

故障修理と修理用知識の獲得に関する研究

通信工学専攻

610101 秋吉恒一郎

指導教官 岡本孝太郎教授

平成2年2月13日

目次

1	はじめに	1
1.1	研究の目的と位置付け	1
1.2	論文の構成	2
2	故障修理の支援と故障診断	3
2.1	熟練技術者と理論技術者	3
2.1.1	熟練技術者と理論技術者との対話	3
2.1.2	作業場の基礎知識	5
2.1.3	修理用知識	5
2.2	熟練技術者の知識と故障診断	6
2.3	作業場の基礎知識獲得のために	7
2.4	熟練技術者と理論技術者の対話の特徴	8
2.5	まとめ	9
3	故障修理の実際	11
3.1	対象の特徴	11
3.1.1	電気的な面	11
3.1.2	物理的な面	11
3.2	故障の種類と故障の数	13
3.2.1	故障の種類	13
3.2.2	故障の数	14
3.3	故障の原因を作り出す要因	16
3.3.1	部品不良の要因	16
3.3.2	導通不良の要因	17
3.4	部品不良の例	18
3.5	症状の現われ方	19
3.5.1	症状のとらえ方	19
3.5.2	機種ごとの症状	19
3.6	作業の進め方	21
3.6.1	症状と原因	21
3.6.2	断える症状	21
3.6.3	状態把握	22
3.6.4	故障診断	23
3.6.5	意識外の計測、動機付けのない計測	24
3.6.6	並列的不良、連鎖的不良がある場合の故障診断	25
3.6.7	作業上のテクニック	25

3.7	まとめ	26
4	修理用の知識の獲得	27
4.1	修理の時使われる知識	27
4.1.1	領域の設定	27
4.1.2	事実、状態、症状	29
4.1.3	原因と中間原因	30
4.1.4	計測の定義	31
4.1.5	症状・状態パターンと原因	31
4.1.6	チェックリストとモデル形成	32
4.2	計測カードによる計測整理	33
4.2.1	計測カードの導入	33
4.2.2	PDSサイクルと計測	33
4.2.3	カードおこしの手順	34
4.2.4	カードとエディタ	36
4.3	大局的・局所的診断とその知識	37
4.4	まとめ	39
5	獲得した知識の表現と実装	41
5.1	知識の表現	41
5.1.1	相による分割	41
5.1.2	チェックリスト	41
5.1.3	局所的診断相の知識記述	42
5.2	知識の実装	43
5.2.1	テスト環境	43
5.2.2	チェックリストとアイテムリスト	44
5.2.3	モデルとフレーム	47
5.2.4	原因リストとルール	47
5.2.5	テスト結果と応用	50
5.3	まとめ	53
6	結び	
6.1	研究の到達点	54
6.2	今後の課題	55

謝辞

参考文献

付録

1 はじめに

1. 1 研究の目的と位置付け

「故障修理」という仕事は、技術が発展し複雑化していく今日のあらゆる産業界において重要なものとなってきている。技術の発展が、その性能維持のための技術のある面をないがしろにして進んでいくという傾向が、分野を問わず少なからずある。

例えば、電子機器という分野を見ても、それは歴然である。現実には相当な複雑さをもった機器が世にあふれ、輸出されているわけであるが、それらの性能を十分に維持していくだけの技術が、伴っているかというところではない。確かに設計、生産時における技術的な考慮により、故障率というのは低く抑さえられ、信頼性の高いものを作ろうという努力はあるものの、機器の絶対量と複雑さは、もはや人間技術者が適当な測定機器のある、簡単な作業場で保守、修理できる量、程度を越えている。これは、設計がすでに複数のブロックに分けられ、複数の人間によってなされているという現実から見ても当然のことといえる。機器のどのブロックが異常であるかを決めるためには、一人の技術者が全体の動作を把握しながら進めなければならないため、複数の人間によって設計がなされたものであっても、複数の人間が同時に一つの機器を前にして修理を進めるのは困難である。

こういった故障修理に関する技術は、経験的なあるいは理論的な知識の集合であると、知識工学的にはみなされている^[1]が、それらの性質や獲得法に関する問題は対象や問題領域に固有のものであり、一般化することが難しい。また優秀な修理技術をもつ技術者も、普通、それを第三者に伝える技術や手段を持ちえないことも知られている。そのため、経験的な知識がまさに経験によって得られているだけで、技術者間の知識、情報の交換・継承が必ずしも能率的には行われていない。

一方、知識の受け皿として汎用な知識表現言語を研究している知識工学者の立場からしても、実験実証のためには、相当な複雑さを持つ専門知識が必要となる。ところが、現実には「知識が得られたら」という前提で枠組みを作っていく傾向もあるため、言語そのものだけが一人歩きしてしまう可能性もなくはない。^[2]

修理技術の複雑化に対応するためには技術者（専門家）からの知識を効率よく引き出し、それをしかるべき形で知識として表現できるようになることが必要であると私は考える。専門家は知っていることを表現して誰かに伝えたくてもできない、知識工学者は専門家の協力のもとに専門家の知識が欲しい。この二つのジレンマとでもいうべき事実関係が、いわゆるボトルネックといわれている^[3]ものである。本研究の目的は、このネック部分の解消のために何をすべきかという視点から、私の約10か月におよぶ、あるオーディオ機器修理工場の実習経験をもとに、

1. 実際の故障修理はどのように行われるか、
 2. どのような方法で修理のための知識を体系的に獲得できるか、
- ということを解明し、
3. 獲得した知識の有効性を確認する、

ことを行うことである。

1. 2 論文の構成

2章では、故障修理支援システムを仮定した場合どのような機能が要求されるかを想定しその中で故障診断をすることの役割を認識し、故障修理のための知識獲得の必然性について論ずる。

3章では実際の修理がどのように進められていくかをふりかえりその特徴を抽出し4章への導入を与える。

4章では修理の時に使われる知識のとらえ方とそれらの知識を体系的に整理する枠組みについて論ずる。

5章では獲得した知識を計算機上に実装する場合の知識の表現方法を考え、知識ベースシステム構築用ツールを用いて知識を実装してテストする。そしてそのテスト結果について述べる。

6章では、まとめて今後の課題について述べる。

2 故障修理の支援と故障診断

修理のための技術といっても実際にはかなり広範囲なものを含んでいる。

電子機器を対象にするわけであるから、電子回路の知識を持ってるといふことはいうまでもないにしても、それだけで即その人が修理のできる人になるということにはならない。例えば、教科書で電子回路について十分に習熟していたとしても回路図上でのそれぞれの部品がどのような形をしているとか、極性などに関する知識がなければ、部品のとりつけや、とりはずしは困難をきたすであろう。また測定器の使い方というものを知っているかどうかという問題も、修理を始めた時にすぐさま直面する問題である。

ここでは、修理のできる技術者とはどういうものか、修理のできない技術者にはどういうものを与えれば修理ができるようになるのかを考え、故障修理の支援システムを仮定した場合に、何を支援システムに期待するか、そしてその中の一つの大きな支援になるべき故障診断との関係について論ずる。〔付録1〕

2. 1 熟練技術者と理論技術者

きわめて曖昧ではあるが、「電子回路の技術の基礎を一応習得した技術者」というのを想定してみる。以後この種の技術者を便宜上『理論技術者』と呼ぶ。これに対して実際の修理の仕事に相当期間携わりその専門領域における経験的知識、理論的知識両方を持つ技術者を『熟練技術者』と呼ぶことにする。理論技術者は、実際の機器に修理という立場からは係わったことがなく、部品や測定器についても理論的な意味でしか学習したことがないとする。

理論技術者は電子回路技術について話せばわかるタイプの技術者なのであるが、修理をさせようとする、持っている理論的知識の適用の仕方がわからなかったりすることが多いのである。こういった理論技術者がいる場合、彼に何を与えどのように支援してやれば修理できるようになるのであろうか。熟練技術者と理論技術者との対話をふりかえり、どのような知識が伝授されるのかを考えてみる。

2. 1. 1 熟練技術者と理論技術者との対話

前述の支援システムを想定したとき、そのモデルとなるのが、熟練した修理技術者と理論技術者との関係である。理論技術者が疑問を持ったり困難に遭遇したりしたとき、熟練技術者はどのようなことを教え指導していくのであろうか。私は両者間の対話、係わり合いが支援システムでの支援内容にもっとも重要な内容に相当すると考える。

そこで現実の係わり合い、対話を思い出し、その知識の性質から次の2つに分類してみた。

1. 作業場の基礎知識 [部品、測定器、測定法などに関する知識]

抵抗、コンデンサといった基本的な部品の性質、形状取り扱いから測定器の使い方まで、作業場で常識として知っておかねばならない性質の知識である。ハンダごてやハンダ吸い取り器といった道具の使い方も含む。

この知識が直接、故障原因をつきとめたりすることはできないが測定や作業の手段としてなくてはならないものである。

2. 修理用知識 [故障原因をつきとめるために用いる知識]

修理のための知識というとなら、1.の作業場の基礎知識も含むように聞こえるが、ここではそれ以外のもの、すなわち故障原因をつきとめるために用いる知識をさすことにする。

知識工学的見地から『経験的知識』、『教科書的知識』、『内部知識』、『外部知識』という分類 (11, 13) を上に述べた2つの知識に適用すると次のようになる。○は主に持つ性格、△はときに見られる性格を示す。

	経験的知識	教科書的知識	内部知識	外部知識
作業場の基礎知識	△	○	○	○
修理用知識	○	○	○	△

作業場の基礎知識の中には例えば、『オシロの使い方』といったものもある。これは経験によって得られたものではなく、測定器に普遍のものであり、いわゆるマニュアルに書かれていることに相当する。熟練技術者であっても細部にわたる仕様や使い方まで覚えているとは限らないからこれは内部的な場合と外部的な場合がある。

修理用知識の中には「この状態の時にはこういう原因が考えられる」というような経験的知識がある。しかし常に経験的知識だけで問題解決をしているとは限らない。時には理論的な解析をしているわけだから、教科書的知識をも使っている。これらは普通人間の内部(頭の中)にあるものであるが、ある修理例のメモをとっていたりしてそれを後に参照しているとすれば、外部知識にもなる。

2. 1. 2 作業場の基礎知識

(1) 部品に関するもの

回路図上での部品記号が現実の部品ではどんな大きさ形になっているかを知っているか否かは部品を認識する上で大変重要である。抵抗といえどもその誤差、値、耐電力値などといった基本的な違いだけでなく、カーボンか金属皮膜かセメントかというような違いを認識できるかということも重要である。トランジスタ、IC、コンデンサなどは電極の名称、機能、極性といったものが現実の部品でわからなければならない。

(2) 測定器、測定法に関するもの

「～の電圧を測定せよ」とか「～の電圧波形を観測せよ」といわれても、理論技術者はどのように測定してよいかわからないことがある。「どうやって測定してよいかわからない」という現象は「測定器の使い方がわからない」というタイプのもので「測定方法がわからない」というタイプの2つに大別できる。測定器の使い方は概して「マニュアル」といわれるものを調べればわかるが、「トランジスタの検査方法」というのは「トランジスタチェッカ」といった専用測定器でない限り普通、測定器のマニュアルにはのっていない。熟練技術者はテスタ1台で簡単に良否を判別するが、その時に使う知識というものが「測定法に関する知識」に相当する。

測定器の選択に関しても知識を必要とすることがある。修理の状況によって測定器は使い分けられなければならない。例えば電圧測定一つにしても可動コイル型電圧計が良い場合、デジタル電圧計が良い場合、電子電圧計が良い場合、オシロで見た方が良い場合などさまざまである。「直流にわずかな交流が乗っていて交流分を無視して直流だけを測りたいときデジタル電圧計より可動コイル型電圧計の方が読みやすい」というような知識はその一例である。〔付録2〕

2. 1. 3 修理用知識

理論技術者が何らかの形で前述の作業場の基礎知識を得て自由に使いこなせるようになったとする。熟練技術者が「～の電圧を測定せよ」といえば、理論技術者はどうにかこうにかして、つまりマニュアルを見たり、教わった基本的な測定方法や部品の知識を利用して求めたい測定結果が得られるようになったとする。

ところが、この時点でもまだ理論技術者には修理ができない。環境は整ったものの、どこから手をつけてよいかわからないからである。この段階で熟練技術者は理論技術者にどのように修理を進めるかを提示したり、故障原因をつきとめるための判断や計測の手順を示してやれば、理論技術者は修理を進めることができる。

2. 2 熟練技術者の知識と故障診断

前節で理論技術者と熟練技術者の関係について論じたように理論技術者が修理できるようになるためには、『作業場の知識』、『修理用知識』の2つの知識を身につければ良いことになるが、この2つの知識と故障診断との関係はどのようになっているのだろうか。

修理用の知識は「故障修理の原因をつきとめるための知識である」としたが、これはいわゆる故障診断になくはならない知識であり、故障診断のための知識といってもよい。すなわち熟練技術者は修理用の知識を用いて故障診断をしていると考えられる。

一方、作業場の知識は熟練技術者の頭の中ではどのようにとらえられているかという点、ほとんど重要な事としては意識されていない。基礎的な測定法、部品の知識はすでに頭の中に入っているし、少しこみいった内容、例えばオシロの外部トリガのかけ方とか測定器の仕様、ICのピン番号と機能というようなことは「～のマニュアルを見ればよい」といういわゆる外部知識のポイントという形で持っているので、実際には知らなくても『知っている』という意識の中に位置している。

熟練技術者の意識の中では「どこが悪いのかをできるだけ能率よく的確に見つけるという」大テーマだけが主に存在し、『作業場の基礎知識』は現実の重要度とはうらはらにあまり重要なものとはみなされていない。

支援システムをもう1度想定した場合、これら2種類の知識は共に重要な骨格となる知識であると予想できる。ところが、『作業場の基礎知識』はデータベース的な性格が強く必要な時に参照できればよいというものであるのに対して、『修理用知識』は明らかにそのような性格だけでは表現できない。

私は1つの支援システムの構成として「修理用知識に作業場の基礎知識がぶらさがった形」というのを考えている。中心となる骨格は「どこが故障しているか」という問題解決であり、それを支援するにはまず、「修理用知識」の体系的な獲得、そしてそれを利用した故障診断であると考えられる。『作業場の基礎知識』はその骨格に付随した情報という位置付けでとらえる。

本研究では、両知識が相互に作用して支援を進めうるということを予想しながらも、後者の『修理用知識』が『獲得』という面から特にそのめんどろさを持っていると感じるので、以後『作業場の基礎知識』とは完全に切り離して考え、3章で実際の修理の特徴、4章で『修理用知識』の獲得法、5章で獲得した知識の実装とテストについて論ずる。

2. 3 作業場の基礎知識の獲得のために

先に「作業場の基礎知識はデータベース的な性格が強く、比較的単純である」といったが、実際には知識を整理しようとする時、問題となると思われることがいくつかあるので、本研究では扱わないが、簡単に述べておく。

まず、「技術者のどのレベルまで支援するか」ということを考えなければならない。ある程度部品のことや測定のことをわかっている技術者に簡単すぎることを教えたり用意する必要はない。逆に熟練技術者といえども、マニュアルやデータブックは手元にほしいわけである。したがってどの程度の情報をどのくらいデータベースとして持ち、必要な時に提示してやるかは支援する技術者のレベルによって変わってくる。ある程度ターゲットとなる技術者をせばめるのか、それともすべての情報を持つのかを支援システムの運用方法、位置付けとともに考えなければならないだろう。

次に、データをすべてシステムに持たせるかどうかである。規格表などがデータベースになっていれば確かに検索には便利であるが、どれだけ慣れて使えるかはインターフェースの問題をも含めてよく考慮されなければならない。時代の大きな流れとしていろいろな情報が『紙』以外の媒体で供給されるようになってきているが、現在はまだ、一般に図や表は紙にかかっている方がわかりやすいという人が多いのではないかと思う。今後、回路図1つにしても、紙かCRTがどちらがよいかは、場面、利用の仕方によって使い分けが明確になされるであろう。プロトタイプ程度の支援システムであれば、「～は〇〇のマニュアルにのっている」「～の規格表を見ろ」といったポインタとしての知識で十分であろうと思う。

支援システムの想定のところではふれなかったが、回路図と実際の部品の対応というのが支援されていれば修理は非常に楽になると予想できる。回路図上で故障箇所を検討してある部品の検査をしたくなるとする。このときその部品が割とユニークであれば、探すのにそれほど困難をきたさないが、同じような部品ばかり並んでいてしかも値も同じだったりすると目標の部品を認識するまでけっこうな労力と時間を要する。最近では電子機器の小型化に伴いチップ抵抗、チップコンデンサなどが多用されているので、そのような場面では特に目がチカチカしてわかりづらい。

回路図-実体部品図の対応を示すことの有用性は広く認められているのであるが、支援システムだけのためにこの環境を構築しようとする、部品点数が多い場合データ量が膨大となり単純に保守のために割り当てられる予算では割が合わないためか現場ではなかなか見られないのであろう。しかし、もはや現代では設計がCADであることも多く、プリント基板の設計も当然CADなわけであるから、そのへんのデータをうまく処理・活用できれば支援システムだけの環境作りではないので、少ない労力で実現できるはずである。こうした場面での設計と保守の分野の連携は非常に重要だと思う。

2. 4 熟練技術者と理論技術者の対話の特徴

熟練技術者と理論技術者との対話における特徴について少し触れておく。修理用知識獲得の項でも詳しく述べるが、『作業場の基礎知識』の獲得においても無視できない『熟練技術者』の『理論技術者』との係わり合いに関する特徴がある。

一般に、多くの経験的知識を持つ専門家は、自分の持つ知識を客観的に表現したり人に伝えたりするのが下手であること¹¹⁾が知られている。熟練技術者についても同様のことがいえる。概して熟練技術者は理論技術者に対して不親切である。「～の状態を調べたい」と思った時、熟練技術者はそれに必要な手段・方法をわかっているわけだが、それを理論技術者も既にわかっているとみなしがちである。自分が何の苦もなく使っている知識はどんな技術者にとっても常識であると思ってしまう。そのため、理論技術者に細かな作業方法まで懇切丁寧に教える必要があるとはすぐには思わないのである。その結果、理論技術者は仕方なく何度か失敗してそして経験的にそれを得るのだが、熟練技術者にはとっくにわかっていた、などということが現実の作業時にはよく起こる。

熟練技術者が「これはいかん。きちんと教えてやらねば。」と思うこともあるのだが、そういった場合も熟練技術者は自分にとって『ポイント』となる所しか示してやれず、エッセンスだけ伝えようとする。熟練技術者は本当に重要な点だけをヒントとして与えているつもりなのだが、そうしたいわゆる『気のきいた言葉』というのは理論技術者にとっては、いろんな経験を積んでいないため、『全体の中での重要点』という位置付けでとらえられない。そのため当座の問題を解決するための処方に対してはあまり助けとならないのである。次のようなチンプンカンプンな対話はその例をよく示している。

熟練技術者：「～の波形をオシロで観測しろ。」

理論技術者：「どうやってやるのでしょうか。」

熟練技術者：「トリガは別にかけたほうがよい。OSCから直接オシロの外部トリガ端子に入力しろ。」

理論技術者：「トリガって・・・ああこれですね。でも入力がBNC接栓になってますけど」

理論技術者はオシロのレンジ (V/cm , $Time/cm$) やプローブの使い方について実践的なことをあまり知らなかった。先の熟練技術者の『気のきいた』助言はそれらがわかって始めて意味を持つものなのである。

2.5 まとめ

電子機器の故障修理を行うためには、ただ電子回路の知識だけを持っていればよいというものではなく、『修理』という領域に固有な知識を必要とする。修理を多く経験してきた技術者は、理論的な知識だけでなく修理に独特な知識、症状から原因を予想するといった経験的な知識あるいは部品、測定機、測定方法などに関する知識などを持っていて、それらをうまく使いこなして修理をしている。本研究ではこのような知識を持つ技術者を『熟練技術者』と呼ぶことにし、電子回路の知識だけを持っていて修理には携わったことがないという技術者を『理論技術者』と呼ぶことにした。

理論技術者は一人では修理ができないが、すでに理論的な知識は持っているので熟練技術者の助けがあればなんとかできると考えられる。そのとき熟練技術者が何を教えるかを考えることは、コンピュータ上で『故障修理支援システム』というものを構築しようとする時の大きなヒントとなる。熟練技術者は専門家にありがちな性格を受け継ぐことが多く、一般に理論技術者に対して不親切で効率よく何かを説明する術をあまり持たないことが多いが、修理に関して多くの知識をもっている。したがって、このような熟練技術者から修理に関する知識を引き出して支援システムを構築することは大変有意義であると考えられる。

熟練技術者が持つ知識というのは多岐にわたっているが、本章では支援に役立つ知識として『作業場の基礎知識』と『修理用知識』の2つを考えた。

『作業場の基礎知識』は、作業をする所でいわば常識となっている基本的な知識のことで、部品に関する知識、測定器・道具の使い方に関する知識、部品測定に関する知識、電子回路に関する（実践的な）知識などがある。これらは知っていてあたりまえと思われるので意識もされないことがあるが、計測を行う上でなくてはならないものである。作業場の基礎知識はすべて熟練技術者の頭の中に入っているわけではなく、『測定器のマニュアル』というように外部知識のポインタという形で蓄えられていることもある。また「トランジスタ～の検査は～でやるとやりやすい」のように経験的なものもある。

『修理に関する知識』は上に述べたもの以外の知識で、修理の時に知っていなければならない知識、すなわち故障原因をつきとめるための知識である。修理をどのように進めるか、何を測定し、何を得てそれをどのように判断するか、そしてどのように故障原因を見つけるか、そのような知識である。

故障修理を支援する方法はいろいろあり、『作業場の基礎知識』を効率よく提供するだけでも、それが不足している理論技術者にとっては大きな助けとなる。また『作業場の基礎知識』をある程度持ち合わせている理論技術者であれば、『修理用知識』だけでも役立つ。しかし『故障修理支援システム』が作業場において強力な道具であるためには、『作業場の基礎知識』と『修理用知識』両方を持ち互いに有機的に結びついた形であるべきだと考える。そしてその1つの位置付けとして『修理用知識』は主に故障を診断するために用いられ、その過程で計測を行う時に『作業場の基礎知識』が参照されうる、ととらえる。

『作業場の基礎知識』はデータベース的な性格を多く持つため、たとえその知識獲得に困難な部分があってもそれをを用いた支援環境を作りやすいのに対して、『修理用知識』は

より多くの複雑さを持つと考えられる。いずれにせよ故障診断という機能が支援の中で重要な位置を占めることは間違いなく、『修理』という領域をよく調べ体系的に『修理用知識』を獲得することが必要である。

3 故障修理の実際

2章では『作業場の基礎知識』と『修理用知識』の定義を行ったが『修理用知識』を獲得するために、まず実際の修理がどのように行われるかをふりかえってみる。

3.1 対象の特徴

私が経験した修理の対象はオーディオ機器で、主にプリメインアンプ、パワーアンプ、チューナ、CDプレーヤ、カセットデッキとカーステレオであった。

3.1.1 電気的な面

対象を電気的（電子回路的）に見た場合、扱っている周波数は直流からVHF帯まで、電圧はパワーアンプ類で最高±50Vぐらいであった。回路の種類をアナログ・デジタルと分類した場合、時代の流れが『デジタル化』していることから予想がつくが、以前のようにオーディオ＝アナログ回路というふうにはいかず、デジタル回路の占める割合がかなり多い。それにCDプレーヤとなると原理的にも明らかなようにデジタル回路が大半を占めている。ただしCDプレーヤといえども、デジタルの信号として復調されるまでは『符号を取り出すためのアナログ回路』であり、かなり高速（高周波）な信号をアナログ回路で処理している。

カセットデッキにおいても機械部分（メカニズムという言葉から以後『メカ』ということにする）のオペレーションや録音・再生の切り替え、イコライザ・バイアスの切り替えなどがほとんど電子的に行われているため（電子スイッチ）、デジタル回路が多い。

3.1.2 物理的な面

カーオーディオの場合比較的小さく軽量であるが、プリメインアンプもクラスが上の方になると大型で重くなる傾向がある。重いものだと数10kgのものもある。対象の重さは人間がそれをとりまわす上で、作業能率の善し悪しに特に影響する。重いアンプを測定の際に裏返したり横にしたりするのは大変な労力がかかるものである。

密度という点から見ると、最近回路を集積してLSI化して部品点数を減らしている（同時に信頼性も高まっているが）のはもちろんのこと、部品の配置・基板の配置も適当に考えられムダのない空間利用となっている。これはセットのコンパクト化という意味でユーザにとっては都合の良いことであるが、修理・保守という立場からは部品の測定・交換時に特に注意を払わねばならない要因を作り出している。ある基板をはずしてからでないと、別の基板が見えない、つまり測定や部品交換ができないということがよくある。デッキやCDプレーヤの場合メカ部分を取付けて動作させながら測定しなければならないこともある。そういう時、部品が接近していたり、やりづらい位置にあったりすると測定の

時プローブなどでうっかり部品やプリント配線のパターンを短絡させないように気をつけなければならない。交換時に近くの部品に損傷を与えたりしないようにも気を配らなければならない。

大容量のコンデンサを持つパワーアンプでは、不用意なパターンショートにより装置の電源を切っているにもかかわらず、電解コンデンサにたまっていた電荷が大きいため部品に過大な電流を流してしまい、部品を壊してしまったことも何度かあった。

高密度化という現象は修理技術者に今まで以上に細心な注意を払わせ、技術者に器用であることを要求している。これらは『作業場の基礎知識』として私はとらえるが、修理を進めていくうえで、その順序を制御している知識に多大な影響を与えているので、『修理用知識』獲得の項でも触れる。

3. 2 故障の種類と故障の数

熟練技術者の頭の中では故障箇所というのは『故障診断のために』かなり合理的に（つまり故障原因が何であるかをつきとめる過程をふりかえった時その意味付けがもっともらしく）とらえられているのであるが、客観的に取り出してみると、数や種類の定義づけがよい加減であることが多い。例えば、「Aが故障して部品不良となるとBが部品不良となる」ということがよく起こり、今その現象と部品不良を発見したとしても、彼の意識の中では2箇所の故障とはとらえられていない。「Aが悪い」としか思っていないのだ。「Bも確かに壊れているじゃないですか」ときくと「それは当たり前だ。交換すればいいだけだ。」という答えが返ってくる。熟練技術者が故障の種類と数をどのようにとらえているのかを考察してみる。

3. 2. 1 故障の種類

故障の原因というたいてい部品の不良を思い浮かべるが、必ずしも部品不良だけではない。どんな種類の故障かをいろいろな観点から見ると次のようになる。

・電気的故障か物理的故障か

電子回路だけで構成されているセットであれば、故障の原因は回路の中のどこかにあると考えてよい。ところが、メカを持つ物、例えばカセットデッキ、CDプレーヤーではメカ部のどこかが不良である可能性もある。たいていは症状からどちらに原因があるか大別できるが、CDプレーヤーの場合、例えば『音飛びする』という症状だけでは故障原因が回路部にあるかメカ部にあるかはすぐにはわからない。

・部品不良か否か

「抵抗が焼損してオープンになっている」とか「トランジスタのコレクタ・エミッタ間がショート」というのは明らかに単体の部品不良で、抵抗やトランジスタそのものを修理することはない。スイッチの接点不良やボリュームの摺動子の接触不良というようなこともよく起きる。接点復活剤などを吹き付けて一時的に回復させたりするようなこともあるが、修理・保守の場面ではたいてい信頼性確保のため部品ごと交換する。「ボリュームを修理する」ということはまずない。こういった意味で修理可能な部品の不良も含めて「その部品を交換すればなおる」という観点から故障の原因は部品不良にあるとすると、他にはどういう原因があるだろうか。

私の経験では次のようなものが頻繁に見いだされた。

- (1) 導通不良
- (2) 調整不良
- (3) 取付け不良

(1) にはパターン切れ・配線不良・半田付け不良などがある。どれも「導通のあるべき所で導通がない」という事実において同じ性質である。回路図上で部品記号間を結ぶ線を普通、『部品』と認識する人は少ないので、前述の『部分不良』というわくには入れがたい。もちろん「重要な信号伝達のため特に指定したワイヤで接続せよ」という場合には、そこにはれっきとした部品の概念があるが一般にはプリント基板のパターンやリード線が切れていたりしても部品の不良とはみなさないと考える。

配線不良の中にはコネクタ類の接触不良も含むとする。

(2) は電気的な調整、機械的な調整がある。調整する場所としては、前者が半固定抵抗、コイル、後者がセンサやスイッチの取付け位置などがある。これらの調整不良によりいろいろな故障の症状が起きる。アンプではバイアス電流の調整不良からバランスが崩れて暴走が起きトランジスタを破壊してしまうこともある。CDプレーヤでは主にトラッキングやフォーカスの調整不良のため音飛びが起きる。

(3) は(2)に含めることもできるが、機械部品のグリスアップやトランスの取付けなおしなどの処置をする時の故障原因としてとらえた場合「取付け位置の調整の必然性を本来持っている(センサなど)」というよりむしろ「取付け方が悪かったため」だと考えられる種の故障原因である。

3. 2. 2 故障の数

故障原因の種類にはいろいろなものがあることがわかったが、故障の数について熟練技術者はどのようにとらえているだろうか。

この節では、導通不良・調整不良・取付け不良の3つも広い意味での部品不良と考えて(つまりそこが治ればよいという意味で)故障の数をどのようにまとめて感じているかを述べる。

(1) 熟練技術者が1箇所の不良とみなす場合

(ア) 1つの部品が不良

単純に1つの部品が不良でそれによってある症状は出るが他の部品、ブロックを壊したりはしない(つまり故障が他に波及しない)場合

(イ) 数個の部品が不良

いくつかの部品が不良であっても、それらが機能上ペアで働いていたりある種の連鎖的な回路を構成している小ブロックであるとする、それは1つの故障原因であると認識される。それらの部品は必然的にしばしば回路図上でも物理的な配置上でも隣接した位置関係になっていることが多い。

この故障の現われ方はさらに『並列的』、『連鎖的』の2つに大別される。部品不良を

もたらした要因については、後で詳しく述べるとして、今何らかの要因が発生したとする。それがほとんど同時に複数の部品を不良にさせたとするそれは『並列的』であるし、ほとんど同時であっても、きちんとした因果を説明できる場合は『連鎖的』であるとする。

・並列的 何らかの要因 \rightarrow 部品Aの不良
 \perp \rightarrow 部品Bの不良

・連鎖的 何らかの要因 \rightarrow 部品Aの不良 \rightarrow 部品Bの不良

この2つのパターンは明確に区別しえない場合もあるし、区別する必要のない場合もあるが、『修理用知識』の獲得の時に知っておいた方がよいと思う意識の使い方である。

また『並列的』・『連鎖的』の組み合わせも考えられ、例えば次のような

・何らかの要因 \rightarrow 部品Aの不良
 \perp \rightarrow 部品Bの不良 \rightarrow 部品Cの不良 \rightarrow 部品Dの不良

故障連鎖は結果として4つの部品不良があるが、熟練技術者は『故障の原因は1箇所』とみなすのである。

(2) 熟練技術者が数箇所の不良とみなす場合

(1)の性質を持つ故障が全く因果関係のない2箇所で発生した場合、すなわち独立して(1)が起きた場合である。頻度としては少ないが、現実起こるので注意する必要がある。

これらの不良に対する対処方法はどうなっているだろうか。(1)の場合、たとえ数箇所の部品不良であっても互いに何らかのつながりがあるので、故障の連鎖をたどることができやすいのに対して(2)の場合はほとんど脈絡がないため、1つの不良を発見し取り除いたあと、次の不良を見つけるのに苦労する場合が多い。

3.3 故障の原因をつくりだす要因

故障の原因は『部品不良、導通不良、調整不良、取付け不良』などであることを述べたが、これは何により生じるのだろうか。もちろん 3.2.2 で述べたようにいくつかの部品が連鎖的に壊れるということもあるが、やはり最初に部品が不良になる要因が何かあるはずである。

3.3.1 部品不良の要因

部品不良により回路の状態が変わり、他の部品を不良にしてしまうということはよくあるわけだが、それでは最初の(primitiveな)要因は何だろうか。

(1) 部品が自身で不良となる

機械部品の場合、磨耗やいろいろなストレスの積み重ねによる劣化が避けられないので原理的にある種の寿命を持っていると考えられる。

電子部品の場合、半導体ならば絶対定格を越えない限り半永久的な寿命を持つと考えられているが、現実には突如壊れたと思われる形跡に何度か出くわした。LSIにおいて、中の一機能が不良になった場合、外からその原因を解析するのは一般に難しい。半導体素子が突然壊れる原因は他の研究に譲るとしてもその現象だけは認識しておかなければならない。

一方、受動部品の中で例えば電解コンデンサのように原理的に劣化が避けられないものもある。劣化させる原因として温度・湿度・振動といった外的なもの、過電圧・リップル・充放電といった電気的なものがある。外的なものの中で特に温度変化が寿命に影響する。電気的な要因では、電源投入・切断時の過度的な現象やスパイク(電圧・電流)といったストレスの積み重ねが考えられる。^[4]

これらの劣化要因は回路設計時によく考慮されていないと思われるものもあるので、修理技術者はそれを含んでおく必要がある。例えば電解コンデンサは電源回路など高温にさらされやすい所に使用されるにもかかわらず、半導体ほど温度(放熱問題)に対して特別な配慮がなされないことが多い。さらに、内部抵抗が大きい(損失が多い)のでリップル電流により発熱するため、劣化に拍車をかける。

また部品の定格と回路の動作範囲にマージンがあまりない場合、部品の製造上のバラツキが最悪の方向にあると、電源投入・切断時あるいは動作中、常に劣化要因またはそれに近い状態にさらされることになる。

このように修理技術者からみると「しかるべき要因による部品不良」と思われるものもあるが、設計時には見落とされがちであったり、またコストによる問題から特に民生機器では多少信頼性を犠牲にしてもマージンをあまりとらないことが多いのであろう。これらの情報は『設計』にもフィードバックされるべきであるが、当面は修理技術者が注意せねばならないであろう。

(2) 操作ミス・ユーザの不注意による部品破損

これはくだらない部類の部品不良であるが、「ユーザが修理しようとして不用意にショートさせて部品を壊した」というような場合である。

3.3.2 導通不良の要因

パターン切れは基板に物理的な力が加わって（例えばねじれとか）割れた時に起こる。基板にひびが入った程度の場合でもパターン切れを起こしていることがある。組立時の基板取付け中に無理にこじたりしたような経歴があって、基板にひびが入って弱くなっていたとすると、しばらく使っているとパターンが切れて突如故障というようなこともある。

ある故障によって瞬間的に大電流が流れパターンを焼き切ってしまうこともある。特にパワーアンプの場合、事故が発生するとその電流のオーダは数10Aになることもあり、細いパターンは消失する。

配線不良は主に修理の時余計な力が加わって生じることが多い。コネクタ類の接触不良はコネクタの接触部分の金具のバネ圧力が弱い時に起きる。

半田付け不良は組立行程（半田槽での半田付け）で半田が十分のっていないとき起こる。製品検査の時はたまたま接触していても移動中の振動などで部品のリード線が回路から浮いてしまうのである。LSIのフラットパッケージの場合、見た目ではきちんと半田付けされているかわからず、上から樹脂でコーティングされていたりしていると導通検査もやりずらく技術者なかせである。

接触不良において共通した症状として「時々故障する」というのがある。

3.4 部品不良の例

部品不良の現われ方は大体部品に固有である。抵抗は焼けることはあってもショートはないし、コイルは断線の他はあまりない。修理メモから故障原因となった回路の部品不良の主なものを整理すると、次のようになる。

抵抗 (カーボン、不燃)	焼損、オープン
半固定抵抗	接触不良
電解コンデンサ	ショート、オープン、リーク (容量減、抵抗増)
チップコンデンサ	オープン
コイル、インダクタ	断線
トランス	断線
スイッチ	接触不良
ダイオード (スイッチング、ツェナー、整流用)	ショート、オープン
トランジスタ	C-E間ショート、 B-E間オープン

3.5 症状の現われ方

故障の原因の程度や数、その要因というものの性質がわかったが、故障原因によってその症状はどのように現われるだろうか。

3.5.1 症状のとらえ方

『症状』という言葉は知識工学的には特に『事実』、『原因』、『状態』などと区別されなければならないので、詳しくは4章で述べるが、ここでは日本語で普通にいう「症状=病気や傷の様子」にあてはめ「故障の(表に)現われている様子」を考えることにする。

症状はあまり多くの種類がない。かぜの症状として「くしゃみ、はなづまり、寒け」などいろいろ考えられるが、それに対する原因の数はずっと多い。一般に

原因の数 > 症状の数

であるといえる。

症状について少し深く考え始めるといろいろな症状パターンがあるように一瞬思えるが、それは内部のことであることが多い。例えば「音がでない」というのは症状であるが、「パワーアンプ部が動作していない」というのはもはや外に現われている様子ではなく、内部の状態である。『くしゃみ』という症状と『39度の熱』という事実からそれは『かぜ』であると予想できるかもしれないが、『かぜ』という具体的な症状が表にあるわけではない。そこには『かぜ』にはどういう症状があるかという知識を働かせた跡がある。

同様にとらえかたをオーディオ機器にもあてはめ、あくまで外に現われる様子を『症状』とする。

3.5.2 機種ごとの症状

機種にかかわらず出る症状

音が出ない、電源が入らない

(1) アンプ

煙が出る、プロテクタが動作している、音が歪む

(2) CDプレーヤ

音が飛ぶ、読み込まない、音が歪む

(3) カセットデッキ

音がふるえる、高音が出ない、録音できない、デッキのメカが動作しない

(4) チューナ

感度が悪い、自動選局しない

(5) カーオーディオ

(3)、(4)に準じる

3.6 作業の進め方

対象の特徴、原因と症状の性質などがわかってきたが、さていよいよ機器を前にしてどのようにして修理の作業を進めていくかという話になる。

3.6.1 症状と原因

症状と原因は互いに故障診断において末端の位置にある（詳しくは4章で述べる）。原因がどこかにある。それは何らかの部品不良を招くかもしれない。それは、いろいろな事実「～が何Vである。」とか、状態「～ブロックが故障している」などをもたらし、そして最終的に「音が出ない」というような症状として表に現われる。ユーザは『症状』から「どこかが故障しているな」と思い修理に出すのである。修理技術者は『症状』をもとにいろいろな状態・事実を調べ、知識を使って故障原因をさぐる。原因が見つかってもすぐさま原因を取り除いて正常な状態にできるとは限らない——例えばフラットパッケージのLSIの交換は手間どる——が「原因を見つける」ということが修理において中心となる行いであることはいままでの間もない。実際、修理において一番多く費やされる時間がこの「原因をつきとめる」ための時間であり、修理技術者はみつかった瞬間修理が終わったと感じるものである。

3.6.2 訴える症状

私が経験した作業場はあるメーカーの工場であり、メーカーが『不良交換品』として倉庫に保管してあったものをまとめて修理することが多かった。そのためユーザから直接詳しい症状を聞きだすことは皆無に等しくたいは『伝票』に書かれている短いコメント、「音がでない」「不動作」といった簡単なものしか得られなかった。書かれている症状と実際の症状が違うこともたびたびあった。

こうなってくると、「訴える症状」というのはあまりあてにできなくなってくる。人間を診断する場合と違って機械は口がきけないので、すべての状態をこちらから調べなければならぬ。「電源が入らない」というのを除けば、たいはの症状の確認は『通電』して行わねばならない。

ところが、ここにめんどうさがあって何でもかんでも電源を投じてよいとは限らないのである。例えば「パワーアンプで発煙の跡があった場合（これは症状である）、パワー段のトランジスタが破壊していることが多い」ということを経験的に知っている。このような比較的大事故に入る部類の故障はその波及が一瞬に進むのでユーザが異常に気づいて電源を切ったときはすでに波及が行きつくところまで行ってしまっていることが多い。けれども波及が完了しないうちにユーザが電源を切ってくれていた可能性も考えられる。この場合、修理技術者が不用意に電源を投じて波及を完了させることは、それが状態確認のためであっても避けるべきである。

こういった場合「煙が出た跡がある」のような唯一の症状（電源が入れられないのでそれ以上の症状は調べられない）だけで修理を進めることになる。症状は1つだけのなので、手がかりをつかむために状態を調べたり、測定を始めることになる。しかし、そのままでは電源を入れられないから、どこかを回路から分離させるなどの処置をして電源を入れることになる。これは普通の意味での『電源を入れる』ではなく『計測』のある部分になっている。（計測については4章で詳しく述べる。）計測の『前処理』であり、時には『作業場の基礎知識』も使って状態を調べている。『発煙の跡』という症状をみて、熟練技術者はすぐには電源を入れず各ブロックの状態を調べに行く。最終段のトランジスタのコレクタをVccから浮かし、あるいはメインアンプブロックを切り離して初めて電源を入れる。

3.6.3 状態把握

症状というのは種類も少なく単純でまたユーザの訴える症状はあまりあてにならないことが多いと述べた。ではその不十分な情報から修理技術者は何をし始めるかということ「状態を調べに行く」とも述べた。「状態を調べに行く」ことすなわち『状態把握』についてもう少し考えてみる。

3.6.2で少し述べたように症状をみてとった後、修理技術者は症状に関する少ない情報を参考にしながら、電源を入れるか入れないかの判断をし、少しずつ内部の状態を調べ始める。回路を切り離したり信号を入れたり、時には具体的な数値（電圧や電流、周波数など）を測ったり波形を見たりするが、この段階である1つの部品、または極く小さな構成の回路だけに着目して詳しく調べに行くことはない。あくまで全体的な動作状態をつかむ目的で計測をしている。もちろん、何度か同じ『症状⇒原因』に出会っているために状態把握をとぼし、直ちに原因になっていると（経験的に）思われる部品を（不良であるか）調べ、不良ならば交換して修理完了というようなこともあるが、その場合でもその予想が外れたとき、すなわちその部品が正常だった時、すぐにこの状態把握のレベルにまで戻れるだけの意識の準備がある。

この時点で意味不明だったユーザから訴えられた症状に納得したり、ユーザの操作ミス・勘違いによる「症状のとりちがえ」などがわかることも多い。

新しい機種・モデルにとりかかる時は特にこの部分に時間をさく。アンプにはアンプ、CDプレーヤにはCDプレーヤの典型的なブロック構成があって、それは機種によらず、だいたい一定であるが、ダブルカセットデッキのように基本的な性能（録音・再生）よりも2台のデッキの制御の方が複雑な場合、機種ごとに異なっていることも多く、その動作理解にはしばらく時間がかかる。

状態把握には機器の種類によってある決まったパターンがある。訴える症状が何であるかにかかわらず、だいたい「これとこれとこれは調べる」という決まりきった状態調べの項目がある。概しておおまかな状態把握であるが故障原因をつきとめるために故障ブロッ

クを限定するための大きな情報となる。アンプでは電源が入れられる場合、症状を見た後、必ずLine入力とスピーカ間、Power入力とスピーカ間の入出力特性を発振器とオシロを使って調べる。〔付録3〕

3.6.4 故障診断

症状を理解しある程度決まった状態チェック項目をチェックした。故障診断は症状から始まり故障原因の確定に終わるわけであるから、このわくにおいては3.6.2の段階でもう診断に入っている。熟練技術者が持てる力をフルに発揮しだすのはこのあたりからである。症状と状態チェック項目から修理技術者はだいたいどのブロックが異常かに予想をつける。電源が入れられなかった場合も適当な処置により、正常であろうと思われるブロックには通電できるようになる。どこが異常ブロックなのかの予想はかならずしも当たらない。それはその判断が経験的知識によるものに基づいていることが多いことと、確定させるだけの十分な情報を持ちえないことによる。それにもかかわらず、症状・状態パターンと故障ブロック決定には密接な関係があり^[4]、それをこの経験的知識が有力に示している。

故障ブロックが仮定でき、つまり「どこかがおかしいが、それはこのブロック内にありそうだ。」ということがわかると、さらにそのブロックに注目して調べる。方向としてはこのブロックの規模を小さくしていき、最終的に分割できないブロックに行きついたとき、3.3で述べたような故障原因、部品不良・導通不良が確定する。ただし『調整不良』に関してはこの方法ではわかりづらく、ある程度まとまったブロックの動作パターン（動作状態とその組み合わせ）などから不良な調整箇所を推定することが多い。どちらかというブロックをせぼめてきて、部品不良を疑ったが失敗したという時、調整の不良があると思う場合が多いようである。

故障ブロックを予想したが、そのどこにも故障原因が見いだされなかった時、新たな故障ブロックを仮定することになり、また上に述べたことと同様のことをする。故障ブロックの範囲をせぼめていくいくためにいろいろな計測をし、その結果により次の計測を考えることを繰り返していく。

『計測』についての詳しい説明は4章で行うが、いくつかの例をあげておく。

- ・電圧、電流測定
- ・波形観測
- ・目視検査
- ・音を聞く

3. 6. 5 意識外の計測・動機付けのない計測

いかに熟練した技術者であっても無意識にやっていることはたくさんある。『計測』という言葉の意味を本研究では広い意味で使うことにしているが、例えば『見る』というその1つの行為は目をあけている限りたえずやっていることになる。ことさらに「今この部品を見ているんだ」という意識を持つことなしに部品を見ていることが多い。パイロットランプや各種ディスプレイにしても最初は注意して正常かどうか意識して見ているが、ある程度修理が進んでくると、正常であることの確認や状態把握のために無意識に見ているにもかかわらず、故障診断のための情報（計測結果）だとは熟練技術者の意識には思えなくなるのである。そうすると「見ていないのも同然」というふうに解釈できるかというところではない。これは修理中に各種の測定あるいは測定の処置をしている時ランプ状態・ディスプレイ状態、部品であれば発熱、音ならば異音などを通じて、通常でない状態を（それは異常であるかもしれないし、新しい状態の情報かもしれない）修理技術者に知らせることができる。これはコンピュータでの割り込み処理に似ている。普段は何もしないが状態変化（無意識に電圧を測定した場合は事実の変化）が技術者に対して、現在の処理を中断してその変化の解釈をすることを要求するのである。

意識外の動作と似たものとして「動機付けのない計測」がある。「オシロスコープのプローブを持っていたのでついせに近くの波形も見ておいた」という場合がその例である。作業能率の上から 3.1.2 で述べたように機器のとりまわしのわずらわしさから少しでも解放されるため、「ある点について計測したらその近くをついでにやっておくのは悪くない」ということを熟練技術者は知っている。その測定値、結果は当面使わないということがわかっているにもかかわらず、後に故障ブロックをせばめていく過程で、またはそれが失敗して新たな故障ブロックを仮定するとき有力な事実となるかもしれないからである。そしてそれが異常な値・結果であれば、前述の『割り込み』が発生する。正常または不明の場合は頭の中のどこかに一応ストアされる。『一応』というのが人間らしいところで「使うか使わないかわからないけど一応調べておこう」と思う時の『一応』なのである。

3.6.6 並列的不良、連鎖的不良がある場合の故障診断

3.2.2 で述べたようにたとえ複数の部品不良があっても熟練技術者は「1箇所の不良」とみなすことがある。こういう時どのように修理が進められるのだろうか。

まず並列的な場合、部品は近くに位置していて回路図上でも近いことが多い。故障ブロックをせぼめていく過程のある所で「複数の部品が同時に破壊されたかもしれない」と意識するところがあり、そこでいくつかの部品不良を疑い、調べることになる。「ここはペアで壊れる可能性がある」ということを知識として持っている。

一方、連鎖的な場合、症状の現われ方がいろいろ考えられる。例えば「部品A不良→部品B不良」という2段の場合でもその症状CはA単独によるものなのか、B単独によるのかあるいはA・B両方によるものなのかわからない。しかし普通、症状Cは部品Bによることが多いため、まずBの不良を確認して次に「Bが不良の時はAも不良である可能性がある」という知識を働かせAの不良を確認する。

3.6.7 作業上のテクニック

計測を効率よく行うためにという意味でこれは『作業場の基礎知識』に属することであるが、実際の修理ではいろいろなテクニックが用いられる。

治具や測定器具の工夫は大幅に能率を改善する。しかし機種モデルに固有の装置・方法であったりして、普遍的な知識としては書けないものがいっぱいあった。またある故障ブロックが確定するとそこから先は診断せずに「基板ごと交換」ということもしばしばあった。これは基板のコストと部品レベルまで故障原因をさぐるための技術者の人件費をくらべて基板の方が安ければ、経営工学的にそっくり交換したほうがよいからである。この件に関しては他に譲る。

正常に動作することが保証されている基板（ブロック）というのは測定器具としても重宝する。症状といくつかの状態把握だけでは故障ブロックがはっきりしない時特に威力を発揮した。正常な基板とさしかえて正常な動作をすれば故障はそのさしかえた基板のどこかにあるはずだし、同じ症状であったら故障原因は他のブロックにあると考えられる。脈絡のない2箇所の故障の場合もこの方法を応用することによって比較的簡単に見つけることができた。

これらの手段は本研究では主旨になじまないとして一応ないものとして考えるが、知識獲得という面から見た場合、本研究の方が多くをカバーしているので、これらの埋め込みは容易であると考えられる。

3.7 まとめ

修理用知識を獲得するためには、実際の修理をよく調べ、対象の特徴をよくつかみ、何が問題で、そしてその問題をどのように解決していくのかを解明しなければならない。

修理の対象は、アンプ、チューナ、CDプレーヤなどのオーディオ機器であり。回路的にはデジタル回路とアナログ回路が組み合わさったものが多く、CDプレーヤのように機械的機構（メカ）を持つものもある。大規模集積回路（LSI）や小型化された部品のおかげで、小型計量化・高密度化にある機種では細かな作業能力を必要とし、いぜん重さがステータスとされる高級メインアンプでは、その重さが取り扱いを困難にしている。これらの特徴は、必要な計測とやりやすい計測を必ずしも一致させない。現実の修理では、やりやすい計測をすることが多く、計測の順序や後で述べる『動機付けのない計測』に特に影響を与えている。

故障の原因は電氣的・物理的な要因による部品の劣化および不良が多いが、導通不良、調整不良というものもある。原因の数は1つのこともあるし、複数のこともある。後者には何かの要因が同時に複数の故障原因を作り出す『並列的』な場合、複数の故障原因が因果をもって発生する『連鎖的』な場合、それぞれが何の脈絡もない場合、そしてそれらの組み合わせがある。熟練技術者は脈絡のない複数の故障原因がある場合を除いて、複数原因であっても、それぞれの原因を結び付ける知識によって「1つの故障」とみなすことが多い。

『症状』とは「表に現われる様子」であり、修理とは『症状』を見ることから始まり、最終的に原因を見つけて直すことである。原因を取り除くことは、例えば不良な部品を交換することであるが、修理においてもっとも重要なことは、「原因を見つけること」、すなわち『故障診断』である。症状は一般に種類が少なく、それだけでは修理の作業を始めるのに十分な情報でないため、いろいろな状態を調べに行くことが多い。これは機種ごとに調べる項目が決まっていて、それを『状態把握のパターン』と呼ぶことにした。

故障診断の中ではいろいろな計測が行われるが、それらのすべてがはっきりと意識されているとは限らない。また、測定点が近かったため「ついでに」やっておいたという種の計測もある。こうした『意識外の計測』、『動機付けのない計測』というのは一見ムダのようにみえるが、故障原因をつきとめるために、あるいは作業能率を上げるために大きく貢献していることが多い。

4 修理用知識の獲得

どのような手順で修理を進めていくかはだいたいわかったが、そこで「なぜその測定をするのか」とか「どうしてその部品が悪いと思うのか」というようなことは明らかになっていない。症状を見てから故障の箇所を見つけるまで、熟練技術者はありとあらゆる知識を使う。上の2つの質問の答えはいずれもその知識の一部になる。知識の種類、性質は多岐にわたり量も膨大であるから、知識と思われるものをむやみに書き出してもほとんど使いものにならない。熟練技術者から体系的な形で知識を引き出し、獲得することは知識の訂正・追加、またわかりやすさという面から非常に重要である。

問題の領域をよく吟味し適当な範囲を設定することも重要である。簡単な問題解決の集合とその組み合わせだけではある程度以上の複雑さを持った問題解決ができないし、適切に表現もできない。また最初から領域を広げすぎて手がつけられないようになったり、抽象論・一般論に終わらないようにも注意しなければならない。

さらに2章で述べた『支援システム』の中における故障診断のための知識というものを意識しているので、「論理的に美しい」構成も重要であるがむしろ実践にそくした形で知識獲得したいという狙いがある。そのために必要な修理用知識のとらえかたを4.1で、熟練技術者と知識工学者とのインタフェースについての問題、例えば知識エディタについては本研究ではカードによる試みと問題点について4.2で論ずる。さらに知識獲得の過程で大きなヒントとなった「大局的な視点・局所的な視点」について4.3で論ずる。

4.1 修理の時使われる知識

3章では修理全般を見渡し、実際の修理がどう進むのかの概略をつかんだが、診断における知識やその働かせ方についてはあまり触れなかった。ここでは特に診断のために必要となる知識のとらえかたについて論ずる。

4.1.1 領域の設定

どの程度の範囲の知識を取り扱うかは非常に重要である。現実の問題はとても複雑で、ありとあらゆる知識—専門知識（経験的知識と教科書的知識）から常識の類まで—を使っている。3.6.5で意識外の動作について述べたが、それと同様に意識外の知識の働かせ方というのものもある。常識による判断というのは、ほとんど常に行われていてそれをことさらに意識することはないし、（少なくとも修理中に「常識を使って判断するとこうなる」とは意識していない。）またそれを知識だと思ふ人もいない。

知識工学分野の歴史では膨大な常識を持つシステムがその組み合わせによって幅広い問題解決ができると考えられた時期があったが、結局それは使いものにならないことがわかっている。また簡単な問題解決ができるモジュール（例えば単純なプロダクションシステム）がいくつか集まっても複雑な問題解決はできない、あるいは非能率的であるという

こともわかっている。何に対してどのような領域で問題を設定し、その解決のためにどのような知識を必要とし、どのように使うのかを明確にした上でシステムを構築していくことが重要である。ある程度限られた領域での専門知識を扱うことによってより実用的で実践的かつ適度な複雑さを持つようになる。

本研究では2章で述べたように、電子回路技術についてノ教科書的な知識は持っているが修理に携わったことがないという技術者、『理論技術者』用の『支援システム』というのを意識し、その中で使われるべき知識に注目してきた。すでに『作業場の基礎知識』というものを考え、それを故障診断のための知識とは分けて考えることにした。『作業場の基礎知識』は前述の一般常識と似た性格を持つ面もあり、その分野における常識と考えることもできる。

『作業場の基礎知識』を別にして考えた今、検査や測定は自由にできると考えてよい。次に対象のどの部分を修理するかであるが、電気・電子回路の部分に限ることにする。CDプレーヤやカセットデッキはメカを持つので故障の原因はメカ部にもありえるわけであるが、

1. 計測の方法や対象の特性がメカ部と電気・電子回路部では異なるので、知識を記述する時互いになじみにくい。
 2. メカを持たない機種もあるのに対して電気・電子回路はどの機種も必ず持つ。
 3. 私自身が機械部品についての理論的な素養を持たない。
- などの理由により本研究では、電気・電子回路部だけに故障があると限定して考える。

4. 1. 2 事実、状態、症状

すでに事実、状態、状態という言葉を使ってきたが、ここで本研究での意味を明確にしておく。

1. 事実

客観的に得られるもので一般に外には現われないものを指す。例えば『電圧』というのは普通外には現われないが、測定器具をもって測定すれば測定点に固有な値が誰がやっても同様に得られる。

『事実』としてとらえるものは、電圧、電流、抵抗値、電圧波形、導通などが主なものである。また『部品不良』も事実としてとらえる。トランジスタチェッカなどというものは普通の作業場にはあまりないが、そういったもので検査すれば客観的に良否がわかるという意味で『事実』なのである。(普通は『作業場の基礎知識』を使ってテストなどによってトランジスタの良否を調べることが多い。)

この考え方で修理不能な部品から少し範囲を広げて回路ブロック(基板)の不良も含めることにすると、知識の記述をトップダウンで行いやすくなるし、いろいろな保守の方法にも対応することができる。3.6.7で述べた『基板差し替え』による修理は時間的・経済的制約のある場面ではよく見られる。実際には基板の個々の部品について調べることができるのだがその時点ではできないので、いわば『巨大なLSI』とみなしてそれを交換しているのだとみなせる。3.6.7で述べたようにこの方法による修理は現実に頻繁にとられるのであるが、本研究では考えないことにする。したがって基板不良は『事実』ではなく次に述べる『状態』になる。

2. 状態

『状態』とは事実の組み合わせから生じている様子であり、一般に表に現われない。状態には名前がついていることもあるし、ついていないこともある。そのため事実とか症状と混同されやすいのであるが、あくまで事実からあるまとまった様子を示すものを『状態』ということにする。例えば、今何かの部品不良があってそれが発振すべきでないところを発振させているとすれば、それは『発振』という状態であるといえる。確かにオシロスコープで発振している波形が観測できるがそれは『事実』である。

「あるモジュールまたはブロックの動作がおかしい」という時、それは『状態』である。先に述べた『基板の不良』というのはブロックの不良であることが多いが、回路の規模が小さくても『状態』は存在する。例えば部品1個についての計測からの1つの『事実』が得られた。その『事実』からだけでもそれを的確に説明する『状態』が存在することがある。例「定電圧回路においてツェナーダイオード両端の電圧が0Vに近い」という『事実』は「定電圧源が機能していない」という『状態』である。このような対応の場合『事実』から『状態』への置き換えが意識されなかったり、状態の適切な表現が見つからなかったりするので知識獲得の際、困難をきたすことがたびたびあった。『事実』と『状態』が意

誤の中であまりに近いため熟練技術者の口からなかなか『状態』が出てこないのである。

『状態』は何も不良ばかりを指すのではなく、「正常に動作している」という『状態』を積極的に利用することもある。特に診断の初期では故障していないブロックを確かめることが重要である。

3. 症状

症状とは表に現われる様子である。この定義では「計測によって得られた結果のいくつかについてそれを事実によりわけるか症状によりわけかが不明である」という問題点が生じる。本研究では『症状』を特に診断の初期に使うことをふまえて、そのために有力な計測結果というのを『症状』ということにする。症状の数は少ないのが特徴である。

計測の中でも人間の視覚、聴覚といったものを使って得られるものが症状で、『パイロットランプがつかない』、『音が出ない』、『音が飛ぶ』などといったものがその例である。

4. 1. 3 原因と中間原因

原因についてはその要因と共に 3.2 3.3 で述べた。ここでは電気・電子回路に限って『原因』を定義しておく。「原因は『部品不良』、『導通不良』、『調整不良』のどれかであり、それらの要因は何であれ何らかの事実を生じる」となる。

修理を進めていくと感覚的に「これが原因だ」と思うことがある。通常の日本語の使い方では間違いではないのだが、これが先に示した原因のどれにもあてはまらないことがある。例えば「『局部発振器の発振不良』が原因だ」と思うときである。これは実は状態である。これを状態の中でも特に『中間原因』ということにする。これはわかりやすい例であるが、熟練技術者が対話の中で示す原因らしきことは実は状態であることが多い。そのためそれが原因か状態かはっきり確かめて知識の収集をしなければならない。

比較的小さなモジュールが不良であるという中間原因が得られたとする。その先の修理の進め方は形式化していて原因も2~3の可能性しかなく、作業方法も熟練技術者にとって簡単で時間もかからないとする。そういうとき熟練技術者の意識の中で「なおった」と思っているためそれが本研究でいう『原因』でないにもかかわらず、『原因』だとして片づけてしまうことがある。そして修理メモにもそのことしか記録されないのである。これはメモを調べるときに注意しておかなければならないことである。

4. 1. 4 計測の定義

本研究では『計測』という言葉を広い意味で使うことにしているが、ここで定義を与えておく。

「計測とは人間の五感（視覚、聴覚、嗅覚、味覚、触覚）および測定器具を使って情報を得る行為をすべてを指す。」（ただし、現実には味覚はほとんど使わないから今の場合は、はずしておいてもよい。）

・測定器

作業場に測定器はデジタル電圧計、電子電圧計（真空管電圧計）、オシロスコープ、低周波発振器、周波数カウンタ、ワウフラッタ計、変調器、テスタ、定電圧電源器、疑似負荷などがある。測定器の中には治具・専用検査器も含めることにする。治具は必要に応じて作ることもなるし、指定の物が与えられることもある。『トランジスタチェッカ』『LED検査器』『平衡・不平衡変換器（トランス型）』『スピーカ端子直流もれ検査用電圧計』などを作って作業効率をあげた。

・計測で得られる情報

計測においては『作業場の基礎知識』を十分駆使して能率よくかつ正確に行えるようになっていけるとする。計測によって得られる具体的な情報は例えば次のようなものがある。

視覚では目に見えるものすべて、聴覚では音が出る、出ないだけでなく音質まで、嗅覚ではにおい、とくにこげたにおいが修理では役に立つ。触覚では主に部品の温度を定性的に調べるために使う。一方測定器具を使った計測ではその情報は部品などの良否、波形の形状、具体的な次元を持った値〔V, A, Ω, W〕などが得られる。

4. 1. 5 症状・状態パターンと原因

熟練技術者は症状と状態のいくつかの組み合わせパターンから、故障原因のあるブロックを予見することができる。

修理の様子を見ているとまず症状を確認したあといろいろな計測により状態をつかんでいることがわかる。症状だけでは情報が少ないためよほど特異なものでない限り、原因はおろか中間原因を設定することさえ困難なことがほとんどである。そのため状態を調べに行くのであるがその状態のとらえ方の範囲が徐々に狭まっていくため、どこまでを『症状パターンと原因』の関連づけという経験的知識にたよっているかが明確でない。しかし、積極的にその臨界点を設定することによって知識の記述がしやすくなることがわかった。それが 3.6.3 で述べた『状態把握』である。臨界点の設定については 4.3 で詳しく述べる。

4. 1. 6 チェックリストとモデル形成

機種ごとに状態把握のパターンがある。これは「症状によらず必ず一通り調べる」というものであるから。自動車整備で用いられる整備手帳のチェックリストのようなものと考えられる。一方故障ブロックを仮定する時には、この状態把握で得られた状態だけでなく症状も用いられるため状態も含めて1つのリストにして、機種ごとの鑄型にしておくといよい。症状と状態をあわせて10個程度の項目により故障ブロックを限定していくことが多い。これは現実の熟練技術者の動作、意識の働かせ方と似ていて、知識も引き出しやすく書きやすかった。そこで症状と状態把握のパターンをまとめたものをチェックリストと呼ぶことにする。

次に計測結果をどのように扱うかを考える。3.6.5で述べたように現実には意識外の計測や動機付けのない計測がある。熟練技術者はこれらの計測結果のすべてを記憶しているわけではないし、また利用しているわけでもない。したがって知識獲得の時にこれらをどのように扱うかが問題である。

まず、計測結果は意識外であっても動機付けがなくても修理中に（あとで）使われる可能性があるので、どこかに蓄えられるべきである。いわゆる『割り込み』のかかる事実あるいは状態というのは動的なものであって、これを仮に支援システムを想定した場合に埋め込むのはそれに応じて動的に対処できるだけの速度や環境を必要とするので、ここではなじまないものとして省く。

その他の場合の『意識外の計測』、『動機付けのない計測』であるが、これらが行われたかどうかを認めるのは難しく、しばらくして何かの判断がくだされたとき、その根拠の不足から「あーそういえばこういう結果も使っているな」という程度にしか認識されない。したがって判断のための知識を記述するとき、その計測の履歴をさかのぼってたどり確認することになる。その時、測定点が近いとかやりやすいといった理由で、同時にあるいは無意識に行われたのかどうかを記録しておくことは知識を整理する時、非常に役に立つ。またこの機構をもって故障診断システムを作った時、支援の性能を作業能率を加味したシステムとして高められる可能性がある。

計測結果は使う使わないにかかわらず熟練技術者の中に現実の機種モデルと同じような形でモデル形成されている。したがって知識として記述する時に、このモデルを機種モデルごとに持ち、常に参照できるような機構が必要である。

4. 2 計測カードによる計測整理

修理の経験により実際の修理がどのように行われてきたかをまとめてきたが、現実の知識獲得には専門家の協力と適切なインタフェースが不可欠である。

本研究では、私の修理経験のメモを利用し、そこに専門家がいると仮定し、知識の整理をすることにした。修理メモは不完全なものであり、原因や修理の方法がきちんと記録されていることもあるし、原因だけ、部品名(番号)だけしか記録されていないこともある。

【付録4】比較的きちんと記録されているものでもそれはほとんど本人にしかわからない。何時間もかけて修理した修理の様子をわずか数行のコメントに押し込めてしまうのであるから当然ともいえる。しかしコメントは不完全ではあるが熟練技術者にとっては必要かつ十分な情報である。技術者がすでに知っていることや状態把握の部分はなにも書かないのだが、いわゆるエッセンスの部分を書き出してあるので、熟練技術者はそれを見ると故障修理の全体を思い出すことができる。『作業場の基礎知識』に属するものはあまり記録されていないが、とくに有効な測定方法やその際の注意点などが記されていることもあった。そのメモの中ではもちろん『作業場の基礎知識』とか『修理用知識』という区別の意識はなく、『原因』、『事実』、『状態』、『症状』といった区別もはっきりなされていない。また修理の流れを司る知識も書かれていない。

ここではこのような曖昧で限られた情報から、すなわち「専門知識を伝えたくても自分からはできない」技術者から、3章および4.1で論じてきた枠組みに入れるための第一段階、インタフェースとしての『計測カード』について述べる。

4. 2. 1 計測カードの導入

研究の初期の段階ではメモから何をどのように書き出してよいかまったくわからず、自分の書いたメモを見直すということから、客観的なとらえ方がしにくかった。最終的には何らかのルールの集合になるだろうとは予想できたが、既存の推論システムのわくでとらえようとする試みはなかなかなじみにくかった。「どうもちがう」、「不自然だ」と思うことがたびたびあった。その不自然さが『意識外の計測』、『動機付けのない計測』や『事実』と『状態』のとりちがえから生じているものだということが後にわかったが、手をつける段階ではなかなか気づかない。いろんな主観が邪魔をしてしまう。

そこで「行動については客観的に記録できるはずである」と考えた。そして『行動』とは何かを考えて広い意味で『計測』という言葉を使い『計測』についてカードを用いて整理することにした。

4. 2. 2 PDSサイクルと計測

人間の行動をP(Plan:計画)、D(Do:実行)、S(See:評価)の3つのフェーズに分ける考え方があるが、ここでこのサイクルと計測との係わり合いを考える。

何か行動があればそれは普通目にとまる。例えば修理の風景全般をビデオに撮影したとする。熟練技術者は終始無言のまま修理を完了したとする。そこに写るものは「何をしたか」つまり先のサイクルでいう『D』の部分だけである。これは本研究でいうところの『計測』に置き換えられる。(視覚や嗅覚による計測をしているかどうかは、例えばビデオ映像からは得るのが困難ではある。)計測によって得られた情報から何を評価し〔See〕、次に何を計画〔Plan〕しているかを読み取るのはほとんど不可能である。

そこで熟練技術者とのインタビューによっておのおのの行動〔計測〕のS部とP部を得ることになるのであるが、本研究ではこれが不可能なため修理メモより修理風景を回想し(先のビデオを再生するのに似ている)、計測を取り出すことから始めた。

計測カードではDSPの順でサイクルを整理することになる。〔付録5〕

4. 2. 3 カードおこしの手順

1. 計測項目の書き出し

計測の項目を書き出していくとその数はとてつもなく多くなる。修理の複雑さにもよるが、2~3時間の修理でだいたい100項目以上である。それらすべてが修理のために行われている計測ではない。ほとんどが「ついでに」やっておいた計測、いわゆる『動機付けのない計測』や、やったかどうかを忘れてしまう『意識外の計測』である。それらを逐一カードに記録していたのではちががかないので、1つの基準「結果が計画に何らかの形で影響を与える」以外の計測はすべて省く。それらは意味のない計測であったと結論づけてよいからである。

ここでいう「何らかの影響」とは必ずしもそのDSPサイクルだけにおけるものでなくてもよい。それは例えば4.1.6でいう『モデル形成』のための計測であるかもしれないからである。その情報は後に使われる、すなわち何かの計画をする時にそれに必要な計測項目をすべて行うのではなく、モデルの中に存在していたらそれを使えばよいのである。

ここで関連する『計測』項目の抜き取りで注意しておくべきことがある。それがほんとうに『計測』であるかどうかである。「発振しているかどうかを計測する。」これは日本語としては正しいが、本研究ではそれは「状態を調べている」ことになる。「オシロスコープで波形を観測し~の波形が得られた」という計測からそれを理解し「~の波形が~である場合、~ブロックは発振している」という知識を使って「発振している」という状態を得るのである。

2. 計測カードのフォーマット〔付録6〕

熟練技術者の思考の流れをできるだけ無理なく表現できるよう、かつ第3者にもわかりやすいように心がけ、何度かの改良を加えて計測カードのフォーマットを決めた。(付録)

このカードは次に述べるトップダウン方式で計測を記録できるようにしたため、実際には『計測』ではなく『状態調べ』であるものも1枚のカードに記録することになる。したがって『計測カード』という名前は必ずしも正確ではないが、本研究では『状態調べ』も

計測カードに書くことにする。

名前： 計測の名前

対象物： 何に対して計測を行うか

計測内容： 一言でいって何を計測しているかがよくわかるようなコメントをそえる。
カードを整理する時に使う。知識獲得とは直接関係ない。

前処理： 計測の前に準備しておくことがあれば書く。『取り付け』とか『電源投入』
というような『作業場の基礎知識』に属する知識である。カードを書き始める
時にそれが計測の本質と関係ないとわかっていれば何も書かなくてよいが、
整理している間に気付くことが多いので参考に書ける欄を設けた。

手順： ここには単一の計測でもいくつかの計測の集合でも書ける。また状態を調べ
ている時はその内容も書ける。『計測』か『状態』かの区別を記し、下位の
(呼び出している計測の)番号を書く。結果欄には結果を書く。

結果の解釈・判断に使う知識： 結果をどのように解釈・判断するかを書く。

わかったこと： 何がわかったかを書く。

モデルを解釈する知識： 計測によってモデルにある事実の集合に新たに追加される。
計画はそのモデルをどのように解釈するのかの知識とそれか
らわかることを書く。

後処理： 計測の後に処置しておくべきことがあれば書く。前処理と同様に『作業場の
基礎知識』に属する知識である。前処理と同様の理由で何も書かなくてよい
が、整理している間に気付くことが多いので参考に書ける欄を設けた。

計画： 『本件からわかったこと』と『モデルからわかったこと』の2つから、次に
何の計測(状態調べ)を行うかを書く。

モデル・チェックリスト： 計測の結果がモデルを形成しているものなら『モデル』、
チェックリストの項目を調べた場合は『チェックリスト』
をそれぞれマークする。

3. 計測項目のグルーピング

ある程度まとまった計測の集合を1つにして何個かのグループに分ける。このグルーピ
ングは主観的なものでもよい。だいたい3~4個の計測でまとめるとやりやすい。

こうしてできた計測のグループは「何かの状態を調べている」ことが多い。さらにこの
『状態』をグルーピングして状態をとりまとめた状態にグルーピングする。これ以上でき
ないというところまでボトムアップしていく。この時、計測間に不自然なものはさま
まについてそれが単一のまま浮いてしまうことがあるが気にしないでやる。それは「機種ご
との状態把握のパターン」のチェック項目であったり、その時は使わないが一応モデルに事
実とし登録しておいて後で利用するという計測が多い。状態と計測が同レベルにあっても、
すなわち『手順』の欄に状態調べと単一の計測が書かれていてもかまわない。

4. 状態ごとのカードおこし

状態のグループでカードを書き始める。『手順2』の欄には調べたい状態項目がくる。単一の計測がくることもある。状態・計測の別を記し、『結果』欄には得られた状態または計測結果を書く。その結果群からわかることと、それに用いる知識を書く。その時点でモデルを参照してわかったことがあれば、それを導く知識とともに書く。モデルは事実の集合であるから、症状以外の計測結果すべてが対象となる。

最後に、『わかったこと』から次に何を計測するか、何を調べるかを計画欄に書く。

以上で1枚のカードが書き終わる。『手順』欄で『計測・状態の別』が『状態』になっているものは先にしたグルーピングに従ってさらに下位のカードを同様の方法で書き、計測だけの項目になるまで続ける。

上位カードの結果欄に書かれていることと、すぐ下位の『わかったこと』の内容は一致する。

4. 2. 4 カードとエディタ

カードを整理・作成していると、同じようなことを何度も書くことがたびたびあった。またトップダウン方式で書くのでどうしても上位・下位の関係が出てしまう。まとまりのある計測を1つととらえていくこともある。そのため紙で『計測カード』を実現していると階層関係がわかりにくいし、訂正・追加もやりにくい。階層関係を取り入れないのであれば、一般のワードプロセッサについている「差し込み印刷」の機能を応用することもできる。データを別に作成しカードの罫型に差し込みで印刷するのである。しかしワープロにデータの検索機構を持たないことが多いとか、速度の問題、イメージがでないので印刷しないと内容がわかりにくいなどの欠点がある。

最近ではコンピュータのエディタが使いよくなってきて、階層的にファイルをアクセスできるようなマクロが書けるものもある。そういった既存のエディタを有効的に利用することによってプロトタイピングの段階での作業が楽になるはずである。知識をルール化してコンピュータにのせようという場合もあらためて入力するというよりもそのファイルをフィルタにかけるなり、何らかの処理をほどこすことによって楽に実現できる可能性がある。

より適切なインタフェースとしての知識エディタの開発が望まれる。

4.3 大局的・局所的診断とその知識

3.6.3 で状態把握のパターン、4.1.5 で症状・原因パターンと原因について述べた。ここでこれらのパターンとその1つの見方、大局的・局所的診断について考えてみる。

症状・状態パターンが機種ごとにあるにあって症状によらずだいたいそれらの項目、『症状』なら『ある・なし』、『状態』なら『状態』を調べる。そして故障のあるブロックを予想する。あるブロックにねらいをつけると、そのブロックだけに集中して計測を行い始める。このブロックを予想する部分とブロックに集中する部分をそれぞれ大局的診断相、局所的診断相と名づけるとおおまかに次の性質があるといえる。

A：大局的診断相

マクロ的
観念的
直観的
経験的

B：局所的診断相

ミクロ的
具体的
客観的
理論的

症状・状態パターンから故障ブロックを決定するとき、全体的に（マクロ的に）見渡し、（観念的あるいは直観的に）どのへんが悪いかを確固たる根拠なしに、しかし自らの経験に基づいて判断を下している。

一方、あるブロックに原因があるという仮定で、そのブロックを詳しく見ているときは比較的個々の部品に（ミクロ的に）注目し対象をはっきりと（具体的に）意識して計測し、次の計測の意味を（客観的に）とらえ、（理論的に）筋道だてて考えながら計測を行っている。

A相とB相の境界は明確ではなく連続的である。A相・B相という2つの状態があるわけではなく、どちらかという方向性をもった意識の流れと考えることができる。修理の時には徐々に移り変わっていくと考えられる。B相が進んで故障原因が見つからなかった時はまたA相に戻ってくる。実際にはAとBの間を何度も行き来していることもある。それはA相において仮定できる故障のブロックが複数あった場合で、予想したブロックに故障がなかった時、別のブロックを調べにいくというのが何度か繰り返されていると考えられる。

カードを整理していったB相に入ってくるとカードの『計画』欄がだんだん書きにくくなっていく。ある程度以上故障ブロックがせぼめられるとDSPのサイクルがはっきりとでないのである。「なぜその計測をしたのか」ときかれるとそれなりに答えられるのだがもっともらしさに欠ける。そのあたりでの計測は順序が重要であるというより、むしろ集められる情報はすべて集めようという意識が強いように思う。人間が一度にとらえられる範囲があって（もちろん個人差はあるが）その範囲でものを考える時はすべての状況、本研究でいうところの『事実』や『状態』を考慮しているといえる。

したがって、これらの特徴を生かして知識を記述していくことがより素直で獲得しやすいかつ実践的な方法といえる。具体的には、症状・状態パターンに照らし合わせて計測カードをA相、B相に区切り、A相からチェックリストを作りB相は考えられるすべての原

因をリストアップするところから始まる。詳しくは次章で述べるがこの積極的な区切りは知識の表現のために大変有効である。

4.4 まとめ

修理用の知識を獲得するためには、問題の領域をはっきりさせる必要がある。本研究では、オーディオ機器の中でも、特に電子回路部だけに故障原因があると仮定し、『作業場の基礎知識』を身につけた技術者、あるいは『作業場の基礎知識』に関して支援されうる技術者、すなわち必要な計測は自由に行えるという前提で、故障診断支援をするために必要な修理用知識を考えた。

領域における知識のとらえ方として『事実』、『状態』、『症状』、『原因』、『中間原因』などの言葉を定義した。

『事実』は外には現われないが、何らかの手段で客観的に得られるもので、電圧、電流、抵抗などの値、導通、部品の良否などをさす。

『状態』は事実の組み合わせから生じている様子であり、表に現われない。事実と状態の区別は知識獲得の際、特に重要である。事実(群)に対応する状態に名前がついているものはわかりやすいが、そうでないものはその表現と事実との対応が曖昧になりがちのため、熟練技術者にその対応意識を喚起させる必要がある。『状態』は正常な状態も含む。

『症状』は表に現われる様子で特に診断の初期に使われるものをさす。

『原因』は部品不良、導通不良、調整不良のいずれかである。

『中間原因』は『状態』の中でも不良の状態をさす。

以上をまとめると故障というのは、

「部品不良などの『原因』が『事実』(計測できる)を生じ、その組み合わせが不良な『状態』を生じて、いくつかの『中間原因』を経て、『症状』として表に出る。」と表現できる。

修理を始める時、いくつかの症状と、症状によらず必ず一通り調べるという状態項目がある。それらをまとめて『チェックリスト』と呼ぶことにした。また、人間が修理する場合、計測が動機付けのないものであっても、その結果(事実)は頭のどこかに蓄えられる。したがって、知識を獲得する時も、そのような枠組みがあると便利である。本研究では「事実を蓄えておくもの」を『モデル』と呼ぶことにした。

本研究では、私が修理に携わった時の簡単なメモ(修理メモ)をもとに修理全体を回想して、修理用知識を獲得することにしたが、そこでの第一の問題点は、客観的な手がかりがつかみにくいことであった。そこで「頭で考えていたこと」を思い出すより、「行動について」思い出す方が、やりやすく適切であろうと考え、『計測』という言葉を広い意味で「人間の五感および測定器具を使って情報を得るすべての行為」と定義した。

この『計測』の定義に従い、『計測カード』というものを考案し、『計測』を中心としたサイクルを考え、そこで使われる知識を整理した。先に述べた『原因』から『症状』までの枠組みは、この『計測カード』の整理の過程からはっきりしていったことがらである。さらに、「直観的・経験的な知識を多く用いる時期」を『大局的診断相』と、「客観的・理論的な知識を多く用いる時期」を『局所的診断相』と名づけ、この2つの診断相における性格を取り扱った。それにもとづいて修理の流れを見ることによって、より一般的な修

理戦略の鋪型を提示した。

5 獲得した知識の表現と実装

修理メモから計測カードをおこすことによって、具体的な修理例として作業の進み方と判断に使われた知識が記述された。これはあくまで『例』であってそのままの形では一般的な知識として書けない。

本章では支援に使えるレベルまでの修理用知識の一般的表現方法を考え、それを知識ベースシステムに実装してテストする。当研究室の大学院生、森田敬之氏が開発した『診断型知識ベースシステム構築ツール』[6], [7] を用いて故障診断のための知識ベースシステムのプロトタイピングを試み、知識の表現と実装方法についての評価を行う。

5.1 知識の表現

1つの修理例だけでなく、ある程度まとまった形で一般的な知識を書き出さないとその知識は応用できないしコンピュータ上に実装することもできない。ここでは具体例からの知識をどのように拡大していくのかを述べる。

5.1.1 相による分割

4.3 で述べた大局的診断と局所的診断の考え方にそって修理の流れを分割してみる。

症状と状態把握のパターン、すなわちチェックリストの項目とそれらの結果からどのブロックが悪いかを予想する。〔大局的診断相〕

あるブロックのどこかに故障原因があると仮定してそのブロック内を詳しく計測して故障原因を見つけようとする。〔局所的診断相〕

大局的診断相における知識はその性格（観念的、直観的、経験的）から計測カードから一般化するのが難しい。したがってチェックリストの項目としては私の経験により作成したものを採用することにした。またそれらの結果から故障のあるブロックを仮定するための知識も同様にして得る。

一方、局所的診断相における知識はその性格（具体的、客観的、理論的）から比較的緻密な形で書くことができる。具体的な部品に関する測定などを含むため、おのおのの性格は似ていても機種モデルに固有な知識となる。

5.1.2 チェックリスト

チェックリストの項目はだいたい機種モデルの仕様に関係なく設定できるのであるが、特に機種モデルの仕様に依存したチェック項目を設ける必要があるのであれば、機種モデルごとにチェックリストを作ればよい。いずれにせよ、これは本質的な問題ではない。

現実の修理例をみると必ずしもすべてのチェック項目がチェックされているわけではないし、順序も特に定まっていない。しかし、それは技術者がきまぐれであったためや、

「たまたま調べなかっただけ」で普通は一通り調べると考えてもさしつかえない。チェックリストの中のある項目が診断がかなり進んでから調べられることがあるが、それは必要に迫られてやったというのが多い。

5. 1. 3 局所的診断相の知識記述

今、ある装置のあるブロックのどこかに故障原因があると仮定されたとする。原因はそのブロック内のどこかにあるはずである。原因が部品不良だけであると考えても理論的に可能な原因の数は部品の数だけあると考えられるが現実にはそのようには考えない。熟練技術者は典型的な原因のいくつかと、時々現われる原因のいくつかを知っているが、ほとんど考えられない原因も知っている。例えばあるモジュールの電源電圧が決まっている場合、「その電圧がある抵抗に直接かかったとしてもその抵抗がある程度以上の値であれば耐電力値を越えることはないので壊れることはまずない。」というような例である。同様にめったに壊れない部品というのがかなりあるのでそれらは原因として候補にいけない。もちろん「例外的に」壊れるということもあるが、極端にまれな例が多いと知識を書きにくくするだけでなく、システムの性能を悪くする（現実的でなくなる）可能性もあるから注意を要する。

ブロックが比較的大きいと部品の点数も多く、やりにくいため中間原因をいくつか設定する。中間原因とは 4.1.3 で述べたような『不良の状態』である。中間原因はすなわち「このブロックのどこかが故障というとき、だいたいどういう状態が主に見られますか」という質問の答えである。

ここでの知識記述は段階的に行いにくい。4.3 で述べたようにある程度以上せまい範囲では人間は物事を一気に考えられるからその範囲内では計測は並列的に行われ、その結果から何かの判断が下されると考えてよい。今比較的せまい範囲のモジュールを考えている。与えられているのはそのモジュールの『不良な状態』とその状態を起こしうる最終的な『原因』である。（ここでの原因は『部品不良』に限らなくてもよい。）

さて『原因』と『不良な状態』との関係はどのようになっているかをもう一度考えると次のようになる。

「原因はある事実を生みだし、事実の組み合わせは状態を生じる。また状態と事実の組み合わせからも状態を生じる。」

したがって『不良な状態』から『原因』をつきとめるにはそれらの関係を記述すればよいことになる。モジュールがある程度小さくなると、それは局所的診断相の性格が強くなるわけであるからより具体的、客観的、理論的にその関係を書くことが『修理用知識の獲得』のための必要かつ十分な条件である。これは一般にプロダクションルール¹¹⁾と呼ばれる "If~then" の形のルール集合で書ける。

「ある不良の状態」すなわち中間原因を導くための「可能性のある原因群」との関係を書き、これをプロダクションルールで書くことによって、局所的診断相の知識は記述できることになる。

5.2 知識の実装

5.2.1 テスト環境

・知識ベースシステム構築用ツール

実装テストに際しては、現実のコンピュータのハードウェア・ソフトウェアの環境が必要となる。本研究では、当研究室の森田氏が開発した『知識ベースシステム構築用ツール』（以下『ツール』）を利用することにする。

『ツール』はオブジェクト指向言語の一つである『Smalltalk-80』⁽⁸⁾ 上で開発された、ルールシステムとフレームシステム⁽¹¹⁾ をあわせもつ診断用の強力なツールである。

・実装環境

『ツール』は『Smalltalk-80』でかかれている。ブラウザと呼ばれる一種のエディタが備わっているものの完全な形でエンドユーザに解放されるまでには至っていない。ツールとしての柔軟度が高いためにいろいろなことができる反面、わく外のことをしようとすると、そのつど手続き（もちろん『Smalltalk-80』で）を書かなければならない。知識を記述する人が Smalltalk-80 に精通していればよいが、そうでない場合、例えば私が言語依存の部分に立ち入ろうとしたとき問題が生じる。

そこで本研究では、森田氏との話し合いを重ね、本研究の主旨を説明した上で知識の実装に必要なシンタックス・テンプレートを示してもらい、それにしたがって知識を記述することにした。知識は一般のコンピュータのファイルという形で作成し、それを Smalltalk-80 のマシンに転送し Smalltalk-80 からそのファイルを読み込んでテストした。

なお「Item」、「Inquiry」、「Cause」、「フレーム」、「Rule Base」は『ツール』における知識表現形式の名前である。

『Smalltalk-80』については参考文献を参照されたい。

5. 2. 2 チェックリストとアイテム

4.1 で「症状によらず必ず一通り調べる」症状と状態の項目をチェックリストと呼ぶことにし、4.3 でそれが大局的な診断の相にあることを述べた。実際の修理例からだいたいチェックリストの項目は調べていることもわかった。チェックリストは機種ごとあるいは機種モデルごとに作成することになる。

『ツール』の中でこのチェックリストは『アイテムリスト』に埋め込むことで実現できる。

まず、チェックリスト項目の中の症状と状態を" Yes"、" No" で答えられる形に整える。これは多少あらっばいように思えるかもしれないが、診断相の性格からして『おおまかなつかみ方』というのはそれほど全体に影響を与えない。また考えられる症状もそれほど多くないのでそれらすべてを列記することができる。例えば『音』に関する症状は『出る』、『出ない』の他に『歪む』とか『小さい』とか考えられるが、必要であればそれぞれをアイテムリストに登録することによって考えられる症状をほぼ網羅することができる。『ツール』の仕様に合わせてチェック項目に適当な名前をつけて『Item』に登録する。

次にこのアイテムリストのそれぞれの項目に" Yes"、" No" で答えられる『質問文』を用意する。この『ツール』ではユーザに対して質問を行う手続きが用意されているのでそれを利用するのであるが、その手続きがアイテム1つ1つに対して『質問文』を必要とするので先の症状・状態項目に合わせて質問文を作成し『Inquiry』に登録する。

この質問を行う手続きはアイテムリストの値が参照されたとき起動されるので、この手続きを別のものにしてやることによっていろいろな支援の可能性がでてくる。症状や状態の調べ方を説明するもの（これは『作業場の基礎知識』に属する知識であるが）を設定しておけば簡単な支援ができる。ここでは本研究の主旨にしたがって、簡単な質問文で済ますことにする。

アイテムリストは知識ベースシステムが初期診断で仮説を生成するとき参照され、もし値がないときは先の質問文によって" Yes"、" No" の入力があるがされる。ここでいう『仮説』とは「どこのブロックに故障の可能性があるか」を決めるものであり4.1 で述べた「症状・状態パターンから故障ブロックを仮定する」挙動に相当する。

したがって次に行うべき作業はその「症状・状態パターン」を記述することである。ツールの仕様によりこれは" Possible" という述語が与えられているのでそれを用いて「可能性のある原因ブロック」を記述しそれを then 部にもたせ、if 部には「症状・状態パターンの組み合わせ」を記述する。先のアイテムに対する質問の制限から Item で参照される値は" true" か" false" になっている。例えば

(Item #Sound false)

というのは「音が出ますか」に対して" No" と答えた場合に" false" を返すのでこの述語は真となる。

また正常な状態も積極的に利用することによって故障ブロックの可能性が限定しやすく

なるのはいうまでもない。例えば

```
( Item #ResponseMainSp true )
```

は「メインアンプ入力とスピーカ間」に応答があれば真を返す。

以上の方法でチェックリストとその故障ブロックパターンを記述した。[付録7]

(1) チェックリスト→アイテムリストの記述

```
ItemDescription at: 'PilotLamp' asSymbol
definitionString: 'PilotLamp
  valueType: Boolean
  valueCheck: nil
  requisite: false
  ifNeeded: #Inquire withArguments: #()
  ifAdded: nil withArguments: #()
  note: ``パイロットランプ``
```

説明

```
'PilotLamp' : パイロットランプがつくかどうかの項目名
#inquire    : 質問文を用いることを示す
```

(2) 質問文の記述

```
InquiryDescription at: 'Item-PilotLamp' asSymbol
definitionString: 'Item-PilotLamp
  "Yes or No"
  message: ``Is the pilot lamp on?``
  duration: #temporary
  assertion: false
  note: ``パイロットランプがついていますか``
```

説明

```
'Item-PilotLamp' : アイテム名「パイロットランプ」
message: ``Is the pilot lamp on?`` : 実際に質問される文を書く
```

(3) 症状・状態パターンと可能性のある原因ブロックの関心の記述 (ルール)

```

( RuleBaseDescriber
  name: #FirstConsultation
  definition: ` workingMemory: SystemWorkingMemory
                note: `` ``
  rules: #( `rule1:
              if: #( ``@( Item #Sound false )`` ``@( Item #ResponseLineSp false )``
                  ``@( Item #ResponseMainSp true )`` ``@( Item #PilotLamp true )``
                  ``@( Item #Protector true )`` )
              then: #( ``@( Possible #FaultInPowerForPre )``
                      ``@( Possible #FaultInPreBlock )`` )
              explain: `` ``
            `rule2:
              if: #( ``@( Item #Sound false )`` ``@( Item #Smell true )``
                  ``@( Item #ResponseMainSp false )`` )
              then: #( ``@( Possible #FaultInMainBlock )`` )
              explain: `` ``
            ) )

```

rule1の説明

「音が出ない、LineIn-Speaker間に応答がない、MainIn-Speaker間に応答がある、パイロットランプ点灯、プロテクタ正常動作」ならば

「プリアンプ用電源またはプリアンプに故障の原因の可能性がある。」

5. 2. 3 モデルとフレーム

4.1 で述べたように、修理の中で計測からわかる事実をどこかに蓄えておくところが必要でありそれを『モデル』と呼ぶことにした。ここではこのモデルをフレームによって実現する方法を述べる。

フレームにはスロットと呼ばれるものがあり、これを個々の部品に対応づける。スロット名には個々の部品名が入る。

各スロットには複数のファセットがあり、value (値) や if Needed などの付加手続きがある。value には次元を持つ量 [V, A, Ω, W] や部品の良否に関して "true"、"false" を持つ。if Needed には値が参照されたときに起動する手続きを書くことができる。

ここでは 5.2.2 と同様に質問文を用意しその質問文によって入力をうながしてスロットの値を得る手続きがあるのでそれを利用することにする。

5. 2. 4 原因リストとルール

局所的診断相の知識をどのように実現するかであるが、『原因リスト』と『ルール』によって記述する。

対象となる故障ブロックにおいて可能性のある原因をリストアップし『Cause』に登録する。

```
CauseDescription at: 'BrokenCl' asSymbol
definitionString: 'BrokenCl
  ifPossible: nil withArguments: nil
  ifAffirmed: nil withArguments: nil
  message: ''Cl is wrong.''
  needOfChecking: true
  note: ''Cl が壊れています''
```

またあるブロックに故障の可能性があるときそれは『原因』ではなく、『中間原因』であるが、それも『Cause』に登録する。故障の可能性が認められた時に、利用する診断戦略とルールベースを指定し局所的診断相に移ることができるようにする。

```
CauseDescription at: 'FaultInPowerForPre' asSymbol
definitionString: 'FaultInPowerForPre
  ifPossible: #backwardInference:tactics:assumption:
    withArguments: #( PowerBlockForPreConsultation FirstMatch
      ''@(Cause *)'')
```

```

ifAffirmed: nil withArguments: nil
message:  "The Power supply unit for Pre-block is wrong."
needOfChecking:  false
note:  " "

```

もしこの中間原因 `FaultInPowerForPre` (プリアンプ用電源不良) の可能性が出たら (故障ブロックがここにあると思われるときは) `PowerBlockForPreConsultation` というルールベースを用いて `@(Cause *)` を仮説として `backwardInference` (後ろ向き推論) で行う。(このとき、競合解消戦略は `FirstMatch` (ファーストマッチ) を指定している。)

次にこれらの原因群と『不良状態』をむすびつける知識をルールの形で書く。このとき、中間原因を設定したければいくつ置いてもよい。例えば、付録の例では "Symptom" "Possibility" という2つの語を用いているが、この2つは共に中間原因を表している。この2つの名前はシステムに固有ではなく適当に付けた名前である。

ルール群はまとめて『ルールベース』と呼ばれ、1つのルールベースが1つの故障ブロックと原因との関係を示している。

モデルの値の参照は `AccessValue` という述語を用いて

```

@( AccessValue #Amp #vq1 (equal 0) )
      (アンプフレームのトランジスタQ1のエミッタ電圧が0Vなら真を返す)

```

のような形で書く。もし値がなければフレームの付加手続きが自動的に起動されユーザからの入力を求めるようになっている。[付録7]

ルールベースの例

```

( RuleBaseDescriptor
name: #PowerBlockForPreConsultation
definition:  workingMemory: SystemWorkingMemory
note:  "This is a set of rules for consulting about an amplifier."
rules:  #( 'rule1
if: #( " @( AccessValue #Amp #vq1 (less 5) )" )
then: #( " @( Symptom #VQ1eIsWrong )" )
explain:  " Q1のエミッタ電圧が5Vより小ならエミッタ電圧不良である "
'rule2
if: #( " @( Symptom #VQ1eIsWrong )"
" @( AccessValue #Amp #vc2 (greater 40) )"

```

```

    ``@( AccessValue #Amp #q1 true )`` )
then: #( ``@( Possibility #VSourceWrong )`` )
explain: `` エミッタ電圧不良かつC 2 の電圧が40Vより大かつQ 1 が良ならば
          定電圧源不良の可能性がある ``

```

rule3

```

if: #( ``@( AccessValue #Amp #vdz1 (equal 0) )`` )
then: #( ``@( Possibility #VSourceWrong )`` )
explain: `` ツェナーダイオードD z 1 の電圧が0Vならば
          定電圧源不良の可能性がある ``

```

DiodeDz1

```

if: #( ``@( AccessValue #Amp #shortdz1 true )``
      ``@( Possibility #VSourceWrong )``
      ``@( AccessValue #Amp #c3 true )`` )
then: #( ``@( Cause #BrokenDz1 )`` )
explain: `` ツェナーダイオードD z 1 がショートしている
          かつ定電圧源不良の可能性がある
          かつC 3 は良ならば原因はツェナーダイオードD z 1 である ``

```

.

.

.

)

)

5. 2. 5 テスト結果と応用

Item、Inquiry、フレーム、Cause、RuleBaseをそれぞれ整え、ファイルとして『Smalltalk-80』マシンに転送しそれを読み込んで診断を行ってみた。

まず、大局的な診断相に対応付けられる症状・状態パターンの認識であるが指定した全項目に対して入力がある。いろんなケースを想定して入力してみた結果予想される故障ブロックを特定できた。

この相では特殊な例を入力すると故障ブロックを特定できなかったが、これはルールがすべての組み合わせを考慮していないからである。すべての組み合わせに対応するためにはルールを多く書かなければならない。しかしあまり非現実的なものを取り入れるとほとんど使われないルールの占める割合が多くなりシステムの能率が悪くなる。ここではいかに経験的に熟したルールを書けるかどうかがその性能を決めるといってもよい。

次に局所的診断相における診断がどのように行われたかであるが、原因に関してはルール通りある程度正確に確定することができた。やはりここでも非現実的な状況に対しては原因が見つからないということが起こる。ルール数を多くしてより細かなルールを書くことによって比較的改善されるはずであるが、いたずらにルールを多くするとムダな計測を多くうながしたり診断に時間がかかったりすることがあるので、ルールベースの大きさの設定をよく考える必要がある。

比較的一般的な故障原因であるにもかかわらず、故障原因をつきとめることができないことがあるが、それはルールの書き方に問題があると考えてよい。『ツール』に備わっている『履歴』を参考に、どこで原因もしくは中間原因の検証に失敗したかをつきとめ、欠落しているルールを付け加えたり不自然なルールを現実にそくするように修正することによって改善することができた。なぜルールが欠落していたり不自然であったりするかであるが、4.1.2で述べたように事実と状態を取り違えたままでルールを書きってしまった場合によく起こる。そういうとき「不良な状態は何か」を考えることによって、それを中間原因としてとらえることによりすっきりとしたルールになることがしばしばあった。ルールベースの中で事実と状態が1対1になっているものがあって一見冗長な表現のように見えるが、他の部分での不自然さを取り除くために大きく役立っている。

モデルに関して、フレームで表現する方法はうまくなじんでいると感じた。一度計測したことはもう一度計測しないというのは（人間は結果をよく忘れるのでまたやるが）、人間がやる修理の中でのモデル形成と似ている。あるルールを検証しようとした時、必要な値がすでにモデル内にあれば改めて計測しないし、そこでは不足な計測だけを要求してくる。こういうときディスプレイ上に表示されるルールの検証結果と計測の結果の入力要求からその挙動をよく確認することができた。

テストしてみて少し現実の修理の流れと異なると感じた点は計測の順序である。普通トランジスタのエミッタ電圧を測定したらついでにコレクタ電圧とベース電圧も測定することが多い。それらの結果は当座は使わないかもしれないが後で使うかもしれないので頭の中のモデルに一時蓄えられることになる。普通に局所的診断相のルールを書くときこういっ

た『動機付けのない計測』というのはいないため、この知識で診断を始めると「あっちゅこっちを必要とき測定する」ということになる。これは故障診断のための知識としてはかまわないのであるが、修理を支援するという立場からはユーザに「やりにくさ」を与える可能性があり、あまり好ましくない。おおざっぱに「計測点の近いものはすべて同時に計測しておくようにルールを書くべきである」とまではいかないにせよ、何回かのテストにより明らかにフィーリングの悪い所は改善する余地があると思う。具体的にはフレームの値の参照時、そのルールで利用する値だけでなく近くの測定点も調べるようにうながすことで実現できる。『ツール』ではワイルドカードとして“*”（アスタリスク）が使える。仮にQ1のコレクタ電圧だけがそのルールで必要（40V以上かどうかを知りたい）だったとしても

```
@( AccessValue #Amp #vQ1c (greater 40) )
@( AccessValue #Amp #vQ1e      *      )
@( AccessValue #Amp #vQ1b      *      )
```

のようにしておけば、値は何でもよいので同時に計測できることになる。ただしこの方法だとQ1の電圧を参照するルールすべてに同様のことを記述しておかなければならないことになり、ルールが大きくなりがちである。

故障原因が1箇所だけでなく並列的、連鎖的であることがあると3.6.6で述べた。本研究の知識記述ではこれらの対応方法について述べてこなかったが、ここでこれらの故障診断の実現方法について述べる。

並列的な故障はルールのthen部におけるCauseをただ並列的に記述することで実現できる。すなわち

```
'DiodeDZ1FET1
  if: #( ''@( AccessValue #Amp #shortdz1 true )''
        ''@( Possibility #VSourceWrong )''
        ''@( AccessValue #Amp #c3 true )'' )
  then: #( ''@( Cause #BrokenDZ1 )'' )
         ''@( Cause #BrokenFET1 )'' )
  explain: '' ツェナーダイオードDz1がショートしている
            かつ定電圧源不良の可能性がある
            かつC3は良ならば原因はツェナーダイオードDz1
            とFET1である ''
```

のようにする。

一方、連鎖的な故障に関しては当初それをルールで書くことでしか対応できなかった。

部品が壊れているのわかっているのならそれは故障診断にはならないわけで、故障箇所がルールに書かれているのはどうも不自然さがあると感じていた。そこで森田氏と協議してそのへんの機構を盛り込めないかを検討した結果、「原因の能動性」に着目してそれを実現できる可能性を開いた。もともと『ツール』は「『原因』が自分自身を証明する手立てを知っている」という大きな特徴があり、「『原因』が能動的にふるまう」という性格を持っているのであるが、さらにこの能動性を拡張して「ある原因が肯定された時自動的に起動する付加手続き」を書けるようにした。この付加手続きは「ある原因が肯定された時何をやるかに関して（『Smalltalk-80』に依存してしまうかもしれないが）いろいろ書くことができる。例えば `if Possible` のように戦略とルールベースを指定して、故障連鎖のある故障ブロックの故障を診断することができる。故障連鎖に関する知識は『原因』に記述されることになる。

例えば「フューズが切れている（第一の故障原因）場合、電源ブロックが故障である（そのどこかにある原因が第二の故障原因）」という例で、「フューズ切れ」を1つの症状として大局的診断相に盛り込むこともできるが、「フューズ切れ」を原因（すなわち症状「パイロットランプがつかない」などから導かれる原因）とみて、その原因が肯定された時、「電源ブロックの故障に関するルールベース」が起動されるように `Cause` に原因を登録すればよい。すなわち

```
CauseDescription at: 'BlownFuse' asSymbol
definitionString: 'BlownFuse
  ifPossible: nil withArguments: nil
  ifAffirmed: #backwardInference:tactics:assumption:
    withArguments: #( PowerBlockConsultation FirstMatch ``@(Cause *)``)
  message: `` The fuse has blown. We will check the Power Supply Block. ``
  needOfChecking: true
  note: `` フューズが飛んでいます。電源ブロックの故障原因を調べます ``
```

のようになる。

5.3 まとめ

計測カードによって整理された修理例から、一般的な診断のための知識表現を考え、知識ベースシステムに実装してテストした。実装テストに際しては、当研究室の森田敬之氏が開発した『診断型知識ベースシステム構築ツール』を用いた。

まず、症状と状態把握のパターン、すなわちチェックリストの項目とそれらの結果からどこのブロックが悪いかを予想する『大局的診断相』の知識をアイテムリストとルールによって実現した。『ツール』に用意されているPossibleという述語によって、経験的に得られた「～ならば～の可能性がある。」という知識を表現した。

次に、あるブロックのどこかに故障があると仮定して、そのブロック内を詳しく計測して原因をつきとめていく『局所的診断相』の知識をルールで実現した。その際、適宜『中間原因』を設定して『不良の状態』をまとめて記述することができたため、より自然な知識が書けた。また、『モデル』はフレームによって実現し、現実の人間が頭の中に持つモデルに近いモデルであることを確認した。

大局的診断相の知識は経験的でおおざっぱなものであるが、診断に重要な位置を占めるので、チェックリストの項目の選択と、そのパターンと故障ブロックを結びつけるルールは慎重に書かれなければならない。

局所的診断相で不自然な知識表現が出た時は、そこに『中間原因』をおいて、それをもとに周囲のルールを整理すると不自然さがなくなることが多いことがわかった。

『動機付けのない計測』はフレームへのアクセスで「値が何でもよい」参照方法をとることによって一部実現できるが、いたずらにとりいれるとルールが大きくなるので、何回かテストしながら、作業がやりやすくなると思われる所に組み入れるとよい。目安として、トランジスタ、抵抗、コンデンサの電圧参照を一度にやってしまうところから始めるのも一つの方法である。

連鎖的な故障は当初ルールでしか書くことができなかった。しかし確定している『故障原因』がルールの中にあるのは不自然であり、改善方法をいろいろ検討した結果、『原因』の能動性に着目し、ただ「原因が自分自身を証明する方法を知っている」というだけでなく、故障連鎖の知識を原因に持たせることによって、すなわち「原因が確定したらその次に何をやるか」を原因に記述することにより、現実に近い形で故障連鎖に対応することができるようになった。

6 結び

6.1 研究の到達点

電子機器の故障修理を支援する方法はいくつか考えられるが、測定方法などに関して支援するための『作業場の基礎知識』と、故障診断を支援するための『修理用知識』の枠組みを考えることによって、もっとも引き出しにくいといわれる熟練技術者の修理ノウハウの部分に焦点をあてることができた。

実際の修理において、熟練技術者の頭の中では常に「故障原因はどこかを見つける」という大きな問題だけが意識されているので、曖昧さ、不自然さを含む多くの特徴を適切にとらえ、その合理的な表現を考えることが、体系的に知識を獲得しようとする時、大切であることがわかった。

まず、「～を調べるため」、「～を証明するため」に行う計測に対して、「とりあえず」、「ついでに」行う動機付けのない計測が予想以上に多かった。しかし、それらすべてが無意味なものばかりではなく、その結果が頭の中に蓄えられ、モデルとして形成され、後の診断に使われるものがあり、重要な働きをしていることがわかった。

故障原因が複数の場合でも、熟練技術者の頭の中では原因と原因を結び付ける知識によって1つの故障と意識されていることもわかった。また、事実と状態の区別を明確に意識させることにより、中間原因の位置付けを喚起させ、客観的なルールを引き出しやすくなった。

診断の初期には、症状といくつかの決まった状態項目を調べ、それによって故障ブロックを予想する。その時に使う症状と状態の項目をあわせたものをチェックリストと呼ぶことにした。チェックリストから故障ブロックを予想する時に使われる知識は、直観的・経験的であり、このような性格を持つ診断の時期を『大局的診断相』と名づけた。また故障ブロックを特定した後、そのブロックを詳しく見ている時に使われる知識は、客観的・理論的であり、このような性格を持つ診断の時期を『局所的診断相』と名づけた。

計測という言葉の概念を広げ「五感および測定器具を使って情報を得る行為」と定義することによって、修理全体を計測を中心としたサイクルで整理することができた。計測カードはそのために考案した1つの手段であり、『計測』という側面から修理を見るための枠組みとしての働きをした。

獲得した知識の実装には、当研究室の森田敬之氏の開発した『診断型知識ベースシステム構築ツール』を用いて、知識ベースシステムのプロトタイピングを試みた。モデルをフレームで、チェックリスをアイテムで実現し、チェックリストの項目の結果と故障ブロックを結び付ける知識、原因と不良状態を結び付ける知識、をそれぞれルールで実現した。

大局的診断相の知識は、まさに経験的な知識の性能によるところが大きく、チェックリストの項目の選択とルールの書き方に大きく依存する。10個程度の項目数と5個程度のブロック数であれば、それほど冗長で煩わしい計測を必要とすることはなかった。

局所的診断相においては、適切な中間原因の設定により、知識体系が整い、わかりやす

い知識記述ができた。現実的な故障例に対しては、ほぼ原因をつきとめることができた。

フレームによる修理モデルの実現によって、現実の技術者が持つモデルの挙動に近いものが得られた。またフレームの値参照の方法を工夫することによって、『動機付けのない計測』の埋め込みも可能になった。原因の能動性を利用し、原因同士を結び付ける知識を記述することにより、並列的故障、連鎖的故障にも対応することができ、現実の故障のとらえ方に近い形で知識を記述できるようになった。

6.2 今後の課題

本研究では『計測』というものを中心に、現実の修理をふりかえり、その中でさまざまな特徴を発見し、修理の知識を体系的に獲得するための枠組みを作ることができた。しかし、具体例からの一般化に関しては、手作業だけで行うことしかできなかった。今後、多くの故障例を集め、修理用知識の性能向上のために、より客観的で能率的な知識の一般化の方法が見いだされなければならない。最終的には知識獲得の（半）自動化が望まれるが、その第一ステップとして、本研究の『計測カード』を計算機上のエディタで実現し改良することが考えられる。

テストに用いた知識の規模は比較的小さいものであったため、現実の修理に近い挙動が得られたが、このままの記述方法で、規模を大きくした時に生じうる問題についてはあまりわかっていない。中間原因の種類が増えた時、ルールの数が多くなった時にどうなるか、わかりやすさ、書きやすさはどうかなどはツールに依存することでもあるが、知識表現そのものに含まれる問題がないかどうかよく検討されるべきである。

次に、支援システム全体を考えた今後の方向であるが、前に述べた2つの問題点の解決はもちろんのこと、『作業場の基礎知識』に属する知識の取り扱いをどうするかである。本研究ではその主旨から『作業場の基礎知識』を『修理用知識』と分けて考えてきたが、支援システムを構築する場合、両方の知識がバランスよく獲得されていくべきである。『作業場の基礎知識』に関して支援能力10、『修理用知識』に関して支援能力100を持つ故障修理支援システムと、『作業場の基礎知識』に関して支援能力50、『修理用知識』に関して支援能力50を持つ故障修理支援システムを比べた場合、（誰を対象に支援するのかにもよるが）後者の方が使いやすいはずである。『修理用知識』の獲得のためには専門家の協力が必要であるが、その際、必ず『作業場の基礎知識』も副産物として得られるはずであるから、それを適切に使うべきである。

謝辞

最後になりますが、本研究にあたり、熱心に御指導いただいた岡本孝太郎教授、また細部にわたり有用な助言をしていただき、さらに叱咤激励していただいた桑田正行助教授、そして共同研究者として協力していただいた森田敬之氏、森島毅氏をはじめとした桑田研究室の皆様へ深く感謝し、御礼を申し上げたいと思います。

また、有意義な実習の場を与えて下さいました多摩オーディオサービス(株)の岡本所長にも感謝の意を表します。

参考文献

- [1] 上野晴樹：「知識工学入門」，オーム社(1985)。
- [2] 和多田作一郎：「人工知能のABC」，購談社 (1988)。
- [3] 岩瀬浩治：「電子回路実験のための知的学習支援システムの開発」，昭和60年度通信工学専攻修士論文(1986)。
- [4] トランジスタ技術編集部編：「電子回路部品活用ハンドブック」，CQ出版社(1985)。
- [5] 淵一博監修 古川康一，溝口文雄共編：「メンタルモデルと知識表現」，共立出版 (1986)。
- [6] 森田敬之，桑田正行：「Smalltalk-80による知識ベースシステム構築用ツールの実現」，電子情報通信学会技術研究報告，教育工学，ET89-96(1989)。
- [7] 森田敬之：「診断型知識ベースシステム構築ツール」，平成元年度電子工学科修士論文 (1990)。
- [8] 竹内郁雄 監修，梅村恭次 著：「Smalltalk-80 入門」，サイエンス社(1986)。

この他、本研究および論文全体を通して参照した文献を以下に示す。

- [9] Tim O'Shea, John Self, 坂元昂 監訳：「人工知能による学習革命」，ホルト・サウンダース・ジャパン (1984)。
- [10] 佐伯胖：「コンピュータと教育」，岩波新書 (1986)。
- [11] Donald L. Schilling, Charles Belove, 山中 惣之助 他訳：「電子回路」(I)，(II) 改訂第二版，マグロウヒル (1982)。
- [12] 真利藤雄監修，林謙二編著：「CDプレーヤ入門」，コロナ社 (1983)。
- [13] 上野晴樹，石塚満 共編：「知識の表現と利用」，オーム社(1987)。
- [14] 長尾真：「知識と推論」，岩波書店(1988)。