73章	74
75	第74章	75
76	第75章	76
77	第76章	77
78	第77章	78
79	第78章	79
80	第79章	80
81	第80章	81
82	第81章	82
83	第82章	83
84	第83章	84
85	第84章	85
86	第85章	86
87	第86章	87
88	第87章	88
89	第88章	89
90	第89章	90
91	第90章	91
92	第91章	92
93	第92章	93
94	第93章	94
95	第94章	95
96	第95章	96
97	第96章	97
98	第97章	98
99	第98章	99
100	第99章	100
101	第100章	101

第 1 章

序論

序論

本書の目的は、現代の教育思想を、その歴史と理論の両面から明らかにすることにある。そのために、まず教育思想の歴史的変遷を概観し、次にその理論的基礎を考察する。この二つの側面から、教育思想の発展の過程を追跡し、その特質を明らかにしようとする。

教育思想とは、教育の目的、内容、方法、評価などについて、理論的に考察されたものである。それは、教育実践の基盤となる。したがって、教育思想を研究することは、教育実践を改善するために不可欠である。本書は、その研究の一助をなすことを目指している。

本書の構成は、序論、第1章から第10章までの構成となっている。序論では、本書の目的と研究の方法について述べている。第1章では、教育思想の歴史的変遷を概観し、その特質を明らかにしている。第2章では、教育思想の理論的基礎を考察し、その発展の過程を追跡している。第3章から第10章までは、現代の教育思想の主要な流派について、その理論的基礎と実践的意義を考察している。

本書は、教育研究者、教育実践者、および教育に関心のある一般の人々にとって、有益な読書となすことを目指している。また、本書は、教育思想の発展の過程を追跡し、その特質を明らかにすることを目的としている。したがって、本書は、教育思想の研究の一助をなすことを目指している。

本章では自己修復機械の必要性、および、本研究の目的と方法について述べる。さらに、本論文の構成を示す。

1.1 はじめに

最近の環境、資源、安全性に対する問題意識の高まりは、機械の性能、信頼性に対する評価基準を変えつつある。すなわち、高い性能を発揮する代わりに使い捨てられる機械が機械の必ずしも正しい姿ではないという認識が高まり、環境にやさしく、省資源、長寿命であり、かつ、高安全性を持つ機械というものが求められつつある。例えば、IEA(国際エネルギー機関)の勧告により既に日本での石油専燃火力発電所の新設は認められておらず、既存の発電所を「保全」しつつ、石油の消費、環境への影響を最小限に抑えて使用しなければならない状況にある。吉川[Yoshikawa83]が指摘しているように、このような問題に技術として取り組むのが「保全」であり、その意味で保全技術の必要性が今ほど叫ばれているときはない。

保全技術は、直接的には点検、清掃等の予防保全と故障診断、修復等の事後保全に分かれ、また間接的には、機械に安全性、対環境性、信頼性等をいかにして作り込むかという、保全的視点からの機械の設計戦略の問題と強く結びつく技術分野である。

予防保全、事後保全の具体的保全作業においては、原子力発電所等での危険作業に対する人間の安全性、労働力不足に由来する省力化、自動運転長期化の実現等の問題により、保全作業を自動化することが必要不可欠になりつつある。一方、近年の知識工学、人工知能の発展により、人間の「判断」に相当する推論の一部自動化が可能になりつつあり、また、センサ技術、ロボット技術、メカトロニクス技術の発展により、人間の「五感」、「動作」に相当する部分の一部自動化や機械の柔軟な制御が可能になりつつある。これらの技術を背景として、保全作業の自動化や支援の可能性が高まっているが、現状での研究レベルは充分なものとは言えず、より多くの研究、特に特定の対象に対する特定の問題を解決する手法のみならず、より一般的、汎用的な議論が必要になっていると考えられる。

保全的観点からの機械の設計という問題に対して、従来技術は、信頼性工学という形で機械要素としての高信頼化や故障を起こさない技術開発を中心的な課題として発達してきた。その技術的中心に原子力発電所、宇宙工学、航空工学、軍事産業が位置している。しかし、このような従来のアプローチはある意味で限界に来ているのではないかと思われる。例えば、スリーマイル島、チェルノブイリの原子力発電所に見られるように、信頼性設計の塊のような原子力発電所が小さな故障や操作ミスが重なることにより破滅的な事故を引き起こしてしまう。また、航空機事故も無くならず、

しばしば発生する。その原因として、従来の信頼性工学的なアプローチの技術的飽和、資源、人的資源、コスト等の問題で限界を越えた技術的要求の存在、さらには、機械の複雑化が進行し、機械に対して起こり得る事象を全て予測することがそもそも不可能であること等が考えられる。

そこで本研究では、機械に異常や故障が発生することを前提として、故障が発生した場合にも機械システム全体としては異常や故障に耐え大きな破滅的故障に至らずに、「柔らかく壊れる」機械というものを考えるべきであるという立場を取る。実際、上記の知識工学、メカトロニクス技術の発展により、このような柔らかく壊れる機械を実現可能な時期に来ていると考えられる。そこで本研究では、この柔らかく壊れる機械を機能的な意味からの「自己修復機械」と捉え、知的機械システムとしての「自己修復機械」の実現を目標とすることにする。

1.2 目的と方法

本研究の目的は、柔らかく壊れる機械としての「自己修復機械」の概念を提案し、その設計方法論を明らかにすることにある。

「自己修復機械」、すなわち機械に故障が発生した場合に機械自身が故障を修復し、要求される機能を発揮し続ける機械は、保全の自動化の一つの究極の姿といえることができる。しかし、現在の技術で完全な自己修復機械を実現することは不可能であり、以下の三項目を満たすものを本研究における「自己修復機械」とする。

- 一定期間のメンテナンス・サイクルを考え、その期間中は少なくとも必要最低限の機能を発揮する機械システム
- 属性的には故障しても機能的には故障しない機械システム
- 自己の状態を把握し、機能を発揮するという目的を実現するため自己の状態、構造を変化させる機械システム

このような自己修復機械を実現するために、本研究では以下の三点を明らかにすることを本研究の具体的な目的とする。

1. 自己修復機械の概念および自己修復性実現のための方法論を明らかにする。
2. 自己修復機械の知的判断部の実現方法を提案する。
3. 自己修復性を持つ機械部の設計方法を提案する。

また、以上の手法の有効性を検証するために、自己修復機械の実験機を構築し、実際にその有効性を実験により確かめることも本研究の目的とする。

本研究では、以上の目的を実現するための方法として以下のような方法を取る。

1. に対しては、機能的観点から保全を考える「機能保全」という立場に立ち自己修復性について考察を行なう。またこの考察に基づき、「制御型自己修復」、「機能冗長型自己修復」という二つの修復戦略を提案し、この戦略に基づき自己修復機械の実現を試みる。このとき、自己修復機械の設計方法論とは、上記の戦略に基づき機械システム内で状態監視機能、故障判定機能、故障診断機能、修復計画機能、および、修復実行機能を実現するための方法論であるとする。このため、本研究では実際の保全作業を参考にして、上記の戦略に基づく保全作業の自動化手法を提案することが、すなわち、上記の各機能の設計方法論を提案することであるという立場を取る。このとき、故障判定機能、故障診断機能、修復計画機能は知的判断部により実現し、状態監

視機能、修復実行機能は機械部により実現する知的判断部とメカトロニクス機械部を持つ、知的メカトロニクス機械を対象とすることにする。

2. に対しては、知識工学的手法、特に「定性物理」、「モデルベース推論」という考え方が自己修復機械の知的判断部に有効であるという立場から、機能を含めた機械のモデリング手法、故障診断の自動化手法、および、修復計画の自動化手法を提案する。

3. に関しては、前述の制御型自己修復、機能冗長型自己修復に適した機械の構造という観点からメカトロニクス機械を対象に機械の設計方法、特に、状態監視機能、修復実行機能の実現方法を中心として考察し、その設計を支援する方法を提案する。

さらに、実際に実験機を構築し、これらの手法の有効性と問題点を明らかにする。

1.3 本論文の構成

本論文は全9章からなる。

第2章「保全作業自動化の必要性」では、保全作業自動化の必要性と課題を整理し、本研究の位置付けを行なう。

第3章「自己修復機械実現方法論」では、自己修復機械の枠組と本研究における実現方法論を議論する。ここでは、一般的な自己修復機械に必要な機能と実現方法を整理し、本研究で用いる方法とその理由を述べる。

第4章、第5章、および、第6章では、第3章で議論した自己修復機械の知的判断部の実現方法について具体的に議論する。

第4章「自己修復機械のモデリング」では、自己修復機械を計算機上にモデリングする方法について議論する。まず、機械のモデリング手法について整理し、その結果から、機械の機能を含めたモデリング手法としてFBSダイアグラムを提案する。さらに、概念設計支援ツール「FBSモデラー」によりこの手法の有効性を検証する。最後に、自己修復機械を表現するために本研究で用いる手法について述べる。

第5章「故障診断の自動化」では、故障診断を自動化するための手法について議論する。まず、従来の故障診断の手法について概観し、その問題点を整理する。次に、従来の手法の問題点を解決するため、単一故障発生過程仮説を用いた故障診断手法、ファジィ定性物理を用いた故障診断手法を提案する。

第6章「修復計画の自動化」では、修復計画を自動化するための手法について議論する。まず、従来の修復の自動化手法を概観し、自己修復機械において修復を自動化するための問題点を整理し、その結果から制御型修復計画手法、および、機能冗長型修復計画手法を提案する。

第7章、第8章では、第3章の自己修復機械の枠組に関する議論を受けて、機械部の設計方法を議論する。さらに、第4章～第6章の知的判断部の実現に関する議論と併せて、実際に自己修復機械のプロトタイプを製作、実験することにより、その有効性と問題点を明らかにする。

第7章「制御型自己修復機械設計方法論」では、制御型自己修復機械の機械部の設計方法と実験機による実験結果を示し、制御型自己修復機械の有効性と問題点について考察する。

第8章「機能冗長型自己修復機械設計方法論」では、機能冗長型自己修復機械の機

械部の設計方法を提案し、さらに、実験機による実験結果を示す。これに基づき、機能冗長型自己修復機械の有効性と問題点について考察する。

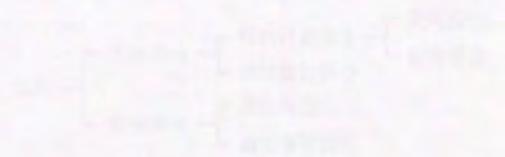
第9章「結論」では、本研究の結論と課題を述べる。最後に、今後の自己修復機械のあり方について議論し、本研究の役割について述べる。

（以下は非常に薄い文字の本文が続く）

第2章

保全作業自動化の必要性

（以下は非常に薄い文字の本文が続く）



（以下は非常に薄い文字の本文が続く）

本章では、保全作業自動化の必要性と課題を整理し、本研究の位置付けをおこなう。まず、第2.1節で保全作業の実際について整理し、保全作業自動化の必要性と課題について議論する。次に、第2.2節で保全の理論について概観し、保全の自動化のための課題を整理する。さらに、第2.3節で本研究の位置付けを行なう。

2.1 保全作業自動化の必要性と実際

2.1.1 保全作業の実際

保全とは、「アイテムを使用及び運用可能状態に維持し、又は故障、欠点などを回復するための全ての処置及び活動」であり、図2.1のように分類される[JIS87]。予防保全とは、故障を未然に防止し、使用可能状態を維持するために計画的に行なう保全であり、具体的には検査、清掃、調整、定期交換等の作業を含む。事後保全とは、故障が起こった後で機械を運用可能状態に回復するために行なう保全であり、故障診断、修復作業により実現される。このように保全は、劣化、故障しない機械が存在しない限り、機械を使用する場においては必ず必要になる作業なのである。



図2.1: 保全の分類 [JIS87]

経済的な要因でいえば、文献[Anonymous82]によると表2.1に示すように日本の製造業における保全費用の総額は2兆4000億程度、売上高の1.33%に達している。また、1988年におけるヨーロッパの製造業の保全費用の売上高に対する割合として、イギリス3.8%、西ドイツ4.0%、スペイン4.15%、フランス4.2%、イタリア4.5%(245億ドル)というデータもある[Roversi91]。このように、保全費用は一定の割合で確実に存在する。

以上のように、保全の特徴は機械には必ず必要で、かつ、コストのかかる作業であることである。さらに問題なことに、これら保全作業が主に労働集約型の作業によってのみ実現されているため、生産性という見地からは低いレベルにあり、かつ、保全作業の現場は原子力発電所の格納容器内等の危険な場所であることが多い[Yoshikawa83]。これら保全作業の生産性、労働者の安全性を高め、かつ、環境への影響を最小限にするためには保全作業の自動化を実現する必要があるといえる。

本研究では、対象を特に事後保全に定め、故障診断、修復作業を自動化するための方法を考えることにする。以後、第2.1.2項で故障診断作業の実際について、第

表 2.1: 製造業における保全費用 (1979年)[Anonymous82]

	m.c./ 製造原価 (%)	m.c./ 売上 (%)	m.c. ¹ (10 億円)
食料品	0.6	0.48	98.8
パルプ、紙、紙加工	2.1	1.68	94.3
化学	3.5	2.66	398.1
石油製品	1.7	1.5	145.5
鉄鋼業	5.0	4.0	630.9
非鉄	2.2	1.76	110.7
金属製品	2.2	1.76	172.4
一般機械	0.7	0.56	87.3
電機機械	0.5	0.375	69.1
輸送機械	0.8	0.7	151.0
精密機械	0.5	0.36	10.7
その他		1.12	481.1
計		平均 1.33	2449.9

¹m.c.: maintenance cost の略

2.1.3項で修復作業の実際について整理し、第2.1.4項で保全の特徴について整理することにする。

2.1.2 故障診断作業の実際

故障診断作業とは、「機械に不具合が発生した場合に、故障の症状 (symptom) から、その故障の症状を引き起こしている故障 (fault) を同定する作業」であると本研究では定義する。一般に、故障の症状は機械の機能や挙動の変化として、故障は機械の属性、構造の変化として観察される。ここで言う、機械の「機能」、「挙動」、「属性」、および、「構造」の定義は第4章で議論する。

大島 [Ooshima91] は、化学プラントの事故原因 (上記の故障に対応) を図2.2のように分類し、その診断方法を過去のデータに基づく経験的診断方法と機械構造の因果関係に基づく論理的診断方法の二種類に分けている。また、塩見 [Shiomi91] は、故障診断、および、その発生メカニズムの検査方法を表2.2のように整理している。

本研究では、小型複写機を対象に主に駆動系に関して故障事例の収集と、実際の技術者の行なう故障診断方法の検討を行なった [Umeda89b]。文献 [Umeda89b] による



図 2.2: 化学プラントの事故原因 [Ooshima91]

表 2.2: 故障解析の各種方法 [Shiomi91]

方法	特徴
機能ブロック図、故障の木 (FTA)	全体と部分の階層的・機能的関連をみる
事象の木 (ETA)	原因から結果までの経時的連鎖をみる
故障モード・影響解析 (FMEA)	部分に発生した異常の全体に対する影響の評価
技術計算、統計解析	強度・ストレスの評価、有限要素法、寿命と分布、多変量解析、時系列分析など
固有技術的各種試験	特性値、予測パラメータを測る。環境試験、限界試験、非破壊試験、寿命試験、再現実験、DPA
機器分析	分解・切断、フラクトグラフィ、SEM、表面解析、有機・無機分析など
過去の情報の利用	データバンク、事例集、チェックリストなど設計から使用までの管理データ

と、技術者の行なう故障診断方法は大きく以下の2種類に分けることができる。

診断方法1 過去に出会ったことのないような故障の症状に対する故障診断は、設計結果から分かる対象の構造や機能の因果関係を辿ることにより実現されている。

診断方法2 自分が過去に同様の故障を扱ったことがある場合は、その過去の経験から因果関係を辿るプロセスを省略し故障を同定する。

さらに、診断方法1による故障診断作業は以下のように整理することができ、そのとき用いられる知識は表2.3のように整理できる。

1. 故障の症状(機能レベルの故障の症状)を同定する。
2. 故障の症状からそれを引き起こしている可能性のある挙動の変化の候補(挙動レベルの故障の症状)集合を導出する。
3. 挙動レベルの故障の症状の候補集合から候補を一つ選び出す。
4. 選んだ候補に関係あるユニット(現像部、搬送部、定着部など)と系(電気系、機械系、光学系など)を選択する。
5. 選択したユニット、系内で因果関係(エネルギー・フローや接続関係など)を辿りながら点検を行ない、故障の症状を引き起こしている故障を同定する。

ここで機能レベルの故障の症状とは、複写機の例でいえば「用紙が出ない」、「画像が汚い」等の人間にとっての機械の働きから見た故障の症状であり、挙動レベルの故障の症状は機能レベルの故障の症状に対応する物理的振舞いの変化であり、例えば、「ローラー3における用紙の停止」、「画像濃度の不良」等である。

表2.3: 故障診断に用いられる知識(診断方法1)[Umeda89b]

段階	用いた知識
1.	対象の動作知識 経験的知識
2.	対象の動作知識
3.	故障症状候補順位付け知識
4.	対象の動作知識 保全容易性知識
5.	対象の動作知識

2.1.3 修復作業の実際

修復作業は、「故障により必要な機能を失った機械に、何らかの操作、変更を加えることにより、喪失機能を回復させる作業」であると定義できる。

佐藤[Sato90]は、製鉄所の圧延工場において、技術者による圧延機の修復作業の現地調査を行なった。この調査の結果得られた修復処置の例を表2.4に示す。これらの

表2.4: 故障に対する修復処置の例[Sato90]

番号	故障	修復処置
1	パイロット圧用電磁弁リーク	予備電磁弁との交換
2	シリンダホース破損	予備ホースと交換
3	スケール詰まりによるロール不転	スケール掃除
4	フィールドロール軸受け部カバー外れ	電溶補修
5	ガイド用ロールへたり	電溶肉盛り
6	サイドガイドディスクホイール不転	ホイール取り外しダミープレート取り付け
7	電磁切り替え弁スティック	切り替え弁手動にて操作
8	腐食による給脂配管洩れ	ゴムによるテーピング
9	グリースポンプ切り替え弁不良	ホース、バルブによるバイパス化
10	油圧ポンプ破損	予備油圧ポンプへの切り替え
11	ガイド板変形	ガイド板除去
12	フィールドロール軸受け部劣化	軸受けへのオイル増し (オイル供給ボリューム位置操作)
13	刃先摩耗によるクランプシャカック切味不良	刃先クリアランス増し (距離調整)
14	シリンダピストン押えのボルト劣化進行	シリンダ背圧の下げ (減圧弁設定値ボリューム位置操作)
15	仕上げ圧延機6号のモータの一つ破損	粗圧延機3号での圧下量増し (設定値変更)

例から、修復操作は以下の二種類に大別することができる。

修復1 故障箇所の交換、補修、清掃等、修復後の機械が正常時の機械とほぼ同じ物理的構造、属性を持つもの。

修復2 修復後の機械が、正常時の機械とは異なる物理的構造、属性を持つもの。ここでは、冗長系を利用し、故障部品を冗長部品に切り替えて使用する場合も含めることにする。さらに、修復2は、以下の二種類に分類可能である。

修復 2a 故障した機械の上で操作可能なパラメータを操作し、機械の動作を変化させることによって、機械の機能を回復させるもの。

修復 2b 修復の際、機械の構造を変化させることにより必要な機能を実現させるもの。

表 2.4 の例でいえば、例 1～5 が修復 1 に、例 6～11 が修復 2b に、また、例 12～15 が修復 2a に相当する。佐藤 [Sato90] によると、全体の 7 割程度は修復 1 に分類される修復処置が取られるが、交換部品が無い、工場を止めたくない等必要な場合には修復 2 の処置が取られる。

以上のような修復作業は、図 2.3 のような形に整理できる [Sato90]。

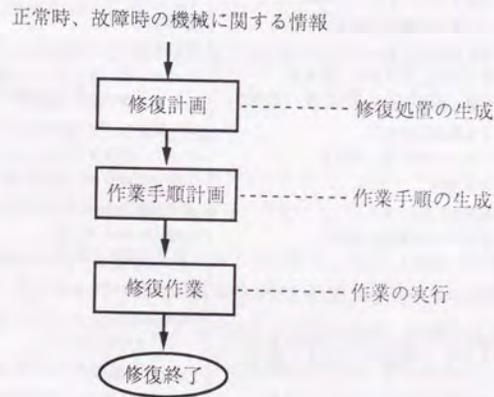


図 2.3: 修復のタスク [Sato90]

2.1.4 保全作業自動化のための課題

以上、第 2.1.2 項、第 2.1.3 項で、故障診断作業、修復作業の実際を述べた。これら、事後保全作業の特徴は以下のように整理できる。

まず、機械は図 2.4 のように表現することができる [Yoshikawa79a]。ここで、実体集合 (Entity Set) は全ての実体を要素とする集合であり、属性空間 (Attribute

Space) は実体の物理的性質である属性により位相を生成した空間、機能空間 (Function Space) は実体の使用目的を含んだ抽象的挙動である機能による位相構造を持つ空間である。

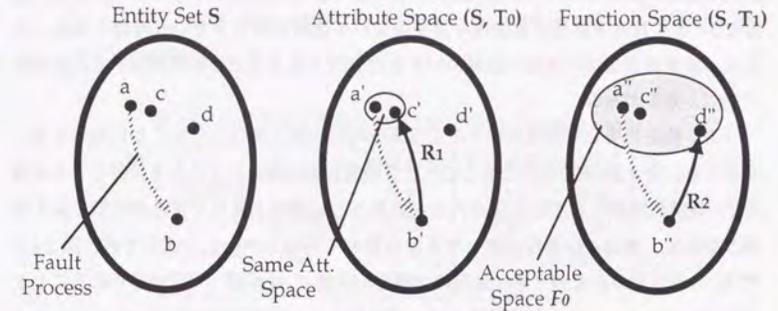


図 2.4: 機械の属性、機能表現

ある正常な機械 a は、属性空間ではその属性的特徴により a' と、機能空間では機能的性質により a'' と表される。このとき、機械 a は、突発故障、劣化故障等により状態が変化する。この変化した機械を b とすると、 b に対応する属性空間、機能空間上での位置は b' 、 b'' と表現される。機能空間上にはあらかじめ機械が正常であると判定される領域 F_0 が設計者、使用者により規定されており、この領域から機械の状態が出ることにより「機械が故障である」と使用者により認識される。例えば、複写機であれば機能を表す各位相 f_1 : making a copy, f_2 : copy quality, f_3 : copying speed、...として、 $F_0 = f_1 \cap f_2 \cap f_3 \cap \dots$ と表すことができる。 $b'' \in F_0$ ならば、機械 b は正常であると言えることができる。

機械が故障状態にあると使用者により認識された場合、故障の症状は機能空間上で b'' が発現しない機能の集合であり、故障は a' と b' の差異のうち、故障の症状を引き起こした属性変化の集合である。このように、故障は機能の変化 (故障の症状) に依存して規定されるものである。これは、ある機械に要求される機能やその正常状態 F_0 が設計者や使用者により決定されるため、故障も設計者や使用者により機能を通じて間接的に決定されることを示している。

このとき、故障診断は、機能空間上の変化から、属性空間上での機械の状態 b 、および、それを引き起こした物理現象を同定することである。後者の物理現象は故障メカニズムと呼ばれ、その同定は「故障解析」と呼ばれることもある[Shiomi91]。第2.1.2項によれば、このとき、図2.2に示したように様々な故障が考えられ、技術者は表2.2に示したように様々な知識や方法を用いて診断を行なうという特徴がある。さらに、診断の方法には既知の故障に対する経験的方法と未知の故障に対する因果関係に基づく方法がある。

一方、修復作業とは故障機械 b を正常領域内の状態に変化させることに相当する。このとき、第2.1.3項で述べたように、正常の属性的状態に戻すことを目的とする修復1と、機能的状態としては修復されるが結果として機械が異なる属性的状態を取る修復2がある。修復1は属性空間上で正常状態 a' を修復目的として実行され、図2.4上で R_1 と表される。このとき、修復後の機能的状態 d' は、 $d' \in F_0$ が得られることが仮定されている。これに対し修復2は、機能空間上で機械を正常領域内に存在させることを目的として実行され、図2.4上では R_2 と表される。この場合は、修復後の機能的状態 $d' \in F_0$ に対応する属性的状態 a' は a' と必ずしも一致しない。本研究では、以上の二つの修復戦略を文献[Anonymous86]に従い、前者を「属性保全」、後者を「機能保全」と呼ぶことにする。

故障診断作業、修復作業は知的判断部分の側面と実際に機械に対して働きかける動作部分の側面という二つの側面からなると考えることができる。知的判断部分は故障診断の中心的役割を果たし、また、図2.3に示した修復作業のうちの修復計画、作業手順計画は知的判断部分である。一方、動作部分としては、故障診断における点検作業や修復作業の中心的部分である部品交換、清掃、修理等の作業が含まれる。

また、吉川[Yoshikawa83]は、保全の自動化実現を難しくしている保全の特徴として以下の4点を挙げている。

1. 横断性

保全技術は特定の技術分野固有のものではなく、あらゆる分野で問題になる技術にも関わらず、保全技術は各分野の固有性と密接に関連しており、一般性を持つ理論が存在しない。

2. 多様性

故障は、多様なメカニズムによって発生するため、様々な分野の様々な技術を用いて診断しなければならない。これは表2.2を見ても明らかである。

3. 不定性

例えば制御技術ならば制御対象が既知のものとして存在するが、保全の場合、まず機械のどの部分を診断、修理するのかという対象の同定、すなわち、対象の構造や特性を同定することから始めなければならない。すなわち、制御の言葉でいえば、制御対象、制御特性、および、制御量が未知という意味で不定性が高い問題領域といえる(表2.5参照)。

表2.5: 制御技術と比較した保全の不定性 [Yoshikawa83]

	制御対象	対象特性	制御量
単純制御	既知	既知	既知
フィードバック制御	既知	既知	未知
適応制御	既知	未知	未知
診断・修理	未知	未知	未知

4. 非繰り返し性

故障が頻発する箇所は改良設計等の何らかの処置が取られることが多く、同種の故障が繰り返し発生するということは多くない。このため、従来的一般技術で用いられる経験・帰納とう方法が単純には適用できないという特徴を持つ。

以上、故障診断、修復の自動化を行なうための課題は次のようにまとめられる。

1. 保全作業は知的判断部と動作部の二つの側面から成っており、この両者を自動化しなければならない。
2. 保全の特徴である横断性より、保全作業自動化のための何らかの一般的理論が必要である。このとき、「機能保全」や「属性保全」という保全の考え方が有効である。
3. 知的判断部は、多様性、不定性、および、非繰り返し性に対応しなければならない。
4. 動作部は、知的判断部により導出される修復を柔軟に実現可能でなければならない。

2.2 保全の理論と問題点

2.2.1 保全の理論

保全技術に関連した学問分野は大きく分けて三種類の方向性がある。第一は、第2.1.1項で述べたような、保全作業の分類、各保全作業実施の方法論を述べた「保全工学」と呼ばれる分野(例えば、[Ichida68])である。第二は、機械の信頼性、保全性を高めるための設計方法を議論する「信頼性工学」(例えば、[Shiomi68])である。そして第三は、故障が発生するメカニズムを解明する「故障解析」の分野である。故障解析の分野では、故障物理(例えば[Shiomi70])等の分野が含まれており、また、表2.2で示したように、FTA[Sayama81]等の各種技法が提案されている。

2.2.2 保全性設計

機械が高信頼性、高保全性を持つように設計するための指針を与える理論が信頼性工学である。信頼性工学のアプローチは基本的に、各 부품の信頼性を信頼性試験等により統計的に予測、管理し、機械システム全体の信頼性を確保することを目的としている。このとき、信頼性とは信頼度、寿命、MTBF(Mean Time Between Failures)等の信頼性特性値により管理される。すなわち、ここでの基本的な手法は、故障を発生させないことを目的として部品品質の管理、システム全体の構成を評価することである。

信頼性工学の範囲内においても故障の発生を仮定した設計方針が幾つか存在する。その主要なものが機械に冗長性を付加するというものである。冗長性は、「規定の機能を遂行するための構成要素又は手段を余分に付加し、その一部が故障しても上位アイテムは故障としない性質」である[JIS87]。機械に冗長性を付加する方法としては、構成要素を並列に配置する並列冗長、予備系を付加する待機冗長、並列冗長を拡張し n 個の要素中 m 個が動作していれば系が正常に働くように構成要素を配置する m -out-of- n 冗長等がある[Nonaka90]。冗長系にはこのように幾つかの方式があるが図2.5に示すような等価な部品を並列に配置する並列冗長が基本となっている。図2.5は流体系におけるポンプの並列冗長系を示しており、一方のポンプAが故障した場合にも残りのポンプBにより動作可能であることを示している。冗長系の特徴は設計者が予測した故障に対しては有効に作用しシステムの信頼性を向上させることにあるが、欠点として、設計が不適切であるとシステムの複雑さを増大させ、かえって信頼性を

低下させたり、コスト、重量を不必要に増大させることが挙げられる。また、設計者が予想しないような故障に対しては冗長系は有効ではない。このため、冗長系の利用やその方式、および、実現方法の決定は設計者の経験や熟練技術に依存している。実際このような冗長系は人工衛星、原子力発電所、航空機等の高信頼性が要求される領域で利用されており[Nonaka90]、また、非停止性が要求される商用計算機の分野で近年一般的に利用されてきている[Watanabe86]。この他の設計戦略としてフェールセーフ設計やフルブールーフ設計がある。フェールセーフ設計は、「アイテムに故障が生じて安全性が保持されるように配慮してある設計」であり、フルブールーフ設計は、「誤操作に対してアイテムの信頼性・安全性を確保する設計」である[JIS87]。これら、フェールセーフ設計、フルブールーフ設計の実現方法も冗長系の付加と同様に各設計者の工夫に任されているというのが現実である。

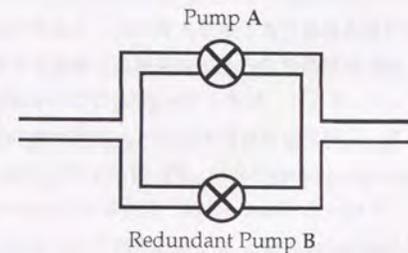


図2.5: 並列冗長系の例

2.2.3 保全の自動化手法

保全の自動化の研究も近年盛んに行なわれつつある。この分野の研究は大きく四種類の方向性に分けることができる。

(1) 知的判断の自動化

第一の分野は、知識工学、人工知能の発展にともない、保全の知的判断部分の知能化、自動化を行なおうとするものである。この分野の研究では故障診断や修復計画の

自動化の研究が盛んに行なわれている。

故障診断の自動化の研究は、主に経験的知識をプロダクションルールの形式で収集し、エキスパートシステムを構築する方法から始まり、現在は因果的な診断対象の構造、挙動の知識を用いたモデルベース推論に基づく方法に移りつつある。前者には、大型直流発電機の故障診断、プロセス制御用コンピュータ、列車電子制御系の故障診断等、幾つかの実用の域に到達しつつあるものもある[Arai87]が、知識獲得が困難であること、未知の故障に対して無力であること、知識ベース構築の問題、知識やシステムに汎用性が無い等の問題点が指摘されている[Umeda89a, Price88]。これらは、保全の特徴である非繰り返し性、多様性と深く関係する問題である。

これらの問題を解決するために後者のモデルベース推論を用いた知識表現、推論手法を用いた診断手法が数多く提案されている。伊理ら[Iri80]は、化学プラントを例とし、パラメータを点、パラメータ間の比例、反比例関係を符号付き有向枝として表した有向グラフで対象の構造を表現し、入力された状態量から異常伝播経路となり得る部分有向グラフの極大強連結成分を取ることで、故障を同定する手法を提案した。さらに、柘植ら[Tsuge84]はこの有向グラフを用いる手法を拡張、実用化した。この手法はモデルベース故障診断の基礎となる手法である。また、Genesereth[Genesereth84]は、設計情報を用いて電子回路を診断する方法を提案しており、Dagueら[Dague87]は、定性物理[Bobrow84, Weld89]に量概念(Order of Magnitude)を導入した故障診断手法を提案している。また、Hamilton[Hamilton88]は、定性物理を利用し、リアルタイムでヘリコプターの故障診断を行なうシステム HELIX を提案している。さらに、Abu-Hanna[Abu-Hanna88]は、経験的知識と因果関係に関する知識の両方を用いる故障診断方法を提案しており、山口ら[Yamaguchi88]は、物理的因果関係から診断用知識を自動的にコンパイルして用いる方法を提案している。これらのモデルベース推論の手法は、推論の組合せ的爆発を防ぐため単一故障仮説(Single Fault Assumption)、すなわち、対象系に最大一つの故障しか発生しないという仮定を採用している。これに対し、de Kleerら[de Kleer87]は、複数故障仮説(Multiple Faults Assumption)、すなわち、対象系に複数の故障が発生していることも考えるという仮定、を実現した GDE(General Diagnostic Engine)を提案しており、Reiter[Reiter87]は GDE と同様の故障診断手法を非単調論理により形式化、整理した。また、Strussら[Struss89]は、この GDE を改良し、物理的因果関係を明示的に用いて診断する方法を提案しているが、この方法では

単一故障仮説を用いている。de Kleerら[de Kleer89]も GDE を拡張し、各部品の変動の振舞いの変化を「挙動モード」として記述することにより、複数故障に対応し、かつ、物理的意味付けを用いた故障診断手法を提案している。

このように、故障診断作業の自動化手法としてモデルベース推論を用いるものが主流になっているが、西田[Nishida91]は、その主な推論手法を以下の四種類に分類している(図 2.6 参照)。

(a) モデルの因果解析に基づくもの

認識された因果構造に従って、例えば予測値と観測値のずれを制約として伝播させる。伊理らや Genesereth の方法がこのタイプである。

(b) 制約緩和型

故障が検出されると、故障候補となる部品に対応する制約を取り除いてシミュレーションする。モデルから予測される挙動と観測されている現象が合致するまでこれを繰り返し、故障部品を同定する。探索範囲を狭めるため、階層型モデルなどが用いられる。上記の HELIX がこのタイプである。

(c) GDE との組合せ

モデルから予測される挙動が観測と矛盾しない故障仮説の極小の組合せを計算する。上記の GDE や Struss らの方法、de Kleer らの方法がこのタイプである。

(d) 故障シミュレーション(全挙動予測)に基づく診断規則の生成

対象モデルにさまざまな故障仮説を加えた故障モデルの挙動を解析して、診断のための規則を自動的に生成する。上記の山口らの方法がこのタイプである。

一方、修復に関する知的判断部の自動化に関する研究は診断の自動化に比較すると少数である。これは、診断作業が知的判断が中心であるのに対し、修復作業は動作が中心となることに起因している。しかし例えば、篠原[Shinohara88]は、電車で故障が生じた場合に応急的な修復処置を提示するエキスパートシステムを開発している。ここでの応急処置は、スイッチ類の ON/OFF であり、これにより故障部分を切り離し、機能の回復、異常動作の排除等を実現することを目的としている。ここで用いられている知識は、乗務員等が利用する応急処置に関するマニュアルや乗務員の持っているノウハウ的な知識である。松本[Matsumoto83]は、電力系統の修復のための処置を生成するエキスパートシステムを開発した。ここでは、修復方法に関するノウハ

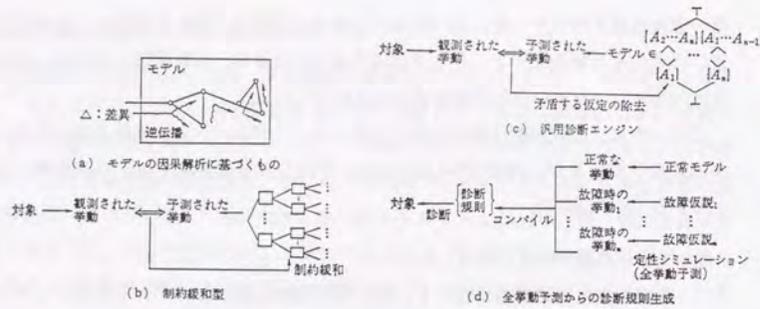


図 2.6: 故障診断システムの主なアーキテクチャ [Nishida91]

ウ的な知識の他に対象系の接続関係に関する知識を利用している。また、この手法では、修復処置の影響を潮流計算プログラムを用いて検証するという特徴がある。また、小林ら [Kobayashi90] は、予防保全に関してであるが、原子力発電所を例にして保全計画を立てる汎用的なシステムを提案している。

高津戸ら [Takatsuto90] は、実験原子炉を対象として異常発生時の運転支援システムを提案している。この手法では、警報を中心に対象をモデル化し、警報発生時の対処法を経験的知識を用いて導出する。熊野ら [Kumano90] は変電所を対象に時系列情報と故障確率に関する知識を利用した故障時の復旧を支援する運転支援エキスパートシステムを開発している。同様に、水上ら [Mizukami90] も一次変電所を対象として、変電所を対象を絞った専用推論システムによる運転支援システムを開発している。これら三者は実用性を狙ったものであり、フィールドテストを行ないつつ有効性を検証しているという特徴を持つ。

さらに、Crowら [Crow91] は Reiter [Reiter87] の故障診断理論を拡張し、新たに修復処置としての再構成を表す *rcfg* 述語を導入し、故障診断と修復措置としての対象系の再構成を統合化した手法 FDIR (Fault Detection, Identification, and Reconfiguration) を形式的に論じている。ただし、この手法は、「再構成」の形式化と形式的な推論手法について論じたものであり、実際の機械の再構成の方法や対象のモデル化の方法については議論していない。

一方、作業手順生成の自動化を対象としたものとしては、浅間 [Asama89] の研究、西沢ら [Nishizawa90] の保守作業の計画を支援する研究等がある。

(2) センサ技術の高度化

近年のセンサ技術と収集された情報に対する処理技術の発展により、従来は収集不可能であったような保全に有効な情報をリアルタイム、かつ、オンラインで収集することが可能になりつつある。例えば、アコースティックエミッションによる切削工具のチッピングやベアリングの摩耗の検出、半導体の雑音レベルからの半導体の劣化状態の測定が実現されている [Akizuki91]。さらに、表 2.6 に示すように機械の様々な劣化現象をモニタリングすることが可能になりつつある。

表 2.6: 代表的な劣化現象とその典型的な検出法 [Akizuki91]

劣化現象	典型的な検出法
漏れ	加圧、化学試薬、超音波、放射性同位元素
割れ	AE、放射線探傷、超音波、磁気探傷、渦流探傷
腐食	放射線、渦流探傷、超音波、電気抵抗
異常振動	振動ピックアップ
異常音	マイクロホン、騒音計
異常温度	熱電対、抵抗温度計、サーモクレオン、赤外線カメラ
材質劣化	超音波、X線
ゆるみ	振動ピックアップ
油劣化	フェログラフィ、SOAP
絶縁劣化	電圧降下試験、高圧法、 $\tan \delta$ 、部分放電

さらに上記に示すようなセンサ技術と情報処理技術やアクチュエータと複合化することによる「知能化センシング技術」 [Yamasaki89]、ニューラルネットワークを利用することにより従来は熟練技術者にしか検出できなかった変化の測定 [Watanabe89, Takemori89]、および、様々なセンサ技術を統合化することにより測定対象を正確、詳細に測定するセンサ・ヒュージョン等の様々な新しい手法が開発されつつある。

(3) 修復動作の自動化

第三の分野は修復動作の自動化である。修復作業の自動化の研究としては、知的判断部の自動化よりも、修復の動作の自動化をロボット等を用いて行なおうとする研究が盛んに行なわれている。これは、原子力発電所など修復作業を行なう場所が危険を伴うなどの自動化の必要性が高いことに起因する。例えば、浅間 [Asama89] は、6自由度ロボット・アームを用いて原子力発電所の保全を行なう保全ロボットを開発し

た。また同様の目的を持つ研究が1983年度から通産省の大型プロジェクト「極限作業ロボット」として盛んに行なわれている [Anonymous90]。

(4) 自律分散システム

前三者とは多少異なるものとして、自律分散システムの研究が挙げられる。この分野では、保全問題を中心課題としたものではないが、システムを自律的、分散的構成要素から構成し、その構造を動的に変更可能にする試みがなされており、その範囲内で構成要素の故障に対しシステムが強靱に対処可能な手法が研究されている。これは例えば、10個の同様な要素から成るシステムにおいて1個の要素が故障した場合は残りの9個の要素からシステムを再構成するといった考え方である。この分野で主なものは、計算機ネットワーク・システムを故障に対し強靱にする手法 [Watanabe86] 等がある。また、自律分散ロボット・システムの研究 (例えば、[Fukuda87, Matsumoto89]) が近年盛んになりつつある。

2.2.4 保全の理論の課題

以上、保全に関連する理論を整理したが、これらのみでは保全の自動化を実現することが困難であると考えられる。その理由は次のように整理することができる。

1. 保全工学における機械の保全手法は、主に技術者の経験、手法を体系化、整理したものが中心であり、各固有分野と密接に結び付いている。このため、保全の特徴である横断性に対応する理論的裏付けにはなりにくい。また、これらは技術者、設計者による作業を対象としており、自動化を目的とした場合、直接には基礎理論となりにくい。
2. 信頼性工学は、機械に故障を起こさないことを目的としたものが中心である。しかし、故障の発生を防ぐ、あるいは、全て予測することが不可能であるとすると、機械に故障が発生した場合において、全体として機械システムがどう振舞うべきか、また、そのための機械の設計方法論というものを考える必要がある。
3. 知的判断作業の自動化に関しては、保全の特徴である不定性、非繰り返し性、および、多様性に対応する必要がある、その意味でモデルベースト推論は有効

な手法であるといえる。ただし、各手法には様々な限界、実際問題への適用が行なわれていない等の問題が残されている。

4. 修復動作の自動化の研究は、動作部分の自動化という意味では有効な手法であるといえるが、保全全体の自動化を考えると、例えば、ロボットの保全対象の状態の認識、環境理解、修復目的の理解等、知的判断部の側面に関して十分な能力を持っているとは言えない。すなわち、保全作業や保全対象、および、その環境に対する理解の自動化があって初めて高度で柔軟な保全作業の自動化が実現可能になると考えられる。その意味で、知的判断部の高度化が必要である。
5. 自律分散システムの研究は修復動作をより高度化、柔軟化するための機械形態としての面と知的判断部の自律分散化という面の二つの側面を持つ。ただし、これらの自律分散システムの構築方法、運用方法に関する研究はまだ始まったばかりの段階である。

2.3 本研究の位置付け

本研究は、故障が発生した場合にも機能を発現可能な自己修復機械一般の設計方法を議論するものである。ただし、工場、プラントのような大規模な機械システムからホッチキス、バネ秤のような簡単な機械までの機械一般を議論の対象とすることは困難であるので、ここでは、人工衛星、複写機、工作機械、自動車といった一個のメカトロニクス機械システムを具体的な対象とする。そして自己修復機械の実現方法として、第2.1節で整理した特徴を持つ保全作業、特に、事後保全作業を機械上で自動化する方法を取ることにする。

一般に、図2.7(a)に示すように、設計作業は要求機能という設計者の概念を実現可能な概念への変換を行ないながら、最終的に具体的な実体(機械)に変換する作業である[Yoshikawa79a]。また、概念操作や概念の実体化に関する方法論が広い意味での設計方法論と呼ばれる学問分野である。

一方、保全作業は、図2.7(b)に示すように、運用、運転している実体(機械)があらかじめ与えられ、使用者が機械が異常であると認識した時点から始まり、故障した実体に関する概念を獲得し、概念操作を行ない、再び、正常な実体を作り出す作業である。すなわち、まず、保全作業者は点検、故障診断作業により、故障した機械から故障の症状、故障等の概念を獲得する。次に、あらかじめ作業者が持っている対象機械の正常状態に関する概念と比較しながら修復計画、修復作業手順計画等の概念操作を行ない、故障した実体への修復操作の方法(概念)を決定する。このとき、概念操作の戦略として「属性保全」、「機能保全」、もしくは、第3.1.3節で述べる「機能トレードオフ」等が関係する。最後に、修復操作(概念)を対象機械上で実現することにより修復を実行し、故障した機械を正常な機械に変化させる。図2.7(b)においても、各変換の段階を実行するための方法論が広い意味での保全工学の分野である。

ここで、保全作業と設計作業を比較した場合、保全作業における故障実体の概念獲得段階以降は、修復後の機械に関する概念を生成し、その概念を実体化するという形をしており、一種の設計作業と見ることができる。ただし、実体化の段階で故障した機械を利用しなければいけない。これは、一般の設計が新しい部品や機構を製作、利用できることと比較するとより拘束条件の強い設計問題であると言える。また、故障実体の概念獲得段階は設計と逆の操作であると考えられる。以上のように、保全は設計と共通部分を持ち、またそこで使用される知識やその構造が同様であるという意味

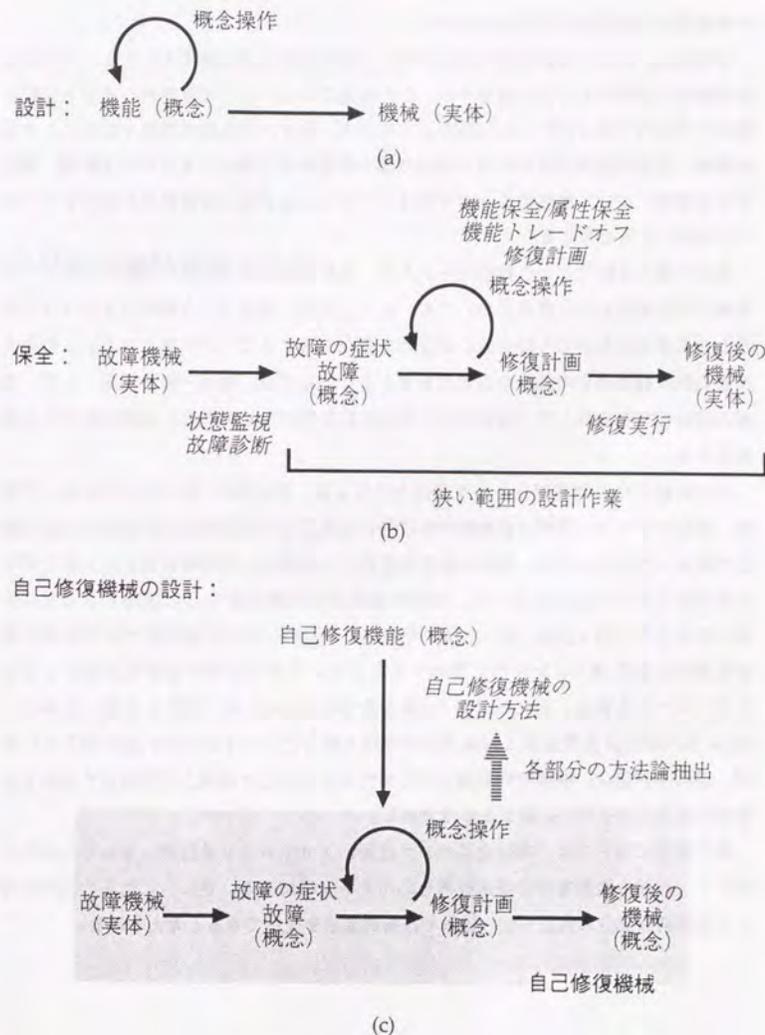


図2.7: 設計作業と保全作業

で問題領域に共通性がある分野である。

本研究は、以上の特徴を持つ保全作業を一個の機械上で自動化することにより自己修復機械を実現する方法を議論する。この議論により、自己修復機械の設計方法論を提案するものである(図2.7(c)参照)。すなわち、まず、自己修復機械を目的として故障診断、修復計画等の図2.7(b)の保全作業の各段階を自動化するための方法論、利用する知識等について整理する。本研究は、この方法論が自己修復機械を設計するための方法論となると考える。

保全作業の自動化という観点からいえば、本研究は知的判断部分や動作部分という各部分の自動化という意味においても、また、設計、運用という機械のライフサイクルからの意味においても統合的に保全の自動化を考えることが可能になる。このような統合的な観点からの保全の自動化を考えることにより、機械一般を対象として、機械の設計戦略等に関して「横断的な」保全の自動化のための理論を提案することを目的とする。

知的判断部分と動作部分という分け方で言えば、第2.2節に述べたようにセンサ技術、保全ロボット、自律分散機械等の研究の進展により動作部分の自動化の実現可能性が高まっているものの、機械の保全を目的とした場合、知的判断部分の自動化が不十分であるという認識に基づき、知的判断部分の自動化を中心に議論することにする。このとき、第2.2節に述べた課題を解決するため、保全作業自体や保全対象に関する意味を計算機上に記号的に表現することによりその理解や判断を実現するというアプローチを取る。例えば、第2.2節に述べたCrowらのFDIRと比較した場合、Crowらの手法は故障診断と対象系の再構成を統合化した手法を形式的に論じているが、逆に本研究は、機械の計算機上でのモデル化の方法や機械上で再構成を実現する機械の設計方法を中心に論じるものである。

動作部分に関しては、本研究はセンサ技術、メカトロニクス技術、さらには保全ロボット、自律分散機械等の要素技術を応用する立場にある。逆に、これらの技術に対して知的判断部分の実現手法の提案や技術的要求を提示できると考えられる。

第3章

自己修復機械実現方法論

本章では、自己修復機械の実現方法を議論する。まず、自己修復機械の実現方法について幾つか述べ、本研究における自己修復機械の実現方法と比較する。次に、本研究の自己修復機械において重要な要素である知的判断部と機械部の仕様と実現方法について述べる。

3.1 自己修復機械の枠組

3.1.1 自己修復機械に必要な機能

本研究では、自己修復機械を「機械内部に故障が発生した場合にも、何らかの方法で修復を行ない、要求される機能を発現することが可能な機械システム」と定義する。このような自己修復機械は本来要求される機能(例えば、複写機は「コピーを取る」、自動車は「走る」等)以外に次の五種類の機能を発現する必要がある。また逆に、機械において以下の五機能を実現できたときに、自己修復機械を設計できたとする。

状態監視機能

自己の状態を監視し、状態を把握する機能。

故障判定機能

把握した状態をもとに、自己が正常状態であるか異常であるかを判定する機能。このとき、一般的には自己の機能に関する知識が必要になる。

故障診断機能

自己の状態が異常の場合には、その原因となっている故障を同定し、また、故障したシステム全体の状態変化を把握する機能。この機能は、保全の特徴である不定性、非繰り返し性、および、多様性に対応可能でなければならない。

修復計画機能

故障診断の結果に基づき、修復計画を生成する機能。この機能も上記の保全の特徴に対応可能でなければならない。

修復実行機能

修復計画機能により計画された修復処置を実現し、故障を実際に修復する機能。本研究では、図2.3で示した作業手順計画も広い意味でこの機能に含まれると考えることにする。

3.1.2 本研究における自己修復機械の実現方法

このような機能を持つ自己修復機械を実現する方法は、幾つか考えることができる。例えば、第2.2.3項で述べた保全ロボットは修復実行機能の実現を中心としたアプローチであり、人工知能によるエキスパートシステムのアプローチは故障診断、修

復計画機能のみの実現である。また、自律分散システムは、現在その実現性は必ずしも明らかではないが、自己修復機械の一実現形態であることが出来る。さらには、自律分散システムを発展させ、生物のように再生機能をも含む生物型機械を考えることもできる [Yoshikawa79b]。

本研究では、人間によって行なわれている保全作業を機械上で自動化することにより自己修復機械を構築するが、この自己修復機械を完全に実現することは現状の技術では不可能である。そこで、以下に述べる二つの限定を導入することにする。

時間的限定

自己修復機械の「自己修復性」に時間的な限定をつける。すなわち、永遠に自己修復性を発揮する機械を実現することは不可能である。そこで、一定期間自己修復性を発揮する機械を考える。そして、一定期間後に技術者による修復を受けることができるものとする。この一定期間というのは、人工衛星のように非常に長期間の場合も考えられるし、複写機のように2、3日程度のものも考えられる。

属性保全の有限性

ある機械に関して予備部品が無数に用意され、あらゆる故障に対して部品交換、清掃、補修等の修復作業が全て自動的、かつ、十分に短時間に行なうことが可能であれば、理想的な自己修復機械を実現することが可能になる。しかし、不定性、非繰り返し性、多様性という保全の特徴を考慮に入れた場合、技術的、時間的、経済的限界からこのような理想的自己修復機械を実現することは不可能である。

以上の限定を満足する自己修復機械は、従来の属性保全という観点から見た時に完全に修復が行なえない場合においても、一定期間、何らかの方法により必要な機能を発揮し続ける機械である。これは、従来の機械が一つの小さな部品の故障によっても動かなくなる場合が多い(故障に対する脆弱性)のに対し、故障に対する強韌性を持つ「柔らかく壊れる機械」と呼べるものである。本研究では、この柔らかく壊れる機械を次項で述べる機能保全という保全の戦略に基づき実現する。

第2.1.3項で述べたように、実際の工場においても、予備部品が無い、緊急に機能の回復が必要な場合には、属性的には完全には初期状態に戻らない修復も行なわれており、またそれが緊急時には非常に重要になってくることが多々あると考えられる。

例えば、属性保全の実現性が非常に限定されている人工衛星には機能保全が適している。これらの理由により、「機能保全」の自動化の理論的整理とその実現手法の開発は重要であると考えられる。しかしながら本研究は、例えば保全ロボットによる部品交換のような属性保全の自動化を否定しているのではなく、属性保全が可能な範囲では属性保全による修復を実行することが有効であるが、属性保全の手段が尽きた時、対応できない故障が発生した時に本研究で述べる機能保全が有効であるという立場を取る。

身近な例として、週末に複写機が動作しなくなった場合を考える。この場合、月曜日までサービスエンジニアは来られないとすると、複写しなければならぬ重要な書類があるにも関わらず、複写機は使えない。このような場合、「自己修復」複写機はエンジニアが来るまでの間、少し遅くなったり、画質が多少落ちたとしても、何らかの方法でその必要機能「コピーを取る」ことを可能にする機械である。

3.1.3 機能保全と柔らかく壊れる機械

17ページで示した図2.4では、機能保全は R_2 として示されている。機能保全の目的は $d' \in F_0$ を満たすような機械の属性的状態 d' を発見し、かつ、状態変化を起こし d' を実現することである。

このような機能保全が機械において可能になるのは、機械の状態と機械の機能が一対一に対応していないからである。この点に関しては、第4.3節で詳しく議論するが、本研究では、機能を「実体の挙動を人間が抽象化した主観的表現」と定義する。例えば、「ものを飛ばす」という機能は、火薬の爆発やバネによる反発力を利用する方法等様々な挙動により実現可能である。このため、要求機能の集合 F_0 を発現させることを修復目標とすると、 F_0 を満足する初期状態 s_0 以外に複数の状態 s' が存在し、利用できる可能性がある。このため、属性保全の能力が限定された自己修復機械では、属性保全よりも修復実現の可能性が高い「機能保全」が適している。

しかし一般には、必ずしも F_0 を全て満たす状態を発見できない、もしくは、その機能的状態変化に到達不可能な場合も多い。一方、故障発生時には、使用者の要求、使用環境、緊急度等により要求機能の集合 F_0 の中に、必ず発現しなければならない機能(必要機能)とそれ以外の機能があると考えられる。そこで、 F_0 の中から必要機能を選択し、目標機能空間を $F'_0 (F_0 \subset F'_0)$ としてやれば、機能保全実現の可能性が高

くなる。例えば、複写機が故障した場合には、複写速度が低下してもコピーが取れるほうが好ましいし、自動車がオーバーヒートした際も安全な位置まで低速でも移動可能であることが望まれる。このような、必要機能の選択は使用者の状況や価値判断に依存するもので、ここではこの過程を「機能のトレードオフ」と呼ぶことにする。自己修復機械は柔軟に修復を実現するために、使用者と相談しながら機能のトレードオフを実行し、それに応じて修復を実現する必要がある。

機能トレードオフを行わない属性保全による自己修復機械と機能トレードオフを実現した自己修復機械の故障への対応の違いを模式的に表したものを図3.1に示す。

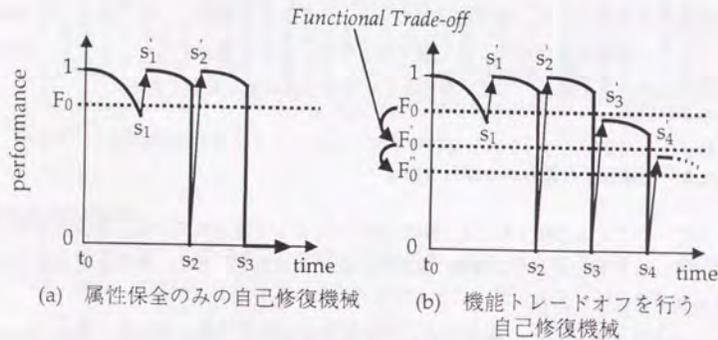


図3.1: 属性保全型自己修復機械と機能トレードオフ型自己修復機械

図3.1(a)は、機能トレードオフを行わない自己修復機械のライフサイクルを示しており、横軸は機械の寿命までの十分に長い時間尺度を、縦軸は機械の機能の発揮度合を示している。このとき、この自己修復機械は時刻 t_0 において製造され、その時点での機能は発現されるので¹、その性能を1と表すことにする。劣化が進行すると性能は低下し、必要機能 F_0 を発現できなくなると機械は故障状態であると判定され、属性保全により修復が実行される。図中では、この過程を $s_1 \rightarrow s_1'$ と表している。また、突発故障が発生したとしても(状態 s_2)、修復可能であれば修復を実行し、機械は機能を発揮し続けることができる(状態 s_2')。しかし、修復可能でない故障が発生した場合

¹ここでは、簡単のため初期故障を考えない。

は(状態 s_3)、機械は多くの場合全ての機能を発揮することができなくなってしまう。このように、対応不能の故障が発生した場合機械の性能が非連続的に性能を低下させてしまう性質は、故障に対する脆さ、もしくは、Hard Failと呼ばれる。

一方、図3.1(b)に示す機能トレードオフを行なう自己修復機械は、機能の回復が可能な場合には、上記(a)の自己修復機械と同様に属性保全、機能保全により修復を行なう(例えば、 $s_1 \rightarrow s_1'$)。しかし、故障状態 s_3 において必要機能の集合 F_0 を全て満たす修復を行えない場合には、知的判断部とユーザにより機能トレードオフを行ない、必要機能集合を設計時の F_0 から F_0' に変化させる。図3.1(b)上でこの機能トレードオフは、閾値 F_0 の変化として表現できる。この機能トレードオフの結果を用いて修復を実行した後の状態は例えば s_3' のように表現され、この状態においては対象機械は使用者にとって十分満足できないまでも、必要な機能を発揮することができる可能性がある。このとき、第2.1.4項で述べたように、機械の故障や故障の症状は設計者や使用者の期待する機能により規定されるため、機能トレードオフに従い同一の対象機械に対する故障や故障の症状の定義も変化する。この機能トレードオフ、修復という過程を繰り返すことにより、自己修復機械は、上記(a)の自己修復機械と異なり、故障に対してHard Failせずに、徐々に必要機能を満たさなくなるという柔らかく壊れる機械を実現可能になる。この過程は、必要機能を満足不能になった時点で終了するが、ここに至るまでの時間が十分長ければ、前節で述べた時間的限定を持つ自己修復機械の要求を満足する。

本研究では、この機能トレードオフ過程を含めた機能保全を修復の戦略とする。

以上で述べた自己修復機械を実現するために、本研究では、図3.2に示すような基本構成を取ることにする。すなわち、知的判断部として、知識工学的手法を用いた推論システムを構築することにより「故障判定機能」、「故障診断機能」、機能保全に基づく「修復計画機能」を実現する。機械部は、柔軟に「修復実行機能」を実現し、センサを配置することにより「状態監視機能」を実現する。知的判断部の実現方法は第3.2節で、機械部の実現方法は第3.3節で述べることにする。

3.1.4 他の手法との比較

以上の、本研究の目的とする自己修復機械、すなわち、「柔らかく壊れる機械」とその他の自己修復機械実現のアプローチ、および、高信頼化手法の特徴を表3.1に示

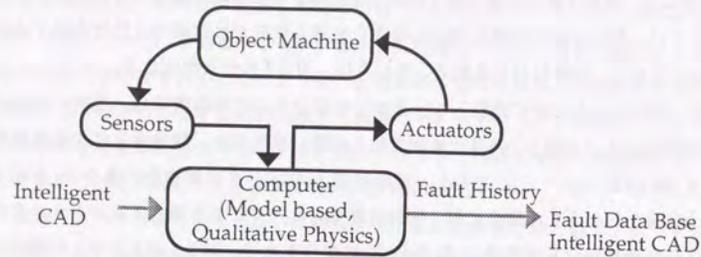


図 3.2: 自己修復機械の基本構成

す。ただし、これらの手法は背反するものではない。例えば、部品の高信頼化は全ての手法で必要なことであるし、冗長部品の利用や保全ロボットも修復実行機能の自動化、高度化という意味では、柔らかく壊れる機械の一部分として利用することが可能であり、必要である。

表 3.1: 高信頼化/自己修復機械の各手法

	既存機械の利用	部品交換	保全形態	知的	利用度
部品の高信頼化	×	×	しない	×	◎
冗長部品の利用	×	×	属性的	×	○
保全ロボット	○	○	属性的	△	△
柔らかく壊れる機械	×	×	機能的	◎	×
自律分散システム	×	×	機能的	?	×
生物型機械	×	○	属性的/機能的	?	×

表 3.1 の各項目は次のような内容により分類される。

- 「既存機械の利用」は、設計時に安全性を主に考えるか、もしくは、既存の機械に適用することを目的とするかという分類である。保全ロボットは、例えば既存の原子力プラントに適用し、危険箇所での技術者の代替する点で他の手法と異なっている。
- 「部品交換」は、故障発生後に機械の部品交換、もしくは、それに相当するような清掃、修理を行なうかという分類である。保全ロボットはこの点を主目的

としており、また、生物型機械はその定義として再生という形での部品交換を含むと考えられる。本研究は、属性保全の有限性に対する柔軟な自己修復性の実現を目的とするため、部品交換を直接の対象とせずに修復を実現するという立場を取る。また、自律分散システムは構成要素の再構成により機能の発現を行ない、部品交換を必要としないものである。

- 「保全形態」は保全の戦略として機能保全を行なうか、属性保全を行なうかという分類である。柔らかく壊れる機械や自律分散システムは修復後の形態が機械の初期状態と異なり得るという点で機能保全を実現する。
- 「知的」は、機械が知的判断部を持つか、持たないかという分類で、特に、保全の特徴である不定性、非繰り返し性、多様性に対処可能かという点を中心に分類している。この場合、柔らかく壊れる機械はこの性質が目的の中心の一つとなっている点が顕著であり、保全ロボットの研究には本研究でいう知的判断部の研究が必ずしも含まれないという点で中間的な立場である。逆に、自律分散機械や生物型機械は反射神経や抗原・抗体反応のように知的ではないが高度な修復機能を持つ可能性がある。
- 「利用度」は、現実の世界で利用されているか、いないかの分類であり、部品の高信頼化と信頼性に対する要求が高い部分における冗長部品の利用が一般的である。

3.2 知的判断部の実現方法

知的判断部の要求項目は以下の通りである。

1. 故障判定機能、故障診断機能、および、修復計画機能を実現すること。
2. 保全の特徴である不定性、非繰り返し性、および、多様性に対応可能であること。
3. 機能保全に対応し、機械部で実行可能な修復計画を導出可能なこと。

本研究では、1.、2.の要求を満たすために定性物理を用いたモデルベースト推論に基づく故障診断/修復計画システムを構築する。第2.2.3項で述べたように、モデルベースト推論を用いると、2.の保全の特徴に以下のように対応可能である。

不定性

保全においては、故障により変化した対象系がどのような構造、特性になっているのかを同定しなければならない。定性物理は物理的世界を表現することによる対象系のモデリングの過程をも含んでおり、記号的に因果関係に基づいて記号的に推論を行なうことにより合理的に対象構造、特性の同定が可能になる。

非繰り返し性

保全対象に依存しない形で十分な量の原理原則的な物理法則に関する知識を記述することにより、物理法則を組み合わせ、対象系が引き起こす可能性のある振舞いを記述された知識の範囲で全て導出することができる。この導出された振舞いの中には、対象系では実際には未発生の故障発生現象も含まれていると考えられる。この意味で、十分な量の汎用的な原理原則的知識の収集により、非繰り返し性に対応可能である。

多様性

定性物理という枠組を利用することにより、機械、電気、流体等の分野によらず物理的世界を統一的に記述することができる。

また、1.故障判定機能や3.機能保全に対応するためには、機械の「機能」の明示的なモデリングが重要な鍵となる。なぜならば、故障の判定は機能が満たされているか、否かにより判定され、機能保全は機能を満たすことを目標として修復を行なうからである。しかし、機能を含めた機械のモデリング技術は現状では十分に発展

しているとはいえない。そこで本研究では、第4章に述べるようにFBS(Function-Behavior-State)ダイアグラムを提案し、機能を含めた機械のモデリングを実現することにする。

以上により上記の自己修復機械の知的判断部に必要な機能を図3.3に示すように実現することができる。具体的な診断、修復計画手法については、第5章、第6章で議論する。

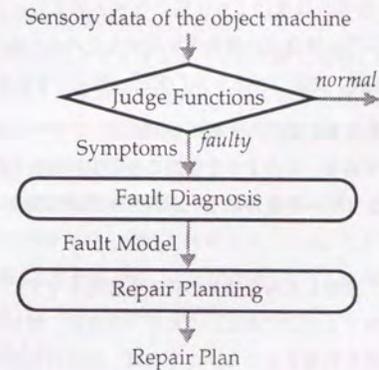


図 3.3: 知的判断部のアルゴリズム

3.3 機械部の実現方法

第3.1.1項で述べたように、機械部は本来要求される機能の他に、状態監視機能、および、修復実行機能を実現しなければならない。さらに、修復実行機能は、機能保全を実現可能な柔軟なものでなければならない。

このとき、本研究が対象とする一つのメカトロニクス機械システムにおいて、一般に状態監視機能はセンサ情報の収集により実現できると考えられる。また、保全ロボット等による部品交換以外に修復実行機能を実現するための自動化可能な操作としては、属性値、状態量の変更、制御シーケンスの変更、構造の変更が考えられる。

機械の属性値、状態量を変更する最も一般的な方法は、アクチュエータにより機械の状態量を制御することである。このような種類の修復実行機能を持つ自己修復機械を「制御型自己修復機械」と呼ぶことにする。制御型修復実行機能については次項で述べる。

制御型自己修復機械は、故障した対象機械の構造を変化させずに、柔軟な制御を行なうことにより機能を回復する方法である。これを一歩進め、機械自らがその構造を変化させることにより修復を実現することができれば、修復可能範囲を広げることができる。本研究では、構造変更を行なうため戦略として従来の冗長系を拡張した「機能冗長」を提案する。機能冗長型修復実行機能を持つ自己修復機械について第3.3.2項で議論する。

さらに、構造変更をより柔軟に実現できる機械の形態として、多数の細胞から構成される生物の形態のように多数の同質の部品が接続して機械システムを構成する自律分散型機械のようなものが考えられる。本研究では、これを「細胞型自己修復機械」と呼ぶ。このような機械に故障が起きた場合、システム全体を再構成し本来の機能を発揮させることが可能になる。

本研究では、細胞型自己修復機械については今後の課題とし、これ以上議論を行なわないが、本研究で提案する手法は細胞型自己修復機械を実現するためにも有効な手法であると考えられる。小池らはこのような細胞型自己修復機械の構成方法について議論し [Koike91]、また、分散した知的判断部の実現方法を提案している [Koike92b]。

3.3.1 制御型自己修復機械

制御型修復機械は機械の構造変更を行わずに状態量を制御することで、ある種の故障を修復することが可能である。

例えば、複写機においてドラムの劣化により正常な濃度のコピーが取れなくなった場合、ハロゲンランプの光量を調節することにより、正常な濃度を得ることができるとする。また、メイン・チャージャや転写チャージャを利用しても機能の回復を実現することが可能である。

この制御型修復実行機能の設計は、本研究の対象とする一つの機械においては、アクチュエータの配置問題とアクチュエータの制御で有効に機能を回復可能な構造の決定問題になると考えられる。この方法については第7章で議論する。しかし、対象が例えば工場やFMSのような機械システムの形態においては、修復実行機能の設計問題は別の形になる可能性もある。

ここで、一般の制御技術と制御型自己修復機械の違いについて考えてみる。一般に、制御対象系に制御点が m 個 (これを i_1, i_2, \dots, i_m とする)、出力である制御目的パラメータが n 個 (これを o_1, o_2, \dots, o_n とする) 存在した場合、制御技術は式 (3.1) に示すように、制御点と制御目的間の伝達関数 f を安定性、応答性等を考慮しながら決定することに相当する。

$$(o_1, o_2, \dots, o_n) = f(i_1, i_2, \dots, i_m) \quad (3.1)$$

吉川 [Yoshikawa83] は、制御技術と保全の特徴の違いを表2.5のようにまとめており、この違いが「不定性」の原因になっているとしている。すなわち、一般に、機械に発生する故障はその取り扱いの困難さに基づいて次のような二種類に分けることができる。

1. 構造変更を引き起こさない故障

劣化現象等によって引き起こされる故障は対象機械の特性を連続的に変化させるが、直接的には対象の構造変更を引き起こすことはない。上記のドラムの劣化による感度不良はこの例である。

2. 構造変更を引き起こす故障

劣化現象が進行した場合や偶発的な現象により、機械の構造変更を引き起こす故障が発生することがある。例えば、電球切れ、ギア割れ、導線のショート等

である。さらにこの種類の故障は、故障診断の困難さに基づいて二種類に分けることができる。

(a) 構造の欠損を引き起こす故障

電球切れ、ヒューズ切れのように、設計時に決められた機械の構造の一部が欠損する故障。

(b) 構造の生成を引き起こす故障

導線のショートや機械部品の干渉のように、設計時に決められた機械構造が変更され、新たな構造部分が生成される故障。この種の故障は計算量の問題から前項の故障よりも診断が困難である。

以上の分類のうち、劣化故障は機械の構造は変化させないが特性を変化させるため、伝達関数 f の形は変化しないが、各係数に相当する特性が変化する。表 2.5 に示すように、このような伝達関数の変化はフィードバック制御では扱えないが、適応制御では扱うことができる。

しかし、構造変更を引き起こすような故障は適応制御でも扱うことが難しいと考えられる。すなわち、構造変更を引き起こす故障は伝達関数 f 自体を変化させてしまう。本研究の手法は制御技術と異なり、故障の状況に応じて伝達関数、制御点、制御ループを動的に選択する。すなわち、第 5 章で述べるように故障診断機能により、構造変更を引き起こす故障が発生した場合を含めて、故障後の対象系の構造、特性を同定し、修復計画機能によりこの故障した対象系、および、制御可能なアクチュエータに応じて、制御点と各制御点に対する目的関数を決定する。これにより、予想されていない故障や制御対象の特性が大きく変化する故障等の制御技術では対処不可能な状況にも対処可能性があると考えられる。

すなわち、自己修復と制御技術の相違は、取り扱い可能な対象上の変化の大きさが、制御よりも大きく、かつ、種類も多くなるとはならないという点で、制御型自己修復に制御技術をそのまま適用できないと考えられる。

例えば、15 ページで示した表 2.4 の 15 番の故障事例について考えてみる。佐藤 [Sato90] によれば、この例は以下のようなものであった。製鉄所における熱間圧延工程における圧延スタンドは図 3.4 に示すように粗圧延機 3 台、仕上げ圧延機 6 台から成る。15 番の故障事例では、このうち最終の仕上げ 6 号圧延機のモータが焼き切れ、圧延機の圧下量が低下した結果、製品としての板の適正な厚みを得られなくなったという故障の状況である。これに対し、応急処置として粗 3 号圧延機の圧下量を増

加させるように設定値の変更が技術者の手で行なわれ、操業を続けた。この例に示すような修復は、通常の制御の範囲を越えているという意味で制御技術とは異なる技術である。本研究の制御型自己修復機械は、操作としてはアクチュエータによる制御量の変更を行なうが、柔軟な推論を行なうことにより有効な操作を故障の状況に応じて導出し、この例のような修復を実現するものである。

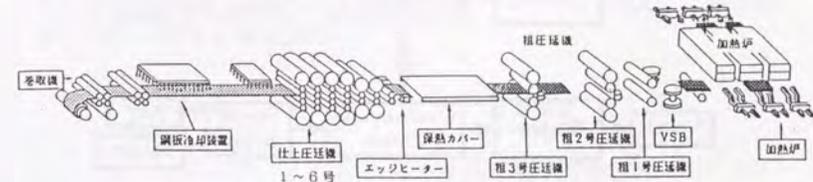


図 3.4: 熱間圧延工場の概略 [Sato90]

3.3.2 機能冗長型自己修復機械

第 2.2.2 項で述べたように、一般に、冗長系は同様の機能を持つ手段を余分に用意することにより、機械全体としての信頼性を高める目的で付加されるが、実際には同等の部品を並列に複数配置することにより実現されており、これを本研究では「部品冗長」と呼ぶことにする。部品冗長は、設計者が予想した故障には有効に作用するが、その結果重量、コストの増大を招き、また、逆に信頼性を低下させる要因となることもある。

そこで本研究では、部品の潜在機能を用いて、必要最低限の部品の増加でより多くの冗長性を機械に付与する方法を考える。ここで、潜在機能とは、「ある実体が、現在発現させておらず状況が変われば発現する可能性のある機能」のことである [Yoshikawa79a]。例えば、マニュアルトランスミッションの自動車は通常、エンジンの出力により走るが、エンジンが作動しない場合にはスタータ・モータにより移動することができる。これは、図 3.5 に示すように、駆動系の構造を変更することによりスタータ・モータに本来の機能「エンジンを始動する」の代わりに潜在機能「車体を動かす」を、エンジンに潜在機能「駆動力を伝える」を発現させて実現しており、部品数を増加せずに冗長性を付与している。ここでは、このように潜在機能を利用して冗長性を付与することを「機能冗長」と呼ぶ。部品冗長を用いて同様の故障の状況に

対処することを考えた場合、予備のエンジンをあらかじめ自動車に付加しておき、エンジンが故障した場合には予備のエンジンに切り替えて利用する方法が考えられる(図3.6参照)が、機械の複雑さ、コスト、重量は機能冗長に比較すると不利である。

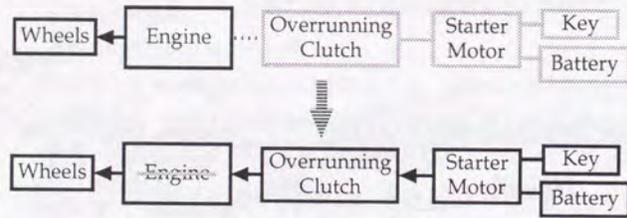


図 3.5: 機能冗長な自動車の例

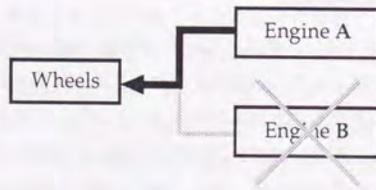


図 3.6: 部品冗長な自動車の例

部品冗長が故障を属性的に等価な冗長部品で置き換えるという属性保全を行なっているのに対し、機能冗長は失われた機能を機能的に等価な系により代替するという機能保全を実現している。また見方を変えると、機能冗長は機械の構造を変更し、機能を再配置し、機械を動作させているということもできる。すなわち、ある機械 a は正常状態でその機能 $F_0 = \cap f_i (i = 1, \dots, n)$ を満たしているとする。このとき、この機械 a は各機能 f_i を発現している構成要素 e_i から構成されている、すなわち、 $a = \sum_{i=1}^n e_i$ と考える。このとき、構成要素 e_i は、必ずしも部品と対応しておらず、また、異なる構成要素が実体としては同一であることも考えられる。故障が発生すると、例えば、構成要素 e_k が f_k を発現しなくなる。このとき、機能 f_k を一つ以上の構

成要素 $e'_k, e''_k, \dots \in \{e_1, e_2, \dots, e_i, \dots, e_n\} (i \neq k)$ で発現することができれば機能冗長性を実現可能である。ただし、一般には e'_k, e''_k, \dots により喪失機能 f_k を発現させることにより、 e'_k, e''_k, \dots が本来発揮していた機能 f'_k, f''_k, \dots は f_k とは同時に発現できない可能性があり、機能トレードオフの結果、 f'_k, f''_k, \dots が f_k よりも重要でないことがこの機能冗長系を利用可能な条件になる。以上によって、機能冗長系発揮後の機械 $a' = \sum e_i (i \neq k)$ は、 f'_k, f''_k, \dots 以外の要求機能 $F'_0 = \cap f_i (i \neq k', k'', \dots)$ を発現可能にすることができる(図3.7参照)。

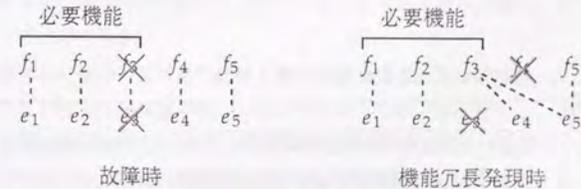


図 3.7: 機能冗長による機能の再配置

このような機能冗長を実現することにより、部品冗長が重量、コストの増大が避けられないが特定の故障に対しては信頼性を高めることが可能なのに対し、機能冗長は重量等の増大を抑えつつ、機能保全を実現することにより機械内外部の状況に柔軟に対応できる性質を持つ可能性がある。以上の部品冗長と機能冗長の相違を表3.2にまとめる。

このような機能冗長型自己修復機械は、メカトロニクス機械のようなコンピュータにより動作を制御する機械では、ソフトウェア上で制御パターンを変更することにより柔軟に構造変更を実現できるので、充分実現可能である。機能冗長系は故障の状況に応じて機械自らが柔軟に導出、実現することが理想であるが、本研究では、設計者が自己修復機械設計時に、機能冗長系を付加することにする。このとき、機能冗長型機械を設計するためには、機械の機能を直接表現可能で、かつ、機能と属性の関係や物理的実現性の評価が可能な設計支援手法が必要になる。このような機能冗長系の設計方法については第8章で議論することにする。また、機能冗長系を利用した修復計画の方法については、第6章で議論する。

表 3.2: 部品冗長と機能冗長

部品冗長	
特徴	<ul style="list-style-type: none"> ・故障した属性を冗長部品の属性で置き換える「属性的保全」。 ・特定部位の特定故障モードに対する信頼性を確実に高められる。 ・設計、信頼性の予測が容易。
問題点	<ul style="list-style-type: none"> ・重量、コストが増大。 ・設計時に予測した故障にのみ対応。
機能冗長	
特徴	<ul style="list-style-type: none"> ・失われた機能を他の部品の潜在機能で置き換える「機能的保全」。 ・必要最低限の部品増加で、より多くの冗長性を付加可能。 ・機能レベルの冗長性を考えることで、故障に柔軟に対応。
問題点	<ul style="list-style-type: none"> ・設計方法、信頼性予測方法が未整備。 ・機能判断、価値判断の必要性。

3.4 第3章のまとめ

本章では、自己修復機械の枠組とその実現方法について議論した。

まず、自己修復機械を実現するために、状態監視機能、故障判定機能、故障診断機能、修復計画機能、および、修復実行機能を実現する必要があることを述べ、本研究では、これらの機能を実現するために知的判断部と機械部の実現方法を議論することを述べた。さらに、本研究では自己修復機械を実現するために、時間的限定、および、属性保全の限界という二つの仮定を導入した。この意味で、本研究の自己修復機械は、様々な自己修復機械の実現方法のなかで「柔らかく壊れる機械」と呼べる実現方法とすることができる。

知的判断部は、保全の特徴に対応し、故障判定機能、故障診断機能、および、機能保全に対応した修復計画機能を実現しなければならない。本研究では、この要求を満たす知的判断部を実現するために、まず、機械のモデリングの方法について第4章で議論する。このとき、機械の機能の表現が故障判定機能、機能保全の実現のために重要な問題となる。また、故障診断機能、修復計画機能を実現するために、定性物理を用いたモデルベース推論手法を利用することにする。第5章、第6章でそれぞれ、故障診断の自動化手法、修復計画の自動化手法について述べる。

機械部は状態監視機能と機能保全に基づく修復実行機能を実現しなければならない。ここでは、その実現方法として制御型自己修復機械、機能冗長型自己修復機械、および、細胞型自己修復機械の三種類が考えられることを述べた。機能冗長型は修復方法として機械の構造を変更することにより、制御型よりも広範囲の故障に対処できる可能性がある。第7章、第8章で制御型、機能冗長型の機械部の設計方法、および、実験機での実験結果について議論する。

本章では、自己修復機械を実現することを目的とし、機械のモデリングの方法について議論する。まず、一般的な機械のモデリングの方法を概観する。これにより定性物理の必要性、有効性を議論し、また、機能のモデリングの困難さを明らかにする。次に、機能を含めて機械を表現するための方法としてFBSダイアグラムを提案し、その有効性をインプリメンテーションを通して検証する。最後に、以上の議論を踏まえ、自己修復機械をモデリングする方法について議論する。

4.1 機械のモデリング手法

機械を計算機上で表現する場合、機械の「機能」、「挙動」、「構造」、もしくは、「属性」といった概念が重要な鍵となっていると考えられる。実際、第2章で述べたように故障診断は「機能」の変化から「構造」もしくは「属性」の変化を導出する作業と考えられるし、設計は「機能」空間から「属性」空間への変換である定義される[Yoshikawa79a]。

本章の目的は、自己修復機械の知的判断部構築の基礎として、これら機械の「機能」、「挙動」、「構造」に関する明確で、一貫性を持ち、計算可能、かつ、一般性のあるモデルを構築することである。そこで本節では、機械のモデリング手法を「機能」、「挙動」、「属性」という三つの概念を軸として概観することにする。

4.1.1 属性のモデリング法

機械の属性のモデリング手法として最も一般的なのは、幾何形状を扱う形状モデル[Mäntylä88]であろう。形状モデルにはワイヤーフレームモデル、サーフェスモデル、ソリッドモデル、非多様体モデル等様々な手法が提案されており、CADの分野を中心として、広く利用されているモデリング手法である。

その他、吉川[Yoshikawa72]は、機械の位相的構造に注目し、部品接続グラフ、面接続グラフ等のモデリング方法を提案している。また、伊藤[Ito90a]はインテリジェントCAD構築のために、設計属性に注目した属性モデリングという考え方を提案している。

しかし、一般には、属性自体に関心を向けてモデリングする手法は少ない。なぜならば、多くのモデリング手法は、動特性、変形等の何らかの振舞いを予測、検証することを目的としていることが多く、その意味で挙動のモデリング法に分類されるからである。

4.1.2 挙動のモデリング法

多くの解析の手法は、機械の振舞い、すなわち、何らかの変化をシミュレートすることを目的としている。このとき、対象とする物理的世界を微分方程式、偏微分方程式等の数学的手法により表現することが一般的であり、力学、材料力学、流体力学、制御工学等多くの分野でこのようなモデリング手法が採用されている。さらに、これ

らのモデリング手法により表現されたモデルは、有限要素法、境界要素法等の数値解析手法により、シミュレートされる [Anonymous90]。これらのモデリング手法を本研究では「数値解析的モデリング」と呼ぶことにする。

一方、最近の知識工学の分野で、物理世界を記号的にモデリング、推論するという目的を持つ「定性物理 [Weld89, Forbus89b]」が注目されている。定性物理については次節で概観するが、ここでは、定性物理と数値解析的モデリングの違いについて述べることにする。

数値解析的モデリングは、力学、熱力学等の特定の分野に関する物理法則を微分方程式等によりモデリングする。しかし、ある対象を解析する場合には、解析分野の選択、適当な物理法則の選択、および、条件の選択といった、対象のモデリングの方法論は数値解析的モデリング手法には含まれておらず、もっぱら、専門家によるノウハウによってモデルが作られている。また、解析結果は、一般に数値として与えられ、その結果を対象のモデル化の正当性を含めて評価する方法論も専門家に任されている。この意味で、数値解析的モデリングは、特定の物理法則に関して詳細にモデル化したものであるといえる。

一方、定性物理は、図 4.1 に示すように、物理的システムのモデル化から解析結果の解釈までの挙動解析のすべての過程で見られる人間の定性的な思考を記号的に表現、推論し、それにより、問題解決を行なおうとするものである [Nishida88]。このように、定性物理は従来の数値解析的モデリングにおいて専門家のノウハウのテクニックに任されていた部分をモデル化したものといえ、その意味で、両者は相互補完的な関係にあるといえる。

しかし、保全の自動化のためには、特に、保全の特徴である不定性を扱うためには対象のモデル化の過程の自動化が必要不可欠であり、その意味で、定性物理が対象としている部分を表現することが要求される。

4.1.3 機能のモデリング法

次に、「機能」のモデリングについて概観する。

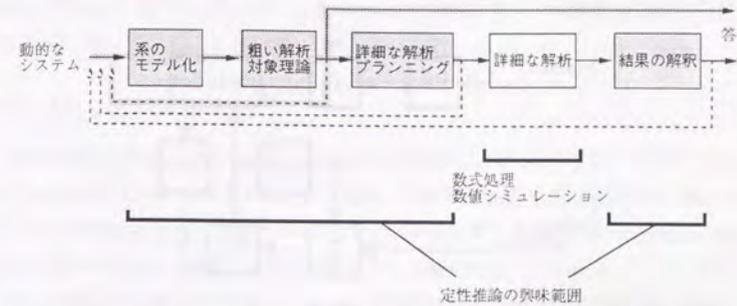


図 4.1: 定性物理のスコープ [Nishida88]

(1) 設計方法論における機能

ドイツにおける設計の研究者らは、いわゆる設計方法論の中で機能を定義、利用している。

Rodenacker [Rodenacker71] は、設計者のガイドラインとなる設計方法論を提案している。この設計方法論は以下のようなものである。まず最初に、設計者は与えられた仕様から設計対象全体の機能を決定しなければならない。次に、設計者は、この全体機能を部分機能に、さらに、部分機能をその部分機能にと機能分割を繰り返す。これにより、機能の構造が構築される。第三に、設計者は、機械要素のカタログを検索し、各部分機能を実現するために最適な機械要素を選択する。そして最後に、機能の構造に基づき、機能分解の逆の過程を辿ることにより、選択された機械要素から機械全体を構築する。この設計方法論において、機能は「物質、エネルギー、情報の入出力関係における変換」であると定義されている。例えば、押し出し機の機能構造は図 4.2 のように表現される。

Roth [Roth82] は、Rodenacker の手法に基づき、機能と機能を発現する機械要素の関係を整理し、系統的な設計カタログを作成した。ここでは、Rodenacker の定義に加え、機械部品の相対的運動を機能の一種であると扱っている。

これらの研究の機能の扱いには幾つの特徴がある。

1. Rodenacker の機能の定義は、入出力関係以外の機能を表現できないという意味

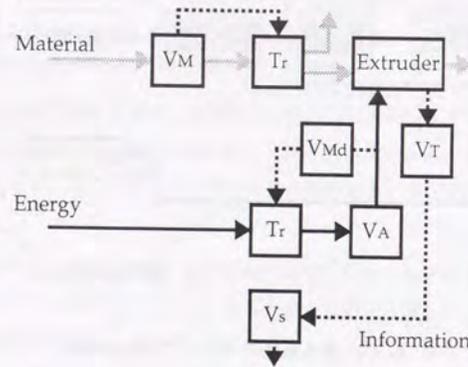


図 4.2: 押し出し機の機能構造 [Rodnacker71]

で不十分と考えられる。例えば、「ものを固定する」機能は入出力関係としては記述困難である。

2. Rothの手法では、明確な機能の定義を行っていない。すなわち、入出力関係から機械部品の相対運動まで様々な抽象度の機能の表現が用いられている。
3. 機能分解を行なうメカニズムが不明確で、主に設計者の主観により実現されるものと捉えられている。
4. Rodnackerの設計方法論においては、機能分解が設計カタログに掲載されているような機能素に到達することが仮定されている。しかし、このような機能素は機構学等のある程度制限された分野でのみ定義可能なものであると考えられる。さらにこの設計方法論では、機械の実体構造と設計者によって構築された機能の構造が一致することが仮定されている。しかし、機械一般で、この仮定が成立しているのかという疑問がある。

Ulrichら [Ulrich88]は、安価でかつ高性能な機械を設計するための戦略として、「機能分担 (function sharing)」を提案している。これは、設計対象の各要素にできるだけ多くの機能を分担させることにより、対象の構造を単純化するという戦略である。この意味で、機能分担は前出のRodnackerの機能構造と実体構造が一致するという仮定の反例となる手法である。機能分担は、機能と挙動の関係の知識を明示的に

記述、利用して実現されている。しかし、ここで使用されている機能の定義自体は明確ではない。

(2) VEにおける機能

Miles[Miles72]は、value engineering(VE)を提案している。VEは、コストを最小に、機能を最大にすることを目的に、材料、設計等を変更することにより、製品やサービスの価値を高める手法である。この手法において、機能は「to do something」という形で表現され、価値は金額に換算されて評価される。

VEは機能の定義自体は明確であるが、機能の評価を目的としているため、機能と実体の挙動、構造との関係については議論していない。

(3) AIにおける機能

人工知能の分野でも、機能、挙動、構造に関して幾つかのアプローチが存在する。de Kleer[de Kleer84a]は、次節で述べるデバイス主義の定性物理を提案している。ここでは、デバイスは部品と部品間におけるエネルギー、物質等の流れによるネットワークとして表現される。この手法においてde Kleerは、構造は「デバイスが何であるか」、挙動は「デバイスがどう振舞うか」、機能は「デバイスが何のためにあるか」を表すものであるとしているが、これらの定義は厳密であるとは言いがたい。この手法は、次節で述べるように、デバイスの構造から挙動を推論する「因果解析」と得られた挙動からデバイス全体の機能を導出する「目的論的解析」から構成される。

この手法には次のような問題点がある。

1. 機能と挙動の区別が明確ではない。
2. デバイスの階層的な構造を扱えない。
3. この手法内では機能を明確に定義していないが、Rodnackerの手法と同様、機能を入出力関係の抽象化として扱っている。なぜならば、機能は目的論的解析により、入出力関係である挙動からパターンマッチングにより導出されるため、入出力関係以外の挙動に関連する機能は表現不可能であるからである。このような機能の表現は、de Kleerが扱っている電気回路の分野では適当であると考えられるが、機械の分野では表現できないものもあると考えられる。例えば、Faltings[Faltings87]は、機械的デバイスの機能は部品間の相対運動により

発現されるもので、機能の表現は部品単体よりもその相対関係に基づいて表現しなければならないと述べている。

近年、Chandrasekaranらのグループを中心として、定性物理の考え方を利用しつつも、機能や機能的階層性をより直接的、明示的に扱おうとする"Functional Reasoning"と呼ばれる研究分野が開かれつつある。

Keuneke[Keuneke91]は、機能は人間が装置に対して期待するものであるとし、ToMake、ToMaintain、ToPrevent、ToControlという四種類の述語を用いて表現し、挙動は機能がいかんして実現されるかの記述であるとし、因果的な状態変化として表現している。この手法は、この対象表現の枠組に基づき故障診断を実現している。このアプローチでは、対象表現は機能の観点に基づいて、それを理由付けするための挙動、状態という記述になっており、機能として注目していない挙動が得られない場合や、主観的な機能に基づいた挙動記述が物理的に正しくない可能性が残されている。また、四種類の述語によりあらゆる機能を記述可能であるかという問題も残されている。

Bradshawら[Bradshaw91]は対象記述において目的に関する知識と、挙動、構造に関する知識の両方が必要であるとし、挙動、構造に関する知識はデバイス主義の定性物理により記述し、目的に関する知識は要求される部品の挙動として表現している。この表現の枠組を用いて、ここでは設計診断を実現している。また、目的の上下関係に基づき機械の階層的表現をユーザの手により構築する方法も提案している。さらに、Franke[Franke91]は機械の目的を表現する枠組として挙動表現の一部を抽象化し、論理的に導出する方法を提案している。ここでは、この抽象的挙動表現を用いて再設計や修正を支援する手法を提案している。

さらに、Bylanderら[Bylander85]は、機械における「階層」を扱うために、デバイス主義の手法を拡張したconsolidationを提案している。この手法においては、デバイスの部分的挙動を「因果パターン」と呼ばれる知識を用いて組み合わせることにより、より抽象化された上位階層の挙動を導出する。この過程がconsolidationと呼ばれ、このパターンマッチングを繰り返すことにより対象系の機能を導出できるとしている。このとき、因果パターンとして考えられる知識は観点や対象の記述方法の違いにより異なるレベルの内容を含んでいると考えられる。また逆に、特定の対象に依存せずにこのような因果パターンを導出可能であるかという疑問が残る。そのため、

大規模システムにconsolidationを適用した場合、因果パターンの知識の適用制御を何らかの方法で実現しなければ人間にとって意味のある機能を導出できないと考えられ、このようなマッチングの制御自体が人間の機能的視点と関連があると考えられる。

これら"Functional Reasoning"の手法では、対象表現がKeunekeの方法のように主観的機能に基づいているものか、もしくは、その他三つの手法のように物理的な挙動記述に基づいたものである。BradshawやFrankeの研究の場合、機能と挙動の区別は明確ではなく、例えば、後者の研究の中で「抽象化」と呼ばれている挙動表現の情報量を少なくした表現がすなわち機能であるのかという疑問が残る。この点、Bylanderの方法はこれを因果パターンの収集により抽象化を実現している。また、挙動の表現としてこれらの研究では、デバイス主義の定性物理を用いているが、これらの研究が共通して扱っている設計や故障診断の分野において、特に対象として機械を扱った場合、デバイス主義的なアプローチはその表現の柔軟性に限界があると考えられる。

(4) まとめ

以上、幾つかの分野における機能の取り扱いを概観したが、問題点は以下のように整理できる。

1. 機能と挙動の概念の違いが明確に区別されていない。もしくは、本来区別できないものであるのか。
2. 機能の定義としては、以下の四種類が提案されており、機能の一般的、かつ、明確な定義が存在していない。
 - (a) Rodenackerやde Kleerの方法では、機能は入出力間の変換であった。
 - (b) RothやFaltingsの方法では、機能は機械の相対運動により発現されるとされていた。
 - (c) VEでは、機能は"to do something"と表される。
 - (d) Functional Reasoningでは、機能は挙動の抽象化、もしくは、挙動は機能の実現方法の説明であった。特に、Keunekeの手法では、ToMake、ToMaintain、ToPrevent、ToControlの四種類の述語により表現される。

3. 機能と機械の階層性は関連すると考えられるが、機械の階層的構造を一般的に扱う方法が明確ではない。例えば、階層構造を構築するメカニズムが明らかにされていないという問題点がある。

4.2 定性物理

第4.1.2項で述べたように、定性物理は機械の挙動をモデリングするために一つの重要な技術を提供していると考えられる。また、次節で述べるように本研究では機能を挙動との関係で表現するので、定性物理は機能を表現する場合にも重要な手法である。本節では、この定性物理について概観する。

4.2.1 定性物理の基本的手法

定性物理、もしくは、定性推論は P.Hayes[Hayes85a, Hayes85b] が先駆けとなり、Artificial Intelligence 誌の定性推論特集号 [Bobrow84] でその基本的概念と手法が確立した。定性推論の基本的な考え方は、de Kleer の研究 [de Kleer84a, de Kleer84b] と Forbus の定性プロセス理論 (Qualitative Process Theory)[Forbus84] に示されている。

de Kleer の手法では、ある装置 (デバイス) が与えられた場合に、デバイスを構成する部品の性質と部品間を流れる物質、エネルギー等の流れ (Conduit) により対象を表現する。このとき、部品の性質や流れは因果関係を記述した合流方程式 (Confluence) で与えられる。この部品と Conduit により構築されるデバイス・トポロジをもとに次のように推論を行なう。まず、合流方程式に基づきデバイスの挙動、すなわちパラメータ値の変化の伝播、を導出する「因果解析」を実行し、デバイスの挙動を表すメカニズム・グラフを得る。次に、デバイス全体の機能を導出するために、メカニズム・グラフに因果パターンと呼ばれる部分グラフ単純化のルールを適用することにより、グラフを単純化する「目的論的解析」を行なう。以上により、デバイスの働き (機能) を推論する。

定性プロセス理論は、物体や物体間の関係上に発生する物理現象を基本として世界をモデル化する。すなわち、物理世界を個体、個体ビュー、および、プロセスにより記述する。個体は、机、ギア、水等の世界に存在する実体であり、個体ビューは水の温度等の個体や個体の集まりが取り得る状態を記述する。そして、プロセスは、水の沸騰のような個体の状態に依存して発現する物理現象を記述する枠組である。例えば、「伸び変形」というプロセスの記述を図4.3に示す。以上の対象記述に基づき、定性プロセス理論は次のような三段階で挙動推論を行なう。まず、対象上の生起する個体ビュー、プロセスを決定する「状態解析」を行なう。次に、決定された状態での

各パラメータの変化を推論し、最後に、各パラメータがランドマークに到達するかどうかを調べる。これを繰り返すことにより、対象の状態遷移(ヒストリー)を導出することができる。

process Stretching

Individuals:	B an elastic object
Preconditions:	\neg Position-Constrained(B)
Quantity Conditions:	\neg Compressed(B) $A_s[\text{applied-force}(B)] = 1$ $A_m[\text{applied-force}(B)] > A_m[\text{internal-force}(B)]$
Relations:	Let SR be a quantity $SR \propto_{Q+} (A_m[\text{applied-force}(B)] - A_m[\text{internal-force}(B)])$
Influences:	I+(length(B),SR)

図 4.3: プロセスの記述例 [Forbus84]

以上の二つの基本的手法は、その世界のモデリング方法の違いから、de Kleer らの方法はデバイス主義、定性プロセス理論はプロセス主義と呼ばれている。両者の基本的な相違は、デバイスや構成要素といった物理的実体に注目するか、その上で生起する物理現象に注目するかということである。このため、プロセス主義の定性物理の方が例えば、水が沸騰することによる対象系の状態変化、構成要素間に生起する現象等の挙動をより自然に表現できる。このような変化は一般に、対象の特性を表す方程式系の変化と考えることができるが、プロセス主義の定性物理ではプロセスの発火によりこのような方程式系を動的に生成するという特徴を持っており、上記の問題を容易に扱える。

4.2.2 定性物理の研究動向と問題点

前項で述べた二つの基本概念をもとにして、現在の定性物理の研究は主に定性シミュレーション技術、自動モデリング、空間へのアプローチ、および、応用への適用等の分野でなされている。定性シミュレーションは、定性物理の特に推論技術に関する研究である。Kuipers[Kuipers86] は、連立定性微分方程式を一般的に推論可能な

QSIM(Qualitative SIMulation) アルゴリズムを開発した。この QSIM アルゴリズムは連立定性微分方程式に対し、あり得る解は全て導出するという意味で「健全性」を持つが、定性値のみを扱うため定量値に比較して情報量が不足し、導出された解が全て定量的な意味で正しいとは限らないという意味で「不完全性」を持つことが知られている。ただし、後者は定性推論一般に当てはまる問題である。また、この分野の研究としては、系外部からの操作を定性プロセス理論を拡張し Action という形式で記述する研究 [Forbus89a]、量の概念を導入した研究 [Mavrovouniotis87]、不連続変化の取り扱いの研究 [Nishida87] 等がある。自動モデリングは、与えられた現象を説明するために最適な、もしくは、もっとも簡単なモデルを自動的に構築しようというもので、Nayak ら [Nayak91] の研究や Williams[Williams91] の研究等がある。また、空間へのアプローチとして、空間の概念や形状を持つ機械部品の運動を定性的に推論する研究が行なわれている [Faltings87, Forbus87, Joskowicz89]。さらに、これらの技術を踏まえて、設計 [Joskowicz88, Kiriyama91, Williams90]、故障診断 [de Kleer87, Dague87] 等の問題解決に应用するための研究が盛んに行なわれている。これら定性物理の研究の問題点として以下の三項目が挙げられる。

1. 定量情報との結合

本来の定性物理の枠組からいえば、おおまかな推論を定性的に行なった後に、より詳細な解析を定量情報を用いて行なうことになっている。しかし、定量情報との結合 [Kuipers88]、もしくは、定性推論への定量情報の導入 [Raiman86, Mavrovouniotis87] の試みがなされているが、成功しているとはいえない。この問題は、定性物理の実用を目的とした場合、大きな問題になる。

2. 空間、幾何情報

記号的表現、推論を行なう定性物理は、本質的に幾何形状、空間概念に関する表現、推論が難しい。この問題に対し、上記のように様々な研究がなされているが、十分有効な手法を提案するには至っていない。

3. モデリングの方法論

定性物理は図 4.1 にも示したように、モデリングの支援、自動化も一つの目的としている。これに対し定性物理はモデリングのために必要な原則的な知識を収集することにより解決するという方法を取っている。しかし、観点の違い、Ontology の違い等をいかにモデリングするか、すなわち、モデリングの方法のモデリングが問題になってきている。

4.3 FBS ダイアグラム

第4.1節でみたように、機械をモデリングする場合に機能が重要な役割を果たすが、その定義、機能と挙動の相違等が明確ではない。そこで本節では、機能、挙動、状態、および、構造間の相違とそれらの関係、さらに、機械の階層構造について考察を行なう。そして、機械のモデリング手法としてFBS(Function-Behavior-State)ダイアグラムを提案する。

4.3.1 FBS ダイアグラムの提案

(1) 言葉の定義

ここでは、物理的世界のみ、すなわち、物理的機能、物理的挙動、物理的構造のみを考え、美的機能、経済的機能等は考えないことにする。

まず最初に、吉川[Yoshikawa79a]に従い、世の中に存在するもの全てを「実体」と呼び、実体を持っている様々な物理的、科学的性質を「属性」と呼ぶ。本研究では特に、実体の概念をその名前等の識別子で記号化し、表現する。また、属性を質量、体積等の属性の項目とその値により表現する。さらに、属性や実体を関係付ける概念を「関係」と呼ぶことにする。このとき、関係同士を関係付けるような関係も考えられる。本研究では、以上の用語を用いて、状態集合 S を以下のように定義する。

$$S = \langle E, A, R \rangle \quad (4.1)$$

すなわち、ある実体の状態 s は、実体の集合 $e \in E$ 、属性の集合 $a \in A$ 、および、関係の集合 $r \in R$ により記述される。このとき、何らかの方法で実体の内部と外部という区別がなされた場合、実体の状態は「内部状態 s_i 」と「外部状態 s_o 」に分けることができる(式4.2、式4.3参照)。

$$s_i = \langle e_i, a_i, r_i \rangle \quad (4.2)$$

$$s_o = s - s_i \quad (4.3)$$

ただし、 e_i 、 a_i 、 r_i はそれぞれ内部実体の集合、内部属性の集合、および、内部関係の集合を表す。例として文鎮の状態記述を図4.4に示す。図4.4において、「Paper Weight」と「Paper」は実体であり、これらは「on」という関係で結ばれている。

また、「Paper Weight」は、重さ 1kg 、体積 100cm^3 、密度 10g/cm^3 という3つの属性を持っており、これらの属性は $D = W/V$ という関係で結ばれている。

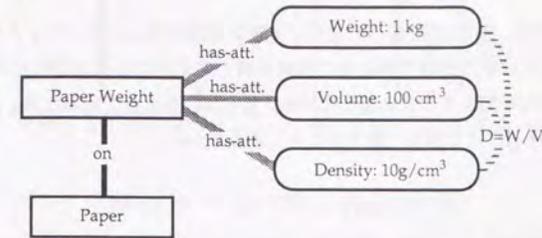


図4.4: 文鎮の状態記述

ここでは、いわゆる「構造」といわゆる「状態」の違いというのは本質的な相違ではないと考え、区別しないことにする。すなわち、一般論として、実体の状態のうち比較的長時間変化しない部分が構造と呼ばれ、短時間に頻繁に変化する部分が状態と呼ばれている。この区別は変化しない時間の長さに依存する相対的なものであり、例えばギア割れ、導線の断線等保全においては機械の構造と思われていたものが変化する。このため、実体の表現上で構造と状態を厳密に区別することは困難、かつ、表現の柔軟性を失わせる。そこで本研究では、これら両者を合わせて「状態」と呼ぶことにする。これは、本状態表現が時間概念を含まないことから当然の帰着である。

次に、本研究では、ある状態 s_1 が決まると何らかの方法で次の状態 s_2 を決めることができるかと仮定する。このとき、「挙動」を「一つ以上の状態の列」と定義する。すなわち、ある挙動 b は、

$$b = s_1 \Rightarrow s_2 \Rightarrow s_3 \Rightarrow \dots \quad (4.4)$$

$$s_1, s_2, s_3, \dots \in S$$

と表現される。ここで、 \Rightarrow は状態変化を表す記号とするが、この「状態変化」の表現方法には二通りの考え方ができる。第一の考え方は、「単位時間」を導入する考え方であり、 \Rightarrow が1単位時間の経過を表す。例えば、電子回路のクロックに基づく動作を表すためにはこの考え方が有効であろう。図4.4に示した文鎮の状態 s が変化し

ないという挙動 b は、この考え方に基づくと以下のように表せる。

$$b = s \Rightarrow s \Rightarrow s \Rightarrow \dots \quad (4.5)$$

第二の考え方は、第4.2節で述べた定性プロセス理論のヒストリーのような、時間の長さに依存せずに状態遷移の列として挙動を表す考え方である。このとき、 \Rightarrow は何らかの状態変化が起きたことを示している。例えば、前出の文鎮の状態 s が変化しないという挙動 b は、この場合、以下のように表現される。

$$b = s \quad (4.6)$$

これらの表現の違いは、対象の挙動をより適当に表現可能な方法が選択されるべきものであるので、本研究では両方を状況に応じて使い分けることにする。

ここで問題となるのは、上で定義した挙動は無数にあるため、全ての挙動を記述することはできないということである。そこで本研究では、このような状態変化は「物理法則」により規定されていると仮定する。すなわち、実際に起こっている状態変化を物理法則の集合によって規定される状態変化の記述で近似表現する。この「物理法則」は、「実体の状態がある物理的条件を満たす時に、実体の物理的挙動(状態変化)を決定する知識」であると定義することにする。例えば、ボールを離して自由落下させた場合に、「ボールが落下する」というのがボールの「挙動」であり、この挙動は「万有引力の法則」により決定されていると考える。以上により、状態と挙動の関係を整理することができた。本研究では、物理法則で結びつけられたこの挙動と状態の関係を「B-S 関係」と呼ぶことにする。

ここで問題となるのは、同じ実体の挙動を表現する場合にも、興味の対象によって異なる方法が存在することである。例えば、「導線に電流が流れる」という挙動も電気工学の観点からは抵抗、電圧、電流値等によって表されるし、電子工学の観点からは、電子の運動として表現することもできる(図4.5参照)。このような違いは、いわゆる Ontology の違いと呼ばれており、上記の例のように分野、もしくは、視点の違いや表現の粒度の違い、詳細度の違いによって区別されるものである。そこで本研究では、これらの Ontology の違いを表現するために「視点(view)」を導入し、視点を「挙動や状態の表現スキーマ」と定義する。一つの視点は、状態を記述するためのボキャブラリー、挙動を記述するためのボキャブラリー、および、視点内では無矛盾な物理法則の集合により記述されるとする。このとき、先に述べた時間概念も視

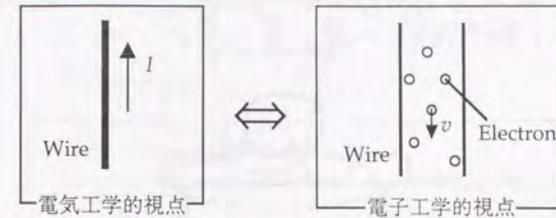


図4.5: 視点の違いによる挙動表現の違い

点の記述の一部として定義される。すなわち本研究では、視点を選択して初めて、挙動や状態を記述することができると思う。

さてここで、「機能」を「その利用を目的として、人間によって認識された挙動の抽象的な表現」と定義する。すなわち、機能は人間によって認識、抽象化された挙動のイメージであると考えられる。Miles[Miles72]に従い、一般にこれは「to do something」の形で表現されるものとする。本研究では、この人間による認識、抽象化の過程を対応 Γ_{ab} として、機能の集合 F と挙動の集合 B の関係を次式のように表し、「F-B 関係」と呼ぶことにする。

$$\Gamma_{ab}: B \mapsto F \quad (4.7)$$

この抽象化は実体の役割や価値という人間の主観的な観点から行なわれるため、多くの場合この F-B 関係は一対一対応にはならない。例えば、「音を出す」という機能に対して対応する挙動には、「二物体の衝突」や「一物体の振動」というように複数考えられるし、逆にある挙動に対し様々な機能を割り当てることもできる。ここで、挙動と対応していない機能は実現方法が未決定の機能を意味している。すなわち、機能は挙動と対応して初めて実現される。これと反対に機能に対応しない挙動は考えられる。例えば、「無人島における木の葉の落下」は一般には機能と関係がない。

機能の表現と挙動の表現がよく似ていることが多いという事実も、機能が挙動を観察した表現であることを示していると考えられるが、一般には、その機能の表現に明示的、もしくは、暗黙的に何らかの人間の意思が含まれている。例えば、図4.6のような変化しない挙動に対し、Aの機能は「Bを加工するために、Bを支える」こと

であり、その時の挙動は「Aに支持されて、Bは静止している」と表現されることが典型的な例として考えられる。

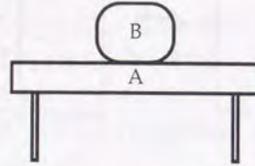


図4.6: 保持機能の例

この定義から分かるように、機能を表示するためには、目的としている機能を発現する挙動の特徴を表示するのに適当な「視点」を選択しなければならない。つまり、機能の表現と視点の選択は互いに関係があり、両者、および、F-B関係の記述は主観的に決まるものであると結論付けることができる。また、このF-B関係が多対多対応であるということが「機能保全」が実現可能であるという裏付けになっており、本研究で機械を表示する場合には、この多対多対応を表示可能にしなければならない。以上に述べた、機能、挙動、および、状態の関係を図4.7に整理して示す。

以上より、機能を客観的に記述できないことが明らかになった。しかし、計算機上に機能を含めて機械を表示するためには、何らかの一般的な機械表現スキーマを提案しなければならないが、以上に述べた機能、挙動、状態の関係はこのために以下の理由により有効な手法を提示している。

- 主観的に行わなければならない部分を、主観的部分として独立させ、明示的に記述できる。本手法で主観的部分は、機能の記述、F-B関係、および、視点の選択に表現される。
- ある実体に関する挙動、状態、B-S関係は視点に依存して客観的に決めることができる。この部分は、計算機による一貫性の管理等を実現可能である。
- 主観的部分(機能)と客観的部分(挙動、状態)を対応付けることにより、主観的部分の無矛盾性、実現性を客観的部分に基づき、間接的に判定可能である。

この図4.7に基づく機械の表現スキーマを本研究では「FBSダイアグラム」と呼ぶことにする(図4.8参照)。

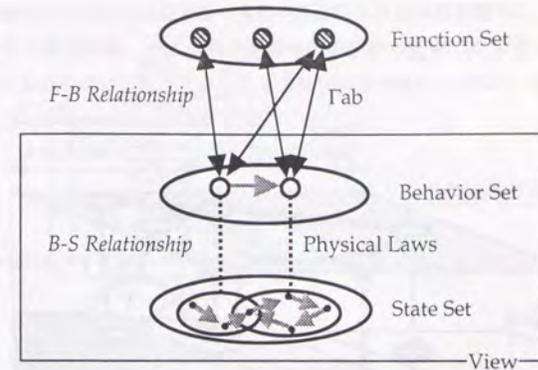


図4.7: 機能、挙動、状態間の関係

4.3.2 機械の階層

(1) 機械の階層表現

次に、機械の階層構造についてFBSダイアグラムに基づき考察する。Rodenacker [Rodenacker71]が指摘しているように、設計者は設計対象を部分的な要素に分割することを繰り返し、部分要素が充分小さくなった後にその状態を設計する。この設計者による部分分割はある種の階層構造を生成するが、その分割方法は設計者の視点によって異なると考えられる。これと同様に、生産技術者が設計対象の図面を見て認識する機械の階層構造や使用者や保全技術者が実物の機械を見て認識する階層構造等も様々に異なると考えられる。例えば、機械内部の動作原理の違いによる分割や製造への利便性を考えた部品の組み立て階層などが階層構造として考えられる。それゆえ、本研究では、このような実体の階層構造というのは客観的に物理法則により一意に決まるものではなく、人間の機能的観点からの実体の認識、抽象化により構築されるものであるとする。このような階層構造を人間が構築する理由は、機械の認識、理解が人間にとって容易になるからである。そこで本研究では、このような階層構造に関する情報も機能の記述の一部としてFBSダイアグラムに記述することにする。

このようにして構築されたFBSダイアグラム上の機械の階層構造を図4.8に示す。

本研究では、この機能階層構造は機能間の全体・部分関係や抽象・具体関係により階層化されると考える。機能の階層構造が構築されたとき、機能階層の各部分機能に対応した挙動、および、状態を表現可能なように視点を選択しなければならない。図

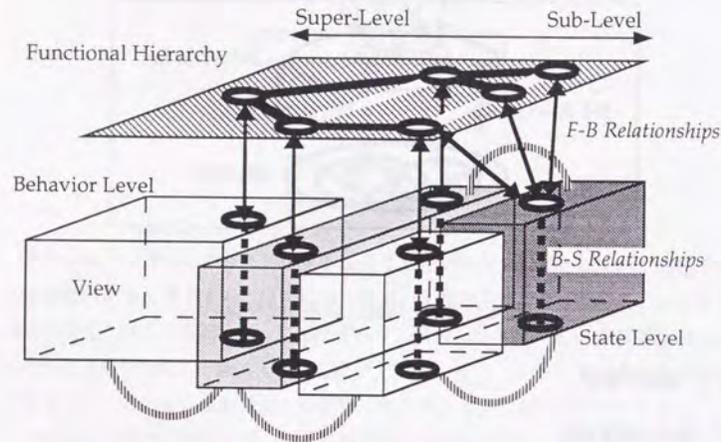


図 4.8: FBS ダイアグラム

4.8に示すように、視点は機械分野、電気分野等様々な分野に類別され、さらに、機能階層の表現レベルに対応して、同じ分野の視点においてもその表現の粒度、詳細度が異なってくる。このように、機能レベルの階層構造は挙動、状態レベルでは視点間の関係に帰着するため、視点間の関係は実体の階層的表現の無矛盾性、一貫性を管理するために重要な役割を担っている。この部分を計算機により自動化することにより、FBS ダイアグラムによる機械の表現を柔軟、かつ、無矛盾なものにすることが可能になると考えられる。メタモデル・メカニズム [Kiryama91] は、この視点間の関係中の分野の違いを定性物理を用いて、計算機上で管理するものであり、FBS ダイアグラム実現のために有効な手法となると考えられる。また、里 [Sato91] は粒度の違いによる視点の違いについて考察を行なっている。

例えば、図 4.9に示すブザーを対象とした FBS ダイアグラムの一部を図 4.10に示す。図 4.10に示すように、本例では、実体 Buzzer は衝突を引き起こす Boss & Oscil-

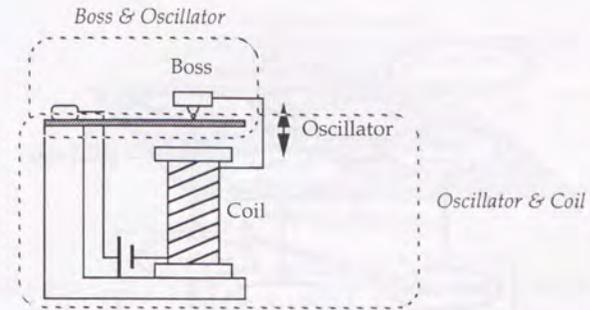


図 4.9: ブザー

ator 部分と Oscillator を曲げる Coil & Oscillator 部分に分ける階層構造により表現されている。

(2) 階層構造構築のメカニズム

ここでは、FBS ダイアグラムに基づいて、ある機械を認識、理解する過程を通して機械の階層的なモデルを構築するメカニズムを考察する。

階層構造構築のメカニズムは、典型的な方法として二つの方法が考えられる。すなわち、機能階層を中心として構築する方法と視点間の関係を中心として構築する方法である。実際に階層構造を構築する場合には、この二つの方法を組み合わせて利用していると考えられる。

ここでは例として、図 4.9のブザーを認識し、モデルを構築する過程を通して、この階層構造を構築するメカニズムを整理することにする。

第一の方法、すなわち、機能階層構造を中心に構築する方法は、次のようなものである。まず、図 4.9のブザーを人間が見た時、このブザーの機能「音を出す」は、例えば、Boss と Oscillator の部分が部分機能「衝突を起こす」を発現し、Coil と Oscillator の部分が部分機能「振子を曲げる」を発現することを交互に繰り返すことにより、発現されていると認識する。このような認識より、図 4.11のような機能階層構造を構築することができる。次に、それぞれの部分機能に対して、その機能を発現している挙動、状態を表現するために適当な視点が選択される。この時、結果として視点

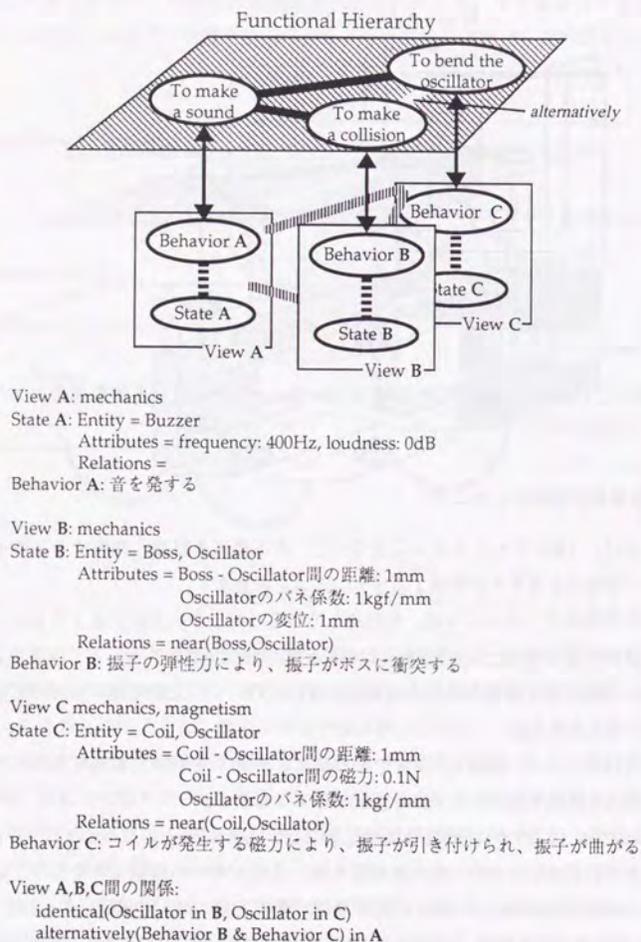


図 4.10: ブザーの FBS ダイアグラム上での表現例

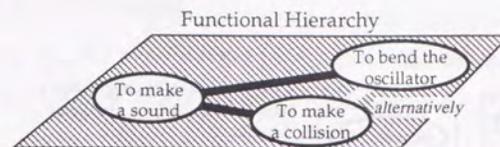


図 4.11: 階層構造構築の例 (1)

間の関係が決まり、図 4.10 の FBS ダイアグラムが構築される。このとき、F-B 関係は多対多対応であるので、視点は機能のみに基づいて一意に設定することはできず、人間により適当に選択される。また、ある機能階層に対応するような適当な視点を設定できない場合も考えられ、この場合は機能階層構造を実現不可能であるということがわかる。すなわち、人間の主観によって決定される機能階層構造は、視点の選択により、物理法則に基づいて物理的実現性を評価されることになる。

第二の方法は、まず、視点を選択する方法である。ある実体の表現は視点に依存し変化する。例えば、図 4.9 のブザーも、電気的視点においては図 4.12 のように表現される。視点間には何らかの関係があるので、視点を決めることにより、B-S 関係、および、視点間の関係が自動的に決定される (図 4.13 参照)。次に、これらの視点間の関

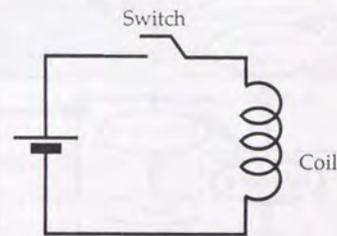


図 4.12: 電気的視点から見たブザー

係に基づいて、視点間の関係に矛盾ないように F-B 関係と機能階層が人間の認識、抽象化により構築される。すなわち、選択された視点内の表現を人間が理解するために機能的記述が後付けされ、図 4.14 に示すような記述が行なわれる。

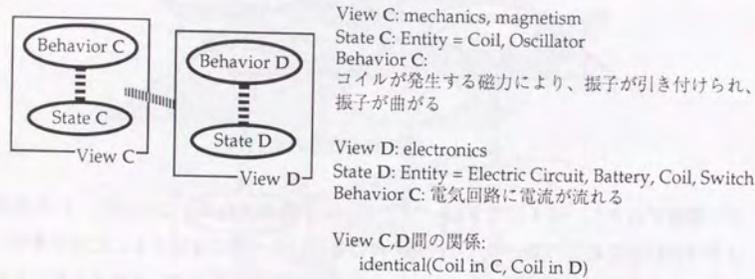


図 4.13: 階層構造構築の例 (2)

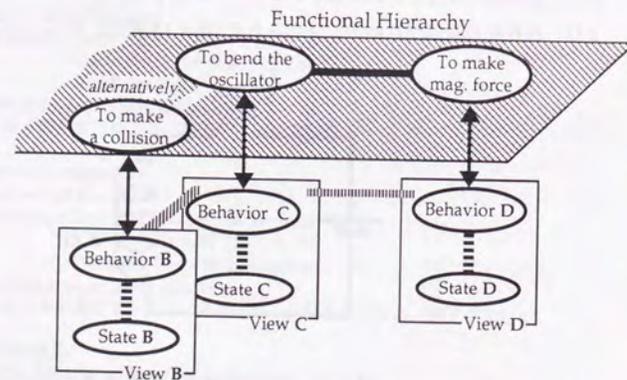


図 4.14: 階層構造構築の例 (3)

4.3.3 FBS ダイアグラムの特徴

以上で述べた FBS ダイアグラムの特徴を示すために、複雑な機能階層構造の例として図 4.9 で示したバザーの機能階層構造の一部を図 4.15 に示す。図 4.15 に示すよう

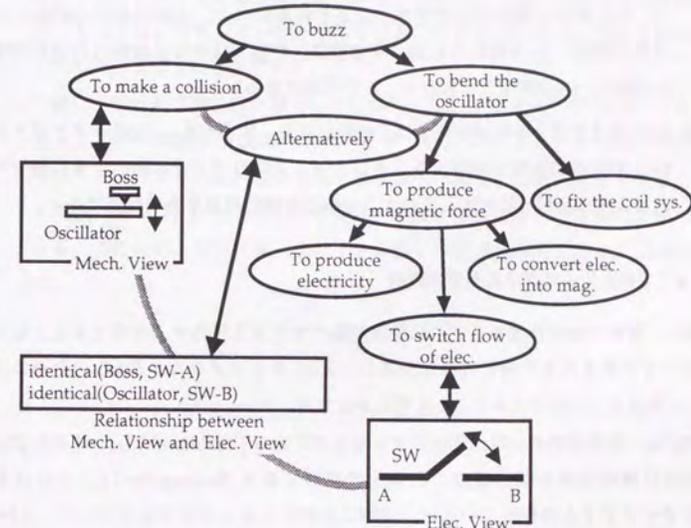


図 4.15: 複雑な機能階層構造の例

に、ある実体の組は機械系の視点では「振子とボス」と表現され、また、電気系の視点では「回路のスイッチ」と表現されており、この実体の組は「衝突を引き起こす」機能と「電流のスイッチングを行なう」機能の両方に関連していることがわかる。これは、Rodacker のように機能と実体の一対一対応を前提とするとうまく表現できない例である。

以上の FBS ダイアグラムの特徴は次のようにまとめられる。

1. 人間の主観をそのまま表現するので、人間にとって理解容易な表現である。すなわち、機械の機能を階層的に表現することにより、上位部分はより抽象的

に、下位部分はより詳細に表現することができる。また階層的記述により、ある部分の全体の目的の中での役割と、その実現方法を表現することができる。

2. 主観的部分と客観的部分が明示的に分けて記述されている。そのため、一般に使用される様々な視点と視点間の関係に関する知識ベースと構築することにより、ユーザが主観的部分で任意の記述を行ないつつ、主観的部分に対応する視点間の関係、B-S関係の自動一貫性管理、挙動、状態の実現性の判定等の管理を知識ベースを利用して行なうことが可能になる。
3. FBSダイアグラムの階層表現は柔軟性に富む。すなわち、FBSダイアグラムのF-B関係の柔軟性を利用することにより、上記のプザーの例のように機能の構造と実体が一対一に対応しないような複雑な機能階層を表現可能である。

4.3.4 FBSダイアグラム上での設計

次に、機械の設計作業をここでは設計対象のモデルを構築する作業であると捉え、FBSダイアグラム上で設計作業を整理し、FBSダイアグラムがCADのためのモデリング手法として有効であることを明らかにする。

設計は、機能空間から属性空間のマッピングであり[Yoshikawa79a]、その作業は、基本的に機能分割と部分機能の具体化の過程である[Rodenacker71]。この作業はFBSダイアグラムの枠組でいえば、実際には存在しない設計対象に対して、71ページで述べた機能展開による階層構築を行なうことにより、設計対象に関する一つの貫したFBSダイアグラムを構築する作業であると捉えることができる。すなわち、設計者は、想像力や直感、もしくは、過去の事例に基づき機能レベルの階層構築を構築しつつ、シミュレーション、実験、データ検索等に基づき機能階層に対応した実現可能な挙動、状態レベルの構造を構築する。また、挙動、状態レベルの評価の結果が不適当な場合には、機能階層構造や挙動、状態レベルの構造を大幅に変更するというも行なわれる。このとき、機能展開による階層構築が設計の中心であるが、73ページで述べた視点の選択による階層構築の方法も併用される。例えば、機械設計においては、「機構のカタログ」的な視点が階層構造の下位レベルの視点として多用されていると考えられる。

FBSダイアグラムは、以下の理由により設計対象の表現に適していると考えられる。

1. 以上に述べた設計者の設計過程に対応した柔軟な設計対象の表現が可能である。
2. 機能レベルに設計者の主観的発想や意図を保持することが可能であり、かつ一方で、視点が持つ物理法則を用いて挙動、状態レベルの物理的実現性、無矛盾性をシミュレートすることが可能である。これにより設計者は必要な挙動が発現、もしくは、発現していない理由を知ることが可能になり、設計を進める上でのヒントを得ることができる。
3. 一般に、設計終了時に設計解として従来与えられる情報はFBSダイアグラムで言うところの初期状態表現のみであるが、FBSダイアグラムを用いて設計を行なうことにより、設計者の意図する各部分の機能、挙動、そこで生起している現象、および、設計者から見た機械の階層構造情報を保持することが可能になる。このため、設計対象に関してより多くの情報を保持することが可能になる。

4.4 インプリメンテーション

前節で述べた FBS ダイアグラムの有効性を検証するために、実験的な概念設計支援システム「FBS モデラー」を構築した。本節では、この FBS モデラーについて述べることにする。

FBS モデラーは概念設計段階において、設計者がブレインストーミングを行ないながら設計対象のいわゆる基本構造を決定する作業を支援するツールである。概念設計支援というアプリケーションにおいて以下の項目を評価することにより、本システムは FBS ダイアグラムの枠組の有効性を検証することを目的としている。

1. FBS ダイアグラムにより機能を記述可能か。
2. FBS ダイアグラムにより機械の階層構造を表現可能か。
3. FBS ダイアグラムを用いて設計を行なうことが可能か。
4. 問題点の洗い出し。

4.4.1 FBS ダイアグラムの実現方法

ここでは FBS ダイアグラムの枠組を以下の方法により実現する。

機能は、一つの機能を表 4.1 に示す「機能プロトタイプ」と呼ぶ形式で記述、収集し機能知識ベースを構築することにより表現する。表 4.1 において、「機能名」はその機能を表すシンボルであり、ここでは「動詞+目的語」の形式をもつシンボルによって表現する。「展開方法」は、過去に存在した、もしくは、設計者が考えられる

表 4.1: 機能プロトタイプの定義

項目	内容
機能名	動詞 + 目的語
展開方法	部分機能ネットワークの集合
実現挙動	フィジカル・フィーチャー候補の集合

その機能の展開方法である。一つの展開方法は展開後の部分機能を表す機能プロトタイプのネットワークにより表現され、一つの機能プロトタイプは複数の展開方法を持つことができる。「実現挙動」は、過去に当該機能を実現した挙動や設計者が実現可能であると考える挙動を記述する。一つの挙動は後述するフィジカル・フィーチャー

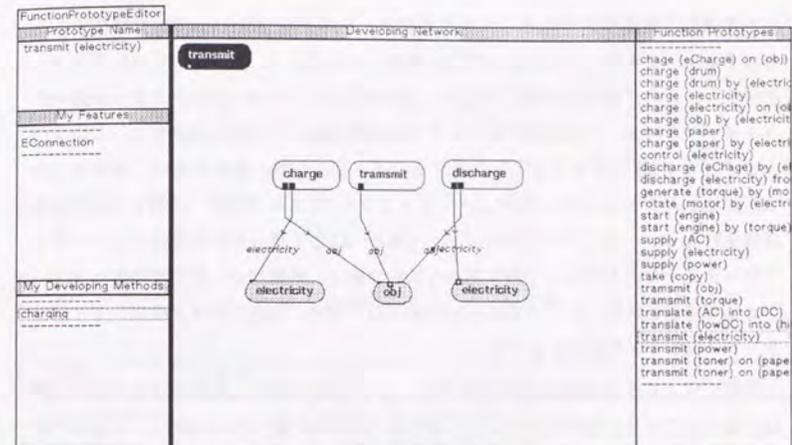


図 4.16: 機能プロトタイプの記述例

(設計者が良く使用する因果的に関連ある物理現象と機械要素の集まり) [Kiriya91] で表すことにする。このとき、一つの機能プロトタイプは複数の実現挙動の候補を持つ可能性がある。

実際のインプリメンテーションでは、PrototypeEditor と呼ばれるツールにより機能プロトタイプの知識獲得を行なうが、例として、機能 "transmit electricity" の記述を図 4.16 に示す。

図 4.16 において、機能 "transmit electricity" は、My Features ウィンドウに示すように、導体間の電気的接続状態を表すフィジカル・フィーチャー "EConnection" により実現可能である。また、"charging" と名付けられた展開方法により、機能 "charge electricity on obj"、機能 "transmit obj"、および、機能 "discharge electricity from obj" を組み合わせることによって実現可能であることが示されている。

挙動、状態レベルの表現は以下のように行なう。ここでは、定性プロセス理論 [Forbus84] を用いて挙動、状態レベルの表現を行なうことにする。その理由は、挙動、状態、物理法則、視点等の物理的概念を定性プロセス理論を用いることにより直接記号的に取り扱うことができるからである。そこで本研究では、挙動・状態レ

ベルの表現、推論システムとしてメタモデル・システム [Kiryama91] を利用することにする。メタモデル・システムでは、個体、個体ビュー、プロセスをインプリメンテーションとしては区別せずに「ビュー」と呼び、このビューのクラスを「挙動プロトタイプ」と呼ぶ。この挙動プロトタイプは挙動知識ベース内に記述され、このプロトタイプを FBS モデル上でインスタンス化した個体、個体ビュー、プロセスの依存関係を表すネットワーク (ビュー・ネットワーク) として挙動・状態レベルの対象表現を行なう。一方、フィジカル・フィーチャーはよく用いられる部分ビュー・ネットワークを一つのものとして扱うための枠組であり、挙動プロトタイプのネットワークとして定義される。以下、文献 [Kiryama91] に従い、挙動プロトタイプ、フィジカル・フィーチャーの定義を述べる。

挙動プロトタイプの定義を表 4.2 に示す。この定義の形式、内容は定性プロセス理論に基づいており、挙動プロトタイプが生成される条件が conditions に、生成後の影響が influences に記述される。ここで、個体が conditions や influences の記述を持たない実体の存在を表すシンボルであるのに対し、個体ビュー、プロセスは conditions や influences の記述を持つという違いがある。個体ビューとプロセスの違いはプロセスが influences 内で物理現象発現の結果として状態変化を生起させる influences¹、s-influences の記述を持つのに対し、個体ビューがこれらの記述を持たないことである。

ここで、一般に挙動プロトタイプは quantity、state と呼ばれる 2 種類のパラメータを持つ。quantity はいわゆる量の概念を表すパラメータであり、量空間を持ち、大小関係を記述できる。一方、state はスイッチの on/off のように大小関係を持たず、値として記号をとるパラメータである。

挙動プロトタイプの定義の例を図 4.17 に示す。図中の例では、挙動プロトタイプ ElectricDischarging が定義されており、その上位プロトタイプは Fundamental、すなわち上位プロトタイプは存在しないことを示している。また、conditions の各小項目から本プロトタイプの成立条件が以下のように規定されているのがわかる。すなわち、挙動プロトタイプ EVoltage、ElectricCharger、ETerminal のインスタンスの存在が前提であり、これらはこの定義内ではそれぞれ voltage、charger、terminal と参照される。また、references に記述されているように、これらの間の関係と

¹influences 内の influences。

表 4.2: 挙動プロトタイプの定義

項目	小項目	内容
phenomenon		挙動プロトタイプ名
supers		上位の挙動プロトタイプ
conditions	prerequisites	このプロトタイプが成立するために必要な挙動プロトタイプのインスタンス
	references	prerequisites 間で成立している依存関係の条件
	q-conditions	quantity パラメータに関する条件
	s-conditions	state パラメータに関する条件
influences	quantities	新たに付加される quantity パラメータ
	states	新たに付加される state パラメータ
	q-relations	quantity パラメータ間に付加される定性的比例、反比例関係
	influences	quantity パラメータ間に付加される定性的微分関係
	s-influences	state パラメータ値の変化

Prototype Browser		phenomenon	ElectricDischarging
prototypes			ElectricDischarging
ElectricDischarging		supers	Fundamental
ElectricDisconnection			
ElectricRectification			
ElectricRectifier			
ElectroMagnet			
ElectroStaticForce			
EndEffector			
ETerminal			
EVoltage			
Flow			
FlowSuper			
Force			
Fundamental			
FusibleObject			
Grip			
Ground			
HarmonicDrive			
HeatGeneration			
Heating			
HorizontalRotation			
HorizontalSlide			
Housing			
Individuals			
Inertia			
Inertia2			
InertiaParts			
InertiaParts2			
InertiaPartsMovement			
InertiaPartsMovement2			
Joint			
JointTransformation			
KinematicComponent			
KinematicMotion			
LinearMotion			
Link			
Liquid			
Magnet			
MagneticAttraction			
MagneticAttractionNS			
MagneticAttractionSN			
MagneticPole			
MagneticRepulsion			
cancel			
accept			
		conditions	
		prerequisites	voltage * EVoltage charger * ElectricCharger terminal * ETerminal
		references	terminal * (terminal voltage->terminal) charger * (charger terminal->conductor)
		g-conditions	zero * {voltage ACvoltage} [(voltage DCvoltage) > zero zero > {voltage DCvoltage}]
		g-conditions	
		influences	
		quantities	chargeAmount * (-minus @zero +plus)
		states	
		g-relations	(self chargeAmount) direct (voltage DCvoltage) % (zero = zero)
		influences	
		g-influences	

図 4.17: 挙動プロトタイプの記事例

して、voltageのprerequisitesであるterminalと本プロトタイプのterminalの一致等が必要である。さらに、voltageのquantity "ACvoltage"がzeroであり、かつ、"DCvoltage"がzero以外であることが要求されている。

また、本挙動プロトタイプが生じた影響が、influences内に以下のように規定されている。まず、値として-、0、+を持ち得るquantityパラメータ"charge - Amount"が定義され、chargeAmountがvoltage内のパラメータDCvoltageに比例するという関係が付加されることが示されている。

以上に定義されたプロトタイプを用いて、図 4.18に示すようにフィジカル・フィーチャーが定義される。図中の例では、フィジカル・フィーチャー"ACPlug"がプラグに交流電圧が印加されている状態として定義されている。図中の各ノードが挙動プロ

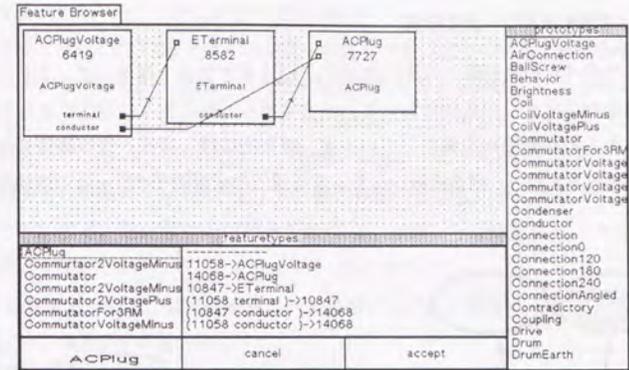


図 4.18: フィジカル・フィーチャーの記事例

トタイプ、アークは挙動プロトタイプ間の依存関係である。

さて、以上の方法により FBS ダイアグラムの機能、挙動、状態の各レベルの記事が可能になった。ここで問題となるのは挙動・状態レベルの記事における「視点」の計算機上での記述、実現方法である。66ページで述べたように、視点は機械工学、電気工学等の分野の違いと表現の詳細度、粒度の違いにより分類される。詳細度、粒度の違いにより異なる視点を実現する技術は現状では十分に発展しているとは言いがたく、今後の課題ということで今回は考慮しないことにした。また、分野の違いによる視点の区別はメタモデル・システムで実現可能であるが、現状のインプリメンテーションでは、一つのメタモデルに対して分野毎にその分野に關係する部分のみを提示するだけであり、メタモデルを直接利用し挙動を表現、推論することと本質的に違いが無いので、本研究では分野の違いを表現しないことにした。すなわち、本研究では挙動・状態レベルでは一つの視点で対象全てを表現することにした。この場合、機能階層に対応した挙動・状態レベルにおける抽象度の異なる表現が不可能になり、機械の階層的表現は機能レベルのみで表現することとした。この点は FBS モデラーの FBS ダイアグラムに対する限定である。

4.4.2 FBS モデラーと実行例

前項で述べた知識表現の方法に基づき、FBS モデラーを構築した。システムの構成を図 4.19 に示す。図 4.19 において、メタモデル・システムは現在のところ定性推論システムの役割のみを果たしているが、将来的には、多視点間の関係の管理、および、幾何モデラー、構造解析システム等各視点の定量的モデラーとの接続を管理する。

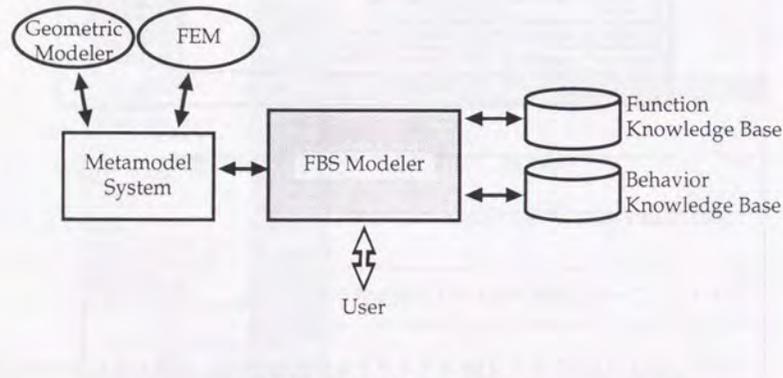


図 4.19: FBS モデラーの構成

FBS モデラーは、設計者の概念設計支援のために以下の働きをする。

1. 機能プロトタイプ、挙動プロトタイプの知識獲得

第 4.4.1 項で述べた機能プロトタイプ、フィジカル・フィーチャー、および、挙動プロトタイプを獲得し、機能知識ベース、挙動知識ベースを構築する。将来的には、機能知識の獲得方法として設計作業からの自動知識獲得等が考えられる。また、松本ら [Matsumoto91] は、設計向き大規模知識ベース構築を目的としてフィジカル・フィーチャーの収集を行なった。

2. 設計対象に関する FBS モデルの作成

この部分がいわゆる設計作業のワークスペースに相当し、設計者は機能知識ベース、挙動知識ベースを自由に検索しながら機能展開、実現挙動の選択、部分的な変更を実行する。この段階では、物理的な厳密性よりも、ブレインストーミング的に設計者の発想を支援することを目的としている。

3. 挙動シミュレーションによる一貫性の検証

前項で作成した FBS モデルの挙動・状態レベルの記述に対して挙動シミュレーションを実行し、一貫性、正当性の検証を行なう。実際のシミュレーションはメタモデル・システムにより実行される。シミュレーション結果をシステムが評価することにより、作成された挙動・状態レベルの物理的正当性、副次的に発生する現象のチェック、および、機能階層の実現性の判定を行ない、設計者に表示する。設計者のその結果を参照し、必要ならば前項に戻りモデルの変更、追加を行ない、再び、シミュレーションを行なう。

以上の操作を繰り返し、設計者が設計対象に関する十分に正しい FBS モデルを作成した時、本システムの設計は終了する。

図 4.20 に FBS モデラー上での設計支援の基本的なアルゴリズムを示す。FBS モデラーは設計者がこのアルゴリズムに基本的には従いつつ、柔軟に設計対象のモデルを構築する過程を支援することを目的としている。

次に、具体例を通して本システムの動作を述べる。ここでは簡単な例として、複写機等で利用される「ドラムを帯電させる」機能を設計する。まず設計者は、機能知識ベースを利用しながら機能展開を行ない、機能階層を構築する (図 4.21 参照)。図中で白抜きノードが機能プロトタイプのインスタンスを示す機能ノードであり、灰色のノードが機能名内の目的語を表すノードである。次に、各部分機能に適切な実現挙動 (フィジカル・フィーチャー) が存在する場合には、それを利用し挙動・状態レベルのモデルを構築する。図 4.22 は、この過程により構築された FBS モデルを示しており、挙動・状態レベルの記述は、長方形で示される挙動プロトタイプのインスタンス (挙動ノードと呼ぶ) のネットワークとして表されている。このとき、各機能ノードには、mode と condition と呼ばれるタグを付けることができる。mode タグは現在の設計対象においてその機能を発現させるために必要な初期条件、境界条件を記述する。これは、当該機能に関連している挙動記述内の部分ネットワークに対する挙動シミュレーションを行なう場合の初期条件になっている。図の例では、mode タグの内容として「AC プラグがコンセントに接続している ((ACPlug plugged) = in)」「コンセントから AC 定格電圧が出力されている ((ACPlugVoltage ACvoltage) = rated)」等の記述がされている。すなわち、mode タグの条件が設計対象上で満たされた時に目的とする機能が発現されるように設計を行なうのである。また、condition タグは現在の設計対象上で当該機能が発現されていると判定する基準を記述する。図中の例

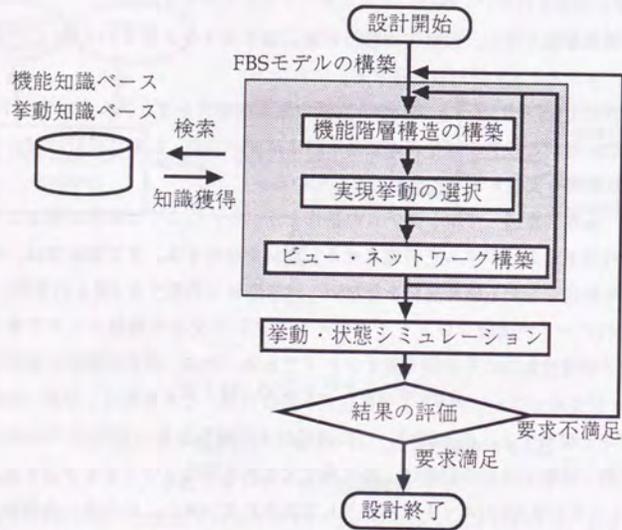


図 4.20: FBS モデラーの基本的アルゴリズム

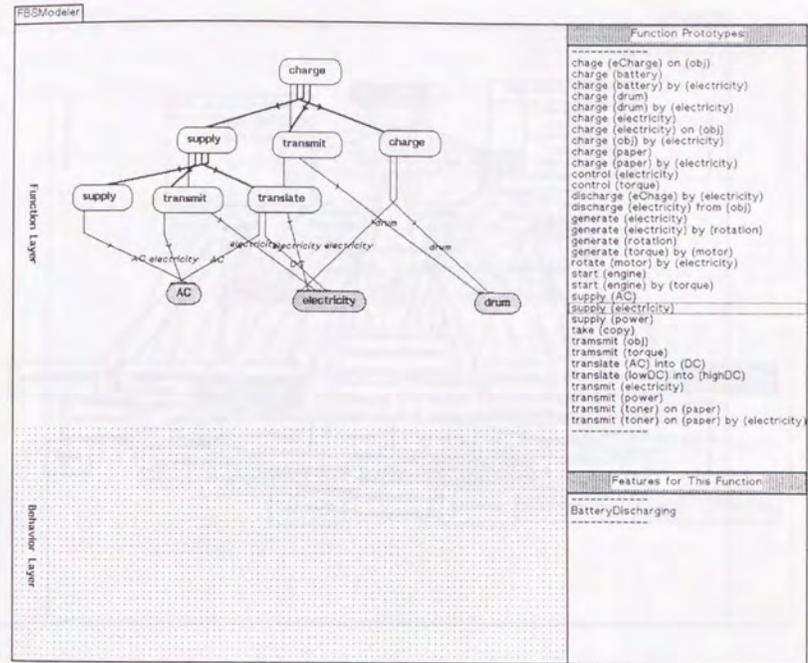


図 4.21: FBS モデラーの実行例 (1)

では「ドラムの電位が正 ((Drum -> EVoltage DCvoltage) > zero)」という条件が満たされていれば、「ドラムを帯電させる」機能が発現したと判定することが示されている。これら mode、condition タグの内容は、注目する機能に対して一般的に適用可能な条件ではなく、現在の対象というある特定の機能構造、挙動ネットワークの中での注目機能に対して記述されるものである。このため、mode、condition タグは機能プロトタイプには記述されない。

以上のようにして作成された FBS モデルに対して次に、挙動推論を実行する。挙動推論の実行方法については文献 [Kiryama91] に従う。挙動推論の結果を図 4.23 に示す。図 4.23 の機能レベルにおいて、白抜きの機能ノードは発現されている機能、黒

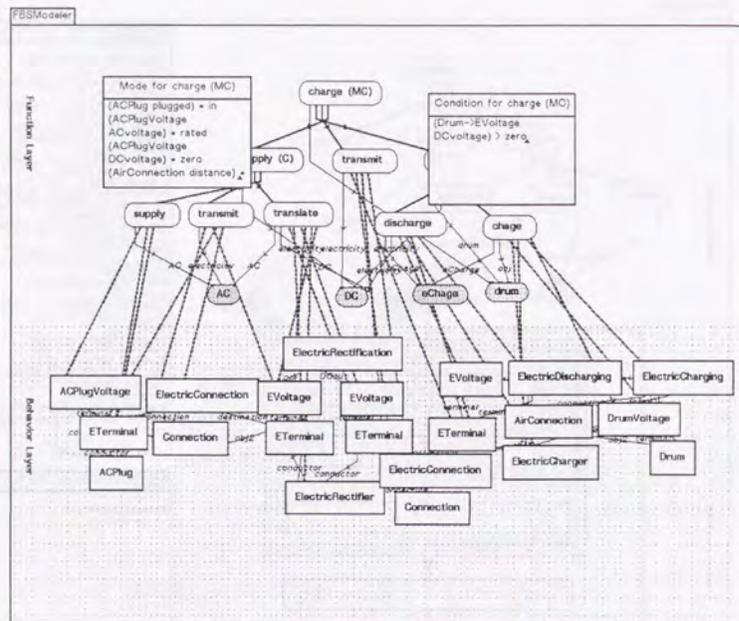


図 4.22: FBS モデラーの実行例 (2)

塗された機能ノードは発現されていない機能、斜線の機能ノードは判定できない機能²をそれぞれ表しており、その判定基準は以下に示すアルゴリズムにより決められる。

1. condition タグを持つ場合は、条件が全て満たされていれば発現されており、満たされていない場合は発現されていないとする。
2. 1. で判定できなければ、当該機能ノードが直接関係する全挙動ノードがシミュレーションの結果生成されたモデル内で全て成立していた場合は、発現されており、成立していない挙動ノードが存在する場合は発現されていないとする。

²図 4.23には判定不能の機能ノードは存在しない。

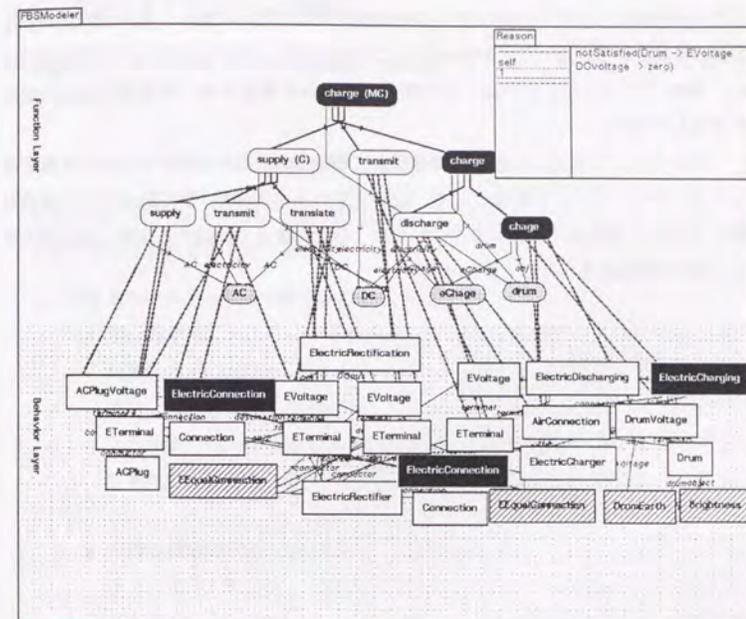


図 4.23: FBS モデラーの実行例 (3)

3. 2. で判定できなければ、当該機能ノードの全てのサブ機能ノードが発現されていれば、発現されたとし、発現されていないサブ機能ノードを持てばその機能ノードも発現されていないとする。
4. 以上で判定できなければ、判定不能とする。

挙動・状態レベルにおいては、黒塗のノードはシミュレーションの結果生起していないことが分かった挙動ノードを示し、斜線のノードは対象記述にはなかったが、シミュレーションの結果生起していることが分かったノードを示す。

図 4.23の例では、このとき目的機能「ドラムを帯電する」は発現されていないこと、および、その理由は右上のサブウィンドウに示されるように、ドラムの電位が正であるという condition が満たされていないからであることがわかる。その物理的な理由は、挙動・状態レベルを見ることにより、ドラムの明るさが正であったために、

ドラムの性質としてアースされてしまい電位がゼロになっており、ElectricCharging現象が発現されていないことが分かる³。この結果から、ドラムを暗くする機能を付加し、FBSモデルを修正すれば、目的機能「ドラムを帯電する」を実現することができることが分かる。

このようにして設計は進行し、結果として物理的に正しいFBSモデルを構築することができる。このとき情報として、設計対象の基本的配置(使用部品とその接続関係)、定性的な初期条件、境界条件、および、設計対象上で生起する現象、設計者の考えた機能階層構造を得ることができる。

³このとき、ElectricConnectionノードが否定され、EEEqualConnectionノードが新たに生成されているが、これはElectricConnectionノードが詳細化されているだけであり、特に大きな違いはない。

4.5 FBSダイアグラムの考察

本節では、FBSダイアグラムの枠組、および、インプリメンテーションに関して考察を行なう。

4.5.1 インプリメンテーションに関する考察

前節で述べたFBSモデラーの目的に関して、インプリメンテーションを通じて以下のような結論を得ることができた。

1. FBSダイアグラムにより機能を記述可能か

- 実行例で示したように、機能を記述可能であり、また、設計において機能知識を有効に利用することができた。本研究における機能知識表現の特徴は、機能自体は主観的表現でありそのみの記述には意味が無いとし、実現挙動との対応、および、展開知識による機能階層の中での注目機能の位置付けの記述により、再利用可能な機能知識の記述を実現したものである。
- 実際の機能の記述においては、Rodenacker流の入出力の変換という形が多くの場合一般的で、また、人間にとって記述、理解容易な表現であることが分かった。しかし、「ものの固定」「直線運動のガイド」「ある機能を発現させない」等、入出力の変換のみでは記述できない機能を表現する必要が出てきた。この意味で、本研究の枠組は有効であると考えられる。また、機能の表現として、否定形の表現、副詞的、形容詞的機能の表現等の柔軟性を拡張する必要がある。

2. FBSダイアグラムにより機械の階層構造を表現可能か

FBSモデラーは単一の視点でのFBSダイアグラムを実現しているので、この項目に関しては十分な考察を加えることができない。ただし、設計者が機能階層を構築する過程は十分に支援することができた。

3. FBSダイアグラムを用いて設計を行なうことが可能か。

- 前節の例に示したように、FBSモデラーを用いて概念設計、特に、設計者がブレインストーミングを行ないながら設計対象の基本配置構造と境界条件を決定する過程を、挙動シミュレーションによる評価を利用して有効に支援できることが明らかになった。

- Rodenacker 流の設計方法論と比較した場合、FBS モデラーは次のような特徴を持つ。FBS モデラーの機能展開、機能を発現するフィジカル・フィーチャーの探索という過程は、Rodenacker の設計方法論の機能展開、機能素の選択の過程の計算機化に近い。本研究のフィジカル・フィーチャーは設計者から見た物理的挙動とその関係部品の組であり、その意味で一種の機能素であるともいえる。

このような「機能素」、もしくは、Gero ら [Gero89] が提案している「設計プロトタイプ」等は設計を進めるために有効な知識であると考えられる。本研究では、このような設計に有効な部分的な対象知識を FBS ダイアグラムに基づき機能知識ベース、挙動知識ベースに収集することにより、より柔軟かつ計算可能な形で提供することが可能であり、この意味でも有効に設計を支援可能であると考えられる。

Rodenacker の設計方法論では機能素を機能階層構造通りに組み合わせることにより対象の構造を構築できるとされていたが、第 4.1.3 項で述べたように、この仮定は常に正しいとは限らない。これに対し、FBS ダイアグラムは機能階層構造に基づきつつも、フィジカル・フィーチャーを構成するビュー間の物理的因果関係により整合的に、かつ、同種のビューの同一化等の操作により、ビュー・ネットワークを単純化しつつ柔軟に挙動・状態レベルの対象表現を構築できる。また、シミュレーションによりその物理的無矛盾性を検証できるという特徴を持つ。

- 本システムでは、定性シミュレーションによる評価までを行なったが、概念設計を支援するためには、問題点を解決するための何らかの提案を行なうことが有効であると考えられる。この点に関して、シミュレーションの結果を利用して、機能知識ベース、挙動知識ベースを知的に検索する方向で本システムを拡張することにより、設計者へのより高度な支援を行なうことが可能であると考えられる。

4. 問題点の洗い出し

- ユーザーインターフェースの問題

FBS モデラーはシンボルのみで全てを表現している。これは、幾何モデラーが形状を通じて設計者と対話することを考えた場合、設計対象に関す

るイメージを引き起こすという役割においては幾何モデラーと比較して不十分であるといえる。

- 設計対象物表現の問題

FBS モデラーによる設計において、何が設計対象物であり、何が対象上で生起する現象であるのかが分からない。この情報は、下流の詳細設計や製造において重要になる。例えば、ギアを製造することや、二つのギアを噛み合わせて配置することは可能であるが、「ギアの運動伝達」現象を直接製造することはできない。これは本研究が構造と状態を区別しないこと、および、FBS モデラーが時間変化を含めた挙動を直接は表現できないことに由来する。この問題を解決するためには、FBS モデルの挙動ネットワークを実現するために最低限必要な挙動ノードと境界条件を「設計対象物の構造」という形で表現することが考えられる。

- 知識獲得の問題

本研究で示した機能知識、挙動知識の枠組で大量の知識を集めることが可能か、また、特に機能知識を再利用可能かという問題が残されている。この点は、フィールドワークにより明らかにしなければならない。

- 詳細設計の支援

本章で提案した設計支援手法は、概念設計段階における設計対象の基本的メカニズムと基本構造の設計を支援するものである。本手法により設計された対象は、さらに詳細設計等の段階を経ることによって初めて、設計を完了することができる。この下流の設計を支援する方法の開発、あるいは、下流の設計支援手法との統合化は、本研究の今後の課題である。

4.5.2 機械の階層的表現

本研究では、機械の階層的表現は人間の主観、特に、機能的階層と視点に基づいて決まることを明らかにした。この点は Rodenacker らのアプローチと同様であるが、機能的階層構造は、挙動・状態レベルの視点を定めるのみであり、実体として必ずしも機能階層構造に対応した形態を取る必要が無いという点で、Rodenacker らのアプローチよりも柔軟に機械を表現することができる。今後は、多視点性を実現する FBS ダイアグラムをインプリメントすることにより、本手法の有効性を検証する必要がある。

4.5.3 モデリング手法としてのFBSダイアグラム

58ページで述べた Functional Reasoning は本研究とその目的や方法として定性物理を使用している点で似ている。Functional Reasoning と比較して対象モデリング手法としてのFBSダイアグラムは以下の四点の特徴を持つと考えられる。

1. 挙動・状態レベルの記述として物理現象を直接的に表現、利用する本研究の手法は、デバイス主義の定性物理を用いた表現よりも柔軟である。
2. Bradshaw らの研究や Franke の研究のように特定の挙動の特徴抽出による抽象化が機能であるとは考え難い。このような表現では、機能の記述が特定の対象に依存したものとなり、機能の知識を他の対象に適用することが困難になる。FBSダイアグラムでは、機能は挙動を抽象化、および、主観的に記号化した人間にとって理解容易な表現であり、その間に一般的な法則性を見出すことは難しいと考えた。また、ある機能のある状況に適用した場合の挙動と別の状況に適用した挙動は異なるものであると考えられ、この挙動の抽象化、記号化により機能の一般化を行なうことができると考える。この意味では本研究は Keuneke の考え方に近いものである。
3. 機械の階層構造に関して、Bradshaw は主観に基づく方法、Bylander は consolidation に基づく方法を提案している。本研究では、機械の階層的表現の原則はユーザの任意性に依存する機能階層構造に基づくものの、各部分機能が視点の表現レベルや表現範囲を何らかの方法で指定することによって、挙動・状態レベルの階層表現は視点間の関係として客観的に記述、管理されるべきであることを示した。

設計とは逆に、既存の機械からFBSダイアグラムを構築する方法、すなわち、使用者が機械の働きについて理解する過程を支援することは、設計対象モデリングと比較して対象が存在物である点が異なるのみであり、FBSモデラーで有効に支援できる。この場合も機能が主観的である以上、自動的にFBSダイアグラムを生成することは考えられないが、機能知識ベース、挙動知識ベースを有効に利用することにより、効率的にモデルの構築を支援することが可能になる。

4.6 自己修復のための機械のモデリング

本節では、本研究の目的である自己修復機械の設計、および、運用のために有効な機械のモデリング手法について考える。まず、自己修復機械のモデリング手法として必要な項目を整理する。

1. 制御型自己修復機械を設計するためには、機能の表現、および、アクチュエータの制御による機械の状態と機能の発現状況の変化をシミュレートできなければならない。
2. 機能冗長型自己修復機械を設計するためには、機能の表現、および、機能の冗長性の探索を実現可能でなければならない。
3. 保全の特徴である不定性に対応するためには、故障発生後の対象の状態、挙動を推論可能でなければならない。
4. 保全の特徴である多様性に対応するためには、様々な視点からの機械のモデリングが可能でなければならない。
5. 保全の特徴である非繰り返し性に対応するためには、一般的な原理原則的知識を収集し、物理的世界を計算機上に構築し、未発生の故障を推論可能でなければならない。

以上の要求を満たすために、本研究では、前節で述べたFBSダイアグラムによる機械表現を基本的なモデリング手法とする。これにより、1.に関してはFBSモデラーとその挙動シミュレーション機能により要求を満たすことが可能であり、2.に関してもFBSモデラーと機能知識ベースを利用することにより実現可能である。3.、4.、5.に関しては第3.2節で述べたように定性物理を用いることが有効である。さらに、保全作業自動化のための機械のモデリングという観点から定性物理を利用する理由として以下の二点を挙げるができる。

1. 保全作業、特に故障診断作業において、対象系から常に意味のある定量的な情報を得られるとは限らない。逆に、点検作業において意味のある値というのは、正常値等の基準となる値よりも大きい、小さいという相対的な値の関係に意味があることが多い。定性物理を用いることにより、このような曖昧であるが意味のある情報を直接表現、利用して故障診断作業、修復作業を行なうことが可能になる。

2. 第2.1.1項で見たように、故障診断技術者が未知の故障を診断する場合に対象系を理解する方法と、定性物理の対象系を因果関係と定性的な値を用いて表現する方法が似ている。このことより、保全技術者の知的判断過程を定性物理によりモデル化できる可能性がある。

ここで注意しなければならないのは、設計時の機械の表現と保全時に必要になる機械の表現が必ずしも一致しないことである。設計時には、第4.3.4節で見たように機能や挙動のレベルの操作を中心として機械の挙動や状態を組み上げてゆく。このため、プロセス主義の定性物理に基づきフィジカル・フィーチャーを用いて挙動や状態の自由な組合せ、結合を支援することが有効である。一方で、第2章で見たように保全作業時には正常状態との違いが問題となるため、対象機械の正常な機能、挙動、状態の情報を保持しておき、そこからの変化を推論可能にしなければならない。このとき、正常時の対象の挙動や状態はあらかじめ決まっているため、必ずしもプロセス主義の定性物理で表現する必要はなく、逆に、デバイス主義の定性物理で表現するほうがユーザにとって理解容易で、かつ、変化を明示的に表現できるという特徴を持つ。そこで、本研究では、設計段階ではプロセス主義に基づくFBSモデルを用い、保全作業の段階では、デバイス主義に基づくFBSモデルを用いることにする。このとき、保全作業の初期段階で与えられる部品のネットワークは対象機械の初期状態での「構造」であると考えることができる。

しかし、デバイス主義の定性物理では、物理現象を直接表現したり、構造が変化するような現象を扱うことができない。そこで本研究では、次章で述べるように物理現象に関する知識を直接記述可能なようにデバイス主義の定性物理を拡張して利用する。

このように設計時と保全時ではその対象表現形態が異なる。また、機能階層表現も設計時のモデルと保全時のモデルでは一般に異なっていると考えられる。設計時のモデルは設計者観点から構築された機能階層を持ち、保全時のモデルには保全作業に適した機能階層が記述しており、これらは完全には一致しない。このため、FBSモデラーで作成された対象モデルを直接保全時に利用するというわけにはいかない。これらの間の変換は半自動的に実行可能であると考えられるので、この変換を支援し、CADと保全システムのモデルによる統合化を実現することは今後の課題である。

4.7 第4章のまとめ

本章では、機械のモデリングを行なう方法について議論した。まず、既存のモデリング手法を概観し、機能の表現、取り扱いに問題があること、および、挙動のモデリング手法として定性物理が有効であることを明らかにした。次に、サーベイに基づき、その問題点を解決するためにFBSダイアグラムを提案した。FBSダイアグラムでは、主観的な機能の記述と客観的な挙動、状態の記述を区別し、かつ、互いに関係させることで有効な対象表現を行なうものである。また、機械の階層的な表現を実現した。さらに、FBSダイアグラムを概念設計支援に適用したFBSモデラーを構築し、FBSモデラーにより機能を含めた対象表現、概念設計の支援に有効であることを明らかにし、その問題点を検証した。最後に、本研究で用いる機械のモデリング手法として、設計時にはプロセス主義の定性物理を用いたFBSモデルを、保全作業時にはデバイス主義の定性物理を拡張したFBSモデルを使用することを明らかにした。保全作業時の対象表現の詳細については、第5章で述べることにする。