

修 士 論 文

題 目 人工現実感技術を用いた
機器保守訓練環境構築支援
システムの開発

指 導 教 官 吉川 榮和 教授

京都大学大学院工学研究科 電気工学専攻

氏 名 石井 裕剛

平成 10 年 2 月 10 日

目次

第 1 章 序論	1
第 2 章 研究の背景と目的	3
2.1 人工現実感技術の展望と本研究室における研究の経緯	3
2.2 人工現実感技術を用いた訓練環境を構築する際の問題点	4
2.3 本研究の目的と方法	6
第 3 章 人工現実感技術による訓練環境の構築支援システムの機能検討	8
3.1 訓練環境のインタフェースと仮想空間のあり方について	8
3.2 支援システムを用いて訓練環境を構築する際に作成すべき情報	10
3.3 構築支援システムが具備すべき機能	11
第 4 章 ペトリネットによる仮想空間内作業のモデル化	13
4.1 ペトリネットについて	13
4.2 モデル化の対象とする機器分解組立作業	13
4.2.1 機器分解組立作業の特徴	14
4.2.2 訓練生のジェスチャの分類	15
4.3 ペトリネットによる仮想空間内の作業のモデル化	16
4.3.1 本研究で用いるペトリネット	16
4.3.2 ペトリネットによる仮想空間内の作業のモデル化	19
4.3.3 実際のモデル化の例	22
4.3.4 ペトリネットと仮想空間の関係	22
4.4 構築したペトリネットの自動実演機能への応用	23
4.5 カラーペトリネットを用いたモデル化の方法の検討	24
第 5 章 訓練環境構築支援システムの構築	26
5.1 構築支援システムのソフトウェアとハードウェア構成	26
5.1.1 訓練に必要な情報を入力する場合のシステム構成	26
5.1.2 訓練を実行する場合のシステム構成	27
5.2 仮想空間内で模擬する必要がある物体の運動と物体間の拘束	27
5.3 支援システムで入力可能なパラメータ	28

5.4	訓練環境構築支援システムのインタフェース	31
5.5	構築支援システムでの訓練実行時の情報の流れ	33
5.6	構築支援システムを用いた訓練環境の構築例とその実験による評価	34
5.6.1	支援システムの使用例	34
5.6.2	訓練環境構築支援システムの操作性の実験による評価 . . .	35
5.6.3	支援システムで構築できる訓練環境の実験による評価 . . .	36
第 6 章 結論		40
謝 辞		41
参 考 文 献		42
付録 A ハードウェア詳細		92
A.1	Magell an	92
A.2	Crystal Eyes	92
付録 B ペトリネットの自動実演機能への適用		94
B.1	発火系列を検索する方針	94
B.2	発火系列の検索法	95
B.3	ペトリネットの簡素化と発火系列の検索の例	98
付録 C 支援システムにおける 3 次元座標の記述法		106

第 1 章 序論

近年急速に発展し、将来のヒューマンインタフェースとして注目されている技術に人工現実感 (Virtual Reality: VR) 技術がある。VR 技術は、高速なコンピュータにより人工的に仮想的な映像空間(以下「仮想空間」と呼ぶ)を 3 次元グラフィックス等を用いて構築すると共に、3 次元マウス等のデバイスを用いて、その仮想空間での人間の動作を、より現実世界の行動に近い状態で入力することによって、仮想空間と人間との相互作用を可能とするものである。すなわち、VR 技術はより自然に近い仮想的な環境をコンピュータによって人間に提供するためのインタフェース技術である。

VR 技術の発展により、最近、仮想空間内に訓練環境を構築する研究が盛んに行われるようになってきた。Roftinらは、宇宙空間でハッブル宇宙望遠鏡の修理作業を行うことができる仮想環境を構築し、実際の訓練に用いたが、その後のアンケート調査によって、そのシステムの有効性を確認している¹⁾。本研究室でも既に、仮想空間内で原子力プラントの構成機器であるスイング式逆止弁の分解訓練を行うことができる訓練システムを構築し、その結果、初心者が機器の構造や、各部名称、訓練の手順等を理解するのに有効であるとの結論を得ている²⁾³⁾。

しかし、これまで本研究室の研究では、訓練システムを構築する際のプログラミングの作業量は考慮に入れておらず、特定の訓練システムを構築するのに、個々にプログラミングを行っていた。VR 技術を用いた訓練システムの構築には、高度なプログラミングの知識が前提となり、実際の訓練を行うために必要となる様々な機器に対する訓練システムを構築する場合、プログラミングの作業量が、大きな問題となっていた。

そこで本研究では、VR 技術を用いた訓練環境を構築する作業の効率化を図るため、プログラミングを行わずに、訓練に必要な情報を入力し、編集するだけで多様な訓練環境を柔軟に構築できる支援システムの開発を行うことを研究目的とした。そして、その目的を達成するため、具体的には、

1. 仮想空間内の作業をペトリネットを用いて効率的にモデル化する手法を提案し、そのモデルを支援システムへ効果的に実装すること、
2. 仮想空間内の物体に対する様々な情報をグラフィカルユーザインタフェース (Graphical User Interface: GUI)を用いて入力し、編集するだけで、目的とする

訓練環境が柔軟に構築できる手法を検討し、支援システムへ効果的に実装すること、

の2つの課題に分けて、本支援システムの設計と開発を進めることにした。

本論文の構成は以下の通りである。

第2章では本研究の背景と目的について述べ、第3章ではVR技術による訓練環境の構築支援システムの機能検討を行った結果を述べる。そして、第4章では、訓練環境の効率的な構築のための手法として、仮想空間内に表現すべき様々な「物体」の状態管理にペトリネットを用いる方法について述べ、第5章では、本研究で構築した訓練環境の構築支援システムの全体構成と、システムの評価を行った結果について述べる。最後に第6章において本研究を省みて、結論と今後の課題を述べる。

第 2 章 研究の背景と目的

本研究では、VR 技術を用いた訓練環境を構築する際の作業効率化を図るための、訓練環境の構築支援システムの開発を目指している。本章では、まずはじめに、VR 技術を用いた訓練環境の特徴について述べ、次にその構築を行う際の問題点と解決策、そして本研究の目的について述べる。

2.1 人工現実感技術の展望と本研究室における研究の経緯

近年、人工現実感という概念が生まれ、その技術が急速に実用化され始めている。人工現実感を用いれば、まだできていない製品を手で触ったり、設計中の家の中を歩き回ったり、未知の分子構造を手で組み合わせてみたりするといった今まで考えられなかった新しい世界が拓けると期待されている⁴⁾。VR 技術を用いた訓練環境もその応用の一つであり、その代表的なものとして NASA (米国航空宇宙局) での Hubble 望遠鏡の修理訓練への適用¹⁾が挙げられる。VR 技術を用いた訓練環境は実機を用いた訓練環境に比べて、以下の様な利点が挙げられる。

1. 大規模な機器に対する訓練環境を、限られた広さの場所に構築することができる。
2. 実物機器を用いないので安全である。
3. 動画や音声等のマルチメディアを訓練に用いることができ、実機のみを用いる訓練に比べて効果的な訓練が期待できる。
4. 実機では模擬することが困難な、故障等の状態にある機器を用いて、何度も繰り返し訓練を行うことができる。
5. 訓練生の訓練履歴を自動的に記録する機能等を付加することができ、インストラクタの負担を軽減させることができる。

本研究室の先行研究により、VR 技術を用いた訓練環境は、作業の手順や機器の構造及び各部名称を覚えるのには効果的であることを確認したが、一方、その研究によって、訓練環境の構築には計算機に関する専門的な知識と、多くの労力が必要

であることも判明した。先行研究では、部品の総数が約 20 個、最短の作業手順数が約 100 ステップ程の訓練環境を完成させるのに、C 言語のプログラミングに精通した者 2 人で、約 4 カ月の日数を要した。（機器の構造に関するデータを除いたプログラムサイズは約 8000 行）。この作業では、仮想空間内での作業を可能にするためのアルゴリズム考察の試行錯誤による時間も含まれている。従って、実際に訓練を行う際に必要とされる、数多くの機器に対して訓練環境を構築することは、その作業量の問題から、現状では非常に困難である。

2.2 人工現実感技術を用いた訓練環境を構築する際の課題点

実際に VR 技術を用いた訓練環境を構築する場合、大きく分けて以下の 4 つの作業を行う必要がある。

I 訓練の対象となる機器の 3 次元形状データやテクスチャ*の作成

これらは、訓練中に訓練生に 3 次元画像として提示するためのデータであり、これらのデータをもとに仮想空間内での物体同士の接触の判定や、分離の判定も行う。一般に物体の形状データは CAD (Computer Aided Design) と呼ばれるモデリングツールを用いて作成することができるが、最近では CT（計算機断層撮影）を用いて人体の形状を計測し、それを 3 次元形状データとして利用する方法や⁵⁾、カメラ画像に基づいて物体の表面の 3 次元位置を計算し、それを用いて 3 次元形状データを作成する技術も開発されている⁶⁾。

II 訓練生が訓練を行うためのユーザインタフェースの構築

具体的には、インタフェースの中で、訓練生は仮想的な訓練を行うことになる。ユーザインタフェースを構成するには、まず、訓練生の手の動きを計測し、ジェスチャの判定の処理を行うためのプログラムの作成と、仮想空間内の物体の位置、姿勢、3 次元形状等のデータをもとにシミュレーション計算を行い、訓練生に提示するための 3 次元動画像を生成する、仮想空間内シミュレータのためのプログラムを作成することが必要である。

訓練生が仮想空間内の物体を操作し、実空間と同様の機器保守作業を行うことを可能にするためには、各物体相互間の干渉をシミュレーションする必要がある。例

* 仮想空間内の 3 次元物体の表面に張り付ける 2 次元の画像のこと。テクスチャを張り付けることをテクスチャマッピングと言う。テクスチャを用いると、計算機に比較的負担をかけずに、臨場感のある映像を作り出すことができる。

例えば、手でペンを握っているのであれば、ペンは手の動きに追従して動く必要がある。また、蓋がナットとボルトで固定されているのであれば、ナットが外れるまでは蓋が外れないようにせねばならない。しかし、物体の3次元形状データを元に物体間の完全な接触判定を行い、物体の動きを制限することは、形状データの精度や計算機の処理速度等の問題により、現状では不可能に近い。従って、多くの場合、物体に「状態」の概念を導入し、各物体の取り得る状態に対して、物体同士の干渉を模擬するための計算機上での処理を予め定めておくことによって、訓練生のジェスチャに応じた物体の動きを再現する方法が採用される。しかし、この方法を採用する場合、上記Ⅰ、Ⅱの作業に加えて、以下の2種類の作業も行う必要がある。

Ⅲ 仮想空間内の物体の状態遷移の仕方に関する情報の作成

仮想空間内での作業を可能とするためには、訓練生のジェスチャに応じて、仮想空間内の物体の状態がどのように変化するかについての情報も必要となる。例えば、ペンが「机の上に置かれた状態」で、訓練生が「右手でペンを握る」ジェスチャを行った場合、ペンの状態は、「右手に握られた状態」に遷移する必要がある。このような遷移の仕方を決定するための情報の作成が必要である。

Ⅳ 訓練対象の機器に物理法則を適用するための情報の作成

Ⅲの作業によって作成された情報をもとに、物体の状態が決定すると、今度は、その状態に応じた物体の物理的な振舞動作を仮想空間内でリアルに再現しなければならない。例えば、ペンが「空中で手を放され、机に向かって落ちつつある状態」にあれば、ペンが机と接触するまで、自由落下させる必要がある。またナットが「ボルトにはまった状態」にあれば、ナットの運動はボルトを中心軸とした回転運動に限定する必要がある。このような物体相互間の束縛や、運動を支配するための情報の作成とその動作振舞のシミュレーションを実行するプログラムの作成が必要である。

もし、複数の機器を対象とした訓練環境を構築しようとするには、以上の4種の作業の内、各機器に共通して使用できるものはⅡだけであり、その他の部分は訓練対象の機器の種類と数に応じていちいち作り直さねばならない。Ⅰの作業に関しては市販の適切な作成支援用アプリケーションを用いれば可能であるが、ⅢとⅣの作業に関しては、その作業内容が非常に複雑で労力を必要とするにも関わらず、効果的な作成支援用ツールはまだ存在していなかった。そのため、これらの部分の構築作業が訓練環境を構築する際の大きな負担となっていた。

2.3 本研究の目的と方法

そこで本研究では 2.2 節で述べた 4 つの作業の内, III と IV の作業を効率的に行うための手法を考案し, それを基に訓練環境の構築から訓練の実施までを行うことができる支援システムの開発を行うことを研究目的とすることにした。

以下には,

- (1) 仮想空間内の物体の状態遷移の記述方法と関連情報の整理,
- (2) 訓練対象の機器等の動作振舞の記述方法と関連情報の整理,

の 2 つについて, 本研究で試みる方法の要点を述べる。

まず, (1) に関して, 仮想空間内の物体の状態変化を離散事象過程として捉え, ペトリネットによるモデル化を試みる。離散事象の状態遷移をグラフによって表記するペトリネットモデルを用いることにより, 物体の状態変化の仕方を視覚的に表現できるだけでなく, 構築したペトリネットを数学的に扱うことにより, 作業手順の検索等を行うことができ, 訓練環境に新たな機能を付加することが可能になる。機械の組立工程の表現にペトリネットを用いた研究は多く見られるが⁷⁾, 仮想空間における物体の物理的な制約を含めた作業手順の表現にペトリネットを用いた例は見あたらない。

ついで, (2) に関して, 本研究では, 訓練の実行上必要な機器の動作振舞に限定した関連情報を入力データとして与えるだけで, 煩雑なプログラミングを行うことなく訓練環境を構築できるシステムの構築を目指す。これにより, 訓練環境の構築上の労力を軽減させることができるだけでなく, 専門的な知識を持たない人でも訓練環境の修正が可能となる。従って, 訓練を実施する現場での, 訓練環境の変更等が可能になり, より効果的な訓練環境の構築が期待できる。

GUI ベースで仮想空間を構築できるシステムを開発した研究は他にもあるが⁸⁾, それらは 3 次元アニメーションの提示が対象であり, 実際にユーザーが 3 次元マウス等の入力デバイスを用いて, 仮想空間内の物体を操作できる環境を構築できるシステムはまだ構築されていない。

以上の方針のもとに, 具体的には,

1. 仮想空間内に訓練環境を構築するために必要となる情報の抽出と分類, と
2. 多くの情報を効率的に入力するための手法の検討

を行い, 以下の特徴を有する支援システムの設計と開発を行うこととした。

1. 各物体の物理法則を定める情報を GUI を介して入力することにより, プログラミング作業を行わずに, 容易に訓練環境が構築できる。

2. 仮想空間内の各物体の状態遷移を, ペトリネットモデルで表現し, そのグラフィ的な表現力を利用して, 視覚的に各物体の状態遷移の仕方を設計できる.
3. 構築した訓練環境を容易に修正, 再利用することができる.
4. 入力した情報をもとに, 実際に訓練を行うことができる.

また本研究では, 仮想空間内の物体の状態遷移をモデル化したペトリネットを解析することによって, 訓練手順の自動検索による手順教示機能等の, 訓練を支援する機能の構築の可能性も検討する. 但し, その詳細は多岐に渡るため, 付録 B で述べることにする.

第 3 章 人工現実感技術による訓練環境の構築 支援システムの機能検討

本章ではまず,3.1節において VR 技術を用いた訓練環境が具備すべき要件について述べる. 次に 3.2 節において, 支援システムを用いて訓練環境を構築する際に入力すべき情報についてまとめ,3.3 節において, 支援システムが具備すべき機能について述べる. なお, これらの支援システムへの具体的な実装方法については第 5 章において述べる.

3.1 訓練環境のインタフェースと仮想空間のあり方について

訓練環境の構築支援システムの設計を行うためには, まずそのシステムで構築可能な訓練環境を決定する必要がある. 本研究では, 本研究室での先行研究³⁾の経緯をもとに, 訓練環境には以下の要件を具備させることにした.

1. 仮想訓練環境に対する予備知識がない訓練生にでも容易に利用できること
訓練環境を使用するために新たな知識を必要としないことは, 訓練生を作業の訓練に専念させるために重要である. そのためには, 訓練環境は「現実の作業に似たジェスチャで機器を操作できるインタフェース」を具備する必要がある. これには, 訓練生の自然なジェスチャをシステムが計測する必要があるが, その方法には, データグローブを用いる方法や, 3 次元マウスを用いる方法, 画像処理による位置・形状計測の方法⁹⁾等がある. 本研究では, ジェスチャの計測を手に限定することとし, 計測の精度が高いこと, 作業の場所が比較的小さくて済むこと, 計測装置自体が安価であること等を考慮し, Logitech 社の 3 次元マウス (Megellan) を用いることとした¹⁰⁾ (付録 A 参照).
2. 視点の自由な移動が可能であること
実世界の訓練では, 機器を真上から覗いたり, 機器を周りから見回したりする場合が多い. そのため違和感のない訓練を実現するには, 訓練環境において, 訓練生が自由に視点を移動させることができる機能が必要である. 本研究に

においては、3次元マウスを用いて視点を移動させ、どの視点からでも訓練を行うことが可能なものとするにしました。

3. 訓練環境の臨場感を向上させること

人工現実感技術を用いた訓練環境の場合、訓練生を訓練に没入させるためには、仮想空間の臨場感が高いことも必要となる。本研究では、仮想空間内の物体にテクスチャを貼ることにより、視覚的な臨場感を向上させると共に、物体同士の衝突や、物体同士のこすり合わせ等の時には、効果音を出すことによって、聴覚的な臨場感の向上も図ることができるようにした。

4. 仮想空間内での作業を行いやすくすること

人工現実感技術を用いた訓練環境は、実空間では用いない特殊なデバイスを用いて訓練を実行するため、訓練生のなめらかな操作を可能にするためには、何らかの援助機能が必要である。そこで本研究では、(1)奥行き知覚特性の改善、(2)力覚フィードバックの欠如の補完、の2点に注目し、以下の2点の工夫を行うこととした。なおこれらの工夫の効果は、実験によってその有効性を確認しているが、その実験の詳細は第5章で述べることにする。

(1) 訓練生の仮想空間内における奥行き知覚を改善するための工夫

機器分解組立の作業は一般に3次元空間における機器の操作であるため、訓練生は常に訓練対象機器の各部品の前後関係を知覚する必要がある。訓練生に仮想空間内の物体の前後関係を知覚させるには、CrystalEyes(付録A参照)を用いて立体映像を提示する方法や、仮想空間内の物体に影を付ける方法、訓練生が把持している物体に図31のように、棒状の定規を付加する方法等がある。GystalEyesを用いた3次元画像提示は、比較的容易に実現することができ、実空間と同様の感覚で物体の前後関係を知覚することができる利点があるが、GystalEyesは訓練生の眼に生理的な負担がかかり、長時間の使用には向いていないことや、立体視ができない人もいるという問題点もある。また、棒状の定規を付加する方法は、訓練の臨場感を低下させる可能性がある。そこで本研究では、図32に示すように物体に単純な影を付ける方法を採用し、GystalEyesは必要に応じて使用する、しない、を選択できるようにした。

(2) 力覚フィードバックの欠如の補完

本支援システムで構築可能な訓練環境では、力覚フィードバックを考慮していないので、訓練生の作業を支援するために、何らかの工夫が必要である。本研究では、訓練中に訓練生が何らかのイベントを起こした場合(例

例えば物体を触る, 物体を填める等), 先に述べた効果音に加えて, 必ずイベント音を提示することとした。力覚フィードバックが無くてもこのような補助的な工夫によって手で物を触った等の知覚を十分補完できると期待される。

5. 訓練を行う際の援助機能

効果的な訓練を行うためには, 仮想空間内で単に保修作業ができるだけでなく, 訓練を援助する様々な機能も必要である。本研究では, 機器保修作業に関して初心者である訓練生が, 訓練の作業手順を効果的に学習できるようにするために, 訓練環境に, 作業の自動実演機能を付加する方策を検討する。詳細な内容は付録 B で述べることとする。

本研究では, 上記 5 つの項目を考慮した訓練環境を設計し構築するのに用いるべき支援システムを開発する。具体的には図 3.3 に示すようなシステム構成の支援システムを考えている。すなわち, 本システムを使用すると, 複数の訓練生へ異なった複数の訓練環境も作成することが可能である。本研究では全ての機能が完成に至っていないが, 本システムが完成すると, 次のようなステップで実際の訓練に使用することができる。

- (1) まず, 訓練環境の構築者(訓練のインストラクタを想定)は, 本支援システムを用いて, 複数の訓練生用の訓練環境をその GUI を介して構築する。
- (2) それぞれの訓練生には, それぞれの訓練環境用のデータが提供されるので, それぞれの仮想空間内で個別に訓練を行う。
- (3) それぞれの訓練生の訓練の進行状況について, その訓練状況のデータが自動的に記録され, インストラクタに各々の訓練生の履歴データが送信される。
- (4) インストラクタは, 送信された履歴データを解析して, それぞれの訓練生の習得レベルに応じた適切な教示を訓練生にフィードバックする。

なお, 本研究の範囲では上記の内, (3), (4) については将来の課題となっている。

3.2 支援システムを用いて訓練環境を構築する際に作成すべき情報

仮想空間内に訓練環境を構築するためには, 以下の情報を作成する必要がある。

1. 訓練対象機器の3次元形状データとテクスチャ
2. 訓練を行う際に提示する効果音
3. 各物体がとることができる状態(例えば、「床の上にある」、「手に握られている」等)
4. 各物体の状態遷移の仕方(例えば、「手で触られている状態」から「手で握られている状態」へ遷移する等)
5. 状態遷移を起こさせる契機(例えば、「訓練生が手を握る」、「ペンが床に接触する」等)
6. 各状態における各物体の運動と拘束関係の記述(例えば、「手の動きに追随する」、「床まで自由落下する」等)

本システムにおいては、上記の1と2の情報に関しては、市販のアプリケーションを用いて作成したものを、利用することとする。3から5の情報に関しては、物体の状態遷移の仕方をモデル化したペトリネットをシステム上で作成することにより入力し、6の情報に関しては、物体の動きについて予め設定したメニューの中から該当する項目を選択し、ついで選択した動きに該当する必要な数値を入力することによって設定する。なお、本研究で用いるペトリネットモデルに関しては、第4章で述べることとし、物体の動きに関する情報の内容は第5章において述べることにする。

3.3 構築支援システムが具備すべき機能

これまで検討したシステムの機能の他に、支援システムを、訓練環境を構築するアプリケーションとして捉えた場合、以下の機能を備える必要がある。

- 構築した訓練環境の修正機能

支援システムにより訓練環境を効率的に構築することを可能にする為には、訓練に必要な情報をその都度いちいち初めから入力するだけでなく、既に入力した情報をもとに修正、削除するインタフェース機能も必要である。

- 訓練環境データの管理機能

構築した訓練環境のデータを保存したり、読み込んだりする機能。この機能を利用することにより、読み込むファイルを変更するだけで、複数の訓練を同じ

ハードウェアを用いて実行することが可能となる。また、訓練環境の複製が可能になり、少しずつ変更した訓練環境を容易に構築することが可能になる。

- 支援システムの環境設定機能

支援システムを利用する計算機の性能により、仮想空間の描画方法等を変更したい場合が多々ある。例えば、処理能力の高い計算機を利用する場合は、仮想空間内の物体全てに完全なテクスチャを張り、臨場感のある画像を提示し、処理能力が低い計算機を利用する場合は、テクスチャを貼らず、さらにワイヤーフレームで描画するように設定できると便利である。そこで本システムでは、環境設定の画面で項目を選択することにより、訓練の途中でも、仮想空間の描画方法を変更することを可能にした。

第 4 章 ペトリネットによる仮想空間内作業のモデル化

本章では、人工現実感技術を用いた訓練環境である仮想空間を、事象駆動型の離散事象システムとしてとらえ、その状態変化をペトリネットを用いてモデル化することにより、効率的に各物体間の関係情報を管理する方法について述べる。まず、4.2 節においてモデル化の対象とする機器保修作業について述べ、4.3 節において本研究で用いるペトリネットの定義と実際のモデル化手法について述べる。

4.1 ペトリネットについて

ペトリネットは、プラントのプロセス、コンピュータの回路動作、生産ラインの物の流れ、流通における物流など、システムにおける離散的、並列的な状態遷移のモデル化に用いられる。ペトリネットの特徴は、「グラフィカルな視覚的ツール」、
「シミュレーションツール」、及び「数学的方法論」の三つの機能を同時に備えていることであり、ユーザインタフェースの観点からは強力なコミュニケーション媒体としてもその威力を発する。ペトリネットに関する研究は、ペトリネットの生みの親であるカール・アダム・ペトリ (Carl Adam Petri) 博士が 1962 年に学位論文¹¹⁾を発表して以来、めざましい発展を遂げており、その用途に応じて様々な形で拡張されていて、代表的なものには、「時間ペトリネット¹²⁾」「確率ペトリネット」「カラーペトリネット¹³⁾」等が提案されている。

4.2 モデル化の対象とする機器分解組立作業

本節では、モデル化の対象とする機器分解組立作業の特徴について述べる。本研究では、構築したペトリネットが複雑になることを避けるため、モデル化の対象を、両手を用いた機器分解・点検・組立作業に限定する。この限定は、機器保修作業が、両手以外を用いて行うことが少ないことや、本研究で訓練を実行する環境が、訓練生の手以外を計測しないことから、一般性を損なうことはないと思われる。

4.2.1 機器分解組立作業の特徴

モデル化の対象例として、原子力発電所の保守作業を選び、その保守訓練機器の一つであるスイング式逆止弁の構造を図 4.1 に、保守作業の手順の一部を表 4.1 に示す。この表 4.1 を見ると分かるように、一般の機器保守作業は、スパナやペンチ、ハンマーなど、複数の道具を両手を用いて操作し、ナットやボルト、ピンなどの複数の物体を操作する作業となる。これらの作業は、各物体（両手を含めて）の状態が順次変化していくことにより進行する、と捉えることができる。例えば、保守作業の最初の工程である「ペンで弁箱に合いマーク*を付ける」作業は、手とペンの状態がそれぞれ、図 4.2 に示すように変化することによって完了する。このように、機器保守作業を物体の状態遷移として捉えると、以下の特徴を見い出す事ができる。

- 物体の状態遷移を起こさせる契機は基本的に訓練生の両手である。
- 物体の状態遷移は同時に複数の物体に対して起こりうる。
- 状態遷移は、手が直接操作している物体に限らず、間接的に操作している物体にも生じる。
- 訓練生の両手の状態は必ず作業の初期状態である「何もしていない」状態に戻ることができる。
- 機器保守作業には、点検作業等、必ずしも物体の状態が変化しない作業も存在する。
- 物体の落下等、必ずしも訓練生の両手によって引き起こされない状態遷移も存在する。
- ある物体の状態が、他の物体の状態変化を妨げることが多々ある。例えば、ナットがボルトにはまっている状態にあれば、それによって固定されている他の物体は、ナットが外れた状態になるまで、固定されたままである。

また、機器保守作業の特徴として、「作業の正しい手順」が存在するが、それは、必ずしも、物体の状態変化と一対一に対応するわけではない。例えば、4 組のボルトとナットで弁箱の蓋が固定されている場合、蓋を外す作業は、最初にどのナットを緩めても正しい作業となる。さらに、訓練生が訓練中に行う作業は、訓練用手順書に書かれている作業とは全く関係のない作業も行う可能性があることに留意せねばならない。

* 分解後、再び組み立てる際に部品同士の向きを分解前と一致させる為の目印として付けるマーキング

4.2.2 訓練生のジェスチャの分類

スイング式逆止弁の保修作業を図 4.2 に示す状態変化として捉えた場合、その状態変化を起こさせる契機となる事象は図 4.2 の左側に示す 5 つの訓練生の行動と、1 つの物体の運動によるものであることが分かる。これらの事象の内、訓練生の行動によるものに注目した場合、これらの行動は、以下の 6 つのジェスチャの何れかに対応することが分かる。

1. 2 つの物体を接触させるジェスチャ

ボルトにナットをはめる場合（ボルトとナットが接触する）や、手でペンを触る場合（手とペンが接触する）等がこれに相当する。

2. 2 つの物体を分離させるジェスチャ

ボルトにはまったナットをはずす場合（ボルトとナットが分離する）や、手で触っているペンを離す場合（手とペンが分離する）等がこれに相当する。

3. 手で把持するジェスチャ

ペンを握る場合、ただ単に手で握るだけの場合等がこれに相当する。

4. 手を開放するジェスチャ

ペンを握った手を放す場合、ただ単に手を開くだけの場合等がこれに相当する。

5. 物体を回転させるジェスチャ

ボルトにはまったナットを回転させる等が場合がこれに相当する。

6. 物体を平行移動させるジェスチャ

引き出しを引っ張る場合等がこれに相当する。

逆に、これらの基本的なジェスチャを組み合わせることによって、訓練生の複雑な作業を表現することができる。例えば、「ペンでふたに印を付ける」作業を上記のジェスチャで表現する場合、表 4.2 の 5 つのジェスチャを順番に行うことによって作業が完了する。

ここで留意すべきことは、作業の捉え方によっては、ジェスチャの表現方法が異なってくる場合があることである。例えば、表 4.2 のジェスチャによる表現の 3 つ目の行動は、ジェスチャ 1 の一つだけで表現しているが、ペンでふたに印を付ける作業は「ペンとふたを接触させる（ジェスチャ 1）」と「ペンをふたと接触させたま

ま平行移動させる (ジェスチャ 6)」の 2 つのジェスチャによって実現されることが考えられる。このように、作業のジェスチャによる分解は、幾通りかの方法があるが、これは作業の目的によって使い分けるべきである。例えば、ふたにペンで印を付ける作業が、ただ単に印を付けることだけに注目した作業であれば、これはジェスチャ 1 の一つだけで表現すべきであり、作業が縦方向に印を付けることに注目した作業であれば、これは、ジェスチャ 1 とジェスチャ 6 の両方で表現すべきである。

4.3 ペトリネットによる仮想空間内の作業のモデル化

本節では、本研究で用いるペトリネットの定義を行った後に、実際にペトリネットを用いて、仮想空間内の物体の状態遷移をモデル化する方法について述べる。

4.3.1 本研究で用いるペトリネット

本項ではペトリネットの諸定義を行う¹⁴⁾。図 4.3 にグラフでの表記法の一覧を示す。以下の定義では、カラートークンの定義も行っているが、実際にカラートークンを利用するのは、仮想空間内の物体を制御するときだけであり、モデル化を行う時には、カラーの存在を意識する必要はない。このことは、支援システムにおいて、ペトリネットを構築するときに、高度なカラーペトリネットの知識は必要ないことを意味する。

定義 1 HLP-net

HLP_{net} は 9 項組 $\text{HLPN} = (P, T, F, \Sigma; C, W_1, W_2, K, M)$ として定義される。ただし、

- $P = \{p_1, p_2, \dots, p_m\}, m = |P| > 0$ はプレースの有限集合
- $T = \{t_1, t_2, \dots, t_n\}, n = |T| > 0$ かつ $P \cap T = \emptyset$ はトランジションの有限集合
- $F \subseteq (P \times T) \cup (T \times P)$ はアークの集合
- $\Sigma = \{c_1, c_2, \dots, c_l\}, l = |\Sigma| > 0$ はカラーの有限集合
- $C : P \rightarrow \Sigma$ はプレースのカラー関数。 $C(p)$ はプレース p に入ることのできるトークンのカラーの集合
- $W_1 : F \rightarrow \{0, 1\}$ はノーマルアークの重み関数。 $W_1(p, t)$ は p から t へのノーマルアークの重み、 $W_1(t, p)$ は t から p へのノーマルアークの重み。

- $W_2 : F \rightarrow \{0, 1\}$ は抑止アークの重み関数. $W_2(p, t)$ は p から t への抑止アークの重み. 本モデルでは $W_2(t, p)$ は定義しない.
- $K : P \rightarrow \{1, \infty\}$ はプレースの容量関数. $K(p)$ はプレース p の容量
- $M : P \rightarrow \Sigma$ はマーキング, $M(p)$ はプレース p に割り当てられるマーキングのカラーの集合

ただし $\emptyset$ は空集合 $|X|$ は集合 X の要素の数とする. また, t の入力プレースの集合を $\cdot t$, 出力プレースの集合を $t \cdot$ で表す. 同様に p の入力トランジションの集合を $\cdot p$, 出力トランジションの集合を $p \cdot$ で表す. また, $W_1(p, t)$ と $W_2(p, t)$ は同時に 1 にはならないことにする.

定義 2 シングルプレース (single place)

$\exists p_i \in P$ が $K(p_i) = 1$ を満たすとき, p_i を特にシングルプレースと呼び, グラフでは図 4.3 に示すように, 一重丸で表す.

定義 3 プールプレース (pool place)

$\exists p_i \in P$ が $K(p_i) = \infty$ を満たすとき, p_i を特にプールプレースと呼び, グラフでは図 4.3 に示すように, 一重楕円で表す.

定義 4 制御プレース (controlled place)

定義 2, 3 で述べた両プレースの内, 4.3.2 項で述べる目的で作られたプレースを特に制御プレースと呼び, 他のプレースと区別する. グラフでは, 図 4.3 に示すように二重丸, もしくは二重楕円で表す.

定義 5 ノーマルトランジション (normal transition)

トランジションの内, 定義 1.0 の発火条件が成立しても, イベントが起こらなければ, 発火しないトランジションを定義 6 の自動発火トランジションと区別して, ノーマルトランジション (もしくは単にトランジション) と呼び, グラフでは図 4.3 に示すように一本の縦棒で表現する.

定義 6 自動発火トランジション (auto transition)

トランジションの内, 定義 1.0 の発火条件が成立すると, イベントの発生に関係なく, 必ず発火するトランジションを特に自動発火トランジションと呼び, グラフでは図 4.3 に示すように, 二本の縦棒で表現する.

定義 7 ノーマルアーク (normal arc)

$\exists p_i \in P$ と $\exists t_j \in T$ に対して $W_1(p_i, t_j) = 1$ を満たすとき, p_i を t_j の入力プレースと呼び, グラフでは, 図 4.4 に示すように, プレース p_i からトランジション t_j へ矢印を引くことにより表現する. この矢印を次の抑止アークと特に区別して, ノーマルアークと呼ぶ. 同様に, $W_1(t_j, p_i) = 1$ を満たすとき, p_i を t_j の出力プレースと呼び, グラフでは, トランジション t_j からプレース p_i へ矢印を引くことにより表現する.

定義 8 抑止アーク (inhibitor arc)

$\exists p_i \in P$ と $\exists t_j \in T$ に対して $W_2(p_i, t_j) = 1$ を満たすとき, p_i を t_j の抑止入力プレースと呼び, グラフでは, 図 4.5 に示すように, プレース p_i からトランジション t_j へ矢印の代わりに小円でトランジションにおいて終端する直線 (円矢印と呼ぶ) を引くことにより表現する. この円矢印を抑止アークと呼ぶ.

定義 9 参照アーク (reference arc)

$\exists p_i \in P$ が, $W_1(p_i, t_j) = 1$ かつ $W_1(t_j, p_i) = 1$ を満たすとき, プレース p_i はトランジション $t_j \in T$ の参照プレースという. グラフでは, $W_1(p_i, t_j) = 1$ と $W_1(t_j, p_i) = 1$ が同時に成り立つことを, 図 4.6 に示すように, 双方向の矢印で結ぶことにより表現する.

定義 10 発火可能 (enabling)

トランジション t は以下の条件を全て満たすとき発火可能であるという.

$\forall p \in \cdot t$ となる p に対して,

- 1 $W_2(p, t) = 1$ のとき, $|M(p)| = 0$
- 2 $W_1(p, t) = 1$ のとき, $|M(p)| > 0$

$\forall p \in t \cdot$ となる p に対して,

- 3 $|M(p)| \neq 0$ ならば $W_1(p, t) = W_1(t, p) = 1$, もしくは $K(p) = \infty$ である

トランジション t の発火によりマーキング $M(p)$ は新しいマーキング $M'(p)$ に推移する. ただし, $\setminus$ は差集合を表す.

$$M'(p) = \begin{cases} M(p) \setminus m & (p \in \cdot t, p \notin t \cdot, W_2(p, t) = 0, m \in C(p)) \\ M(p) \cup g & (p \in t \cdot, p \notin \cdot t, g \in C(p)) \\ M(p) & (p \in \cdot t \cap t \cdot) \end{cases} \quad (4.1)$$

4.3.2 ペトリネットによる仮想空間内の作業のモデル化

仮想空間内における物体の状態遷移のモデル化は、物体の取りうる状態をPlacesに、状態遷移をトランジションの発火に対応させることにより行う。具体的には以下の手順でモデル化を行う。

Step1 訓練実行中に状態が変化する物体を抽出する

ここで、状態が変化する物体とは、計算機上での処理が変化する物体のことを意味する。そのため、訓練用手順書に記述されている物体でも、計算機内での処理が訓練中に何ら変化しなければ、ここで抽出する必要はない。例えば、「机の上にあるペンを右手で握って床の上に落とす」作業の場合、抽出すべき物体は、「ペン」と「右手」だけである。（机と床は作業を通して処理を変化させる必要がない）

Seq2 各物体が取りうる状態を抽出する

「机の上にあるペンを右手で握って床の上に落とす」作業の場合、「ペン」の状態は「机の上にある状態」から「床の上にある」状態まで変化するが、その間の「ペン」が取りうる状態を考える。この場合は、「右手に触られた状態」、「右手に握られた状態」、「床に向かって落ちつつある状態」を抽出することができる。ここで、「机の上にある状態」と「右手に触られた状態」とは、ともに右手の動きに関係なく、常に静止しているという点では、計算機上での処理は同じであるが、訓練生が手を握るジェスチャを行った場合、前状態ではペンの状態は変化しないが、後状態では、ペンの状態が「手に握られた状態」に変化するため、この2つの状態は別の状態として考える必要がある。

Seq3 各物体が状態遷移する原因となるイベントを抽出する

「机の上にあるペンを右手で握って床の上に落とす」作業の場合、「右手でペンを触る」、「右手を握る」、「右手を放す」、「ペンと床が接触する」が状態遷移を起こさせる原因となるイベントとなる。

Seq4 ペトリネットを構築する

Seq2 で抽出した状態をPlacesに対応させ、Seq3 で抽出したイベントをトランジションに対応させる。そして、イベントが起こる前の物体の状態を入力Placesに、後の状態を出力Placesに対応させることによって、ペトリネットを構築することができる。

上記のように構築したペトリネットでは、物体がある状態にあることを、その状態に対応するPlacesにトークンが存在することによって表現し、物体の状態遷移

は、トランジションの発火によるトークンの遷移によって表現する。これは、トークン 1 つが、仮想空間内の物体 1 つに対応することを意味する。

Step5 訓練の初期状態を設定する

ペトリネットの初期トークン配置を設定することにより、訓練の初期状態を設定することができる。

また、物体の状態遷移を効率的にモデル化するために、以下の 7 種類の定義を導入する。

自動発火トランジションとノーマルトランジション

上記 Seq3 で抽出した、イベントによる物体の状態変化の他に、ある条件が成立した場合に、イベントに関係なく物体の状態を変化させたい場合がある。例えば、複数のボルトにはまっている全てのナットを外した時に、訓練生に警告を出す訓練環境を構築するためには、ボルトにはまっているナットの総数が 0 になるという条件が成立した時に警告を出すように設計する必要がある。そこで、本研究では、イベントの発生に応じて発火するトランジションをノーマルトランジション、イベントに関係なく条件の成立に応じて発火するトランジションを自動発火トランジションと呼んで区別することにする。

抑止アーク

優先順位のあるシステムは、ペトリネットにプレースのゼロテストを行わう能力が無ければ、モデル化することができない¹⁵⁾。例えば、一つの箱の中にナット A とナット B が複数個あり、ナット A を全て取り出して初めてナット B が取り出せる状態は、ゼロテストを行わなければ、モデル化することができない。これに対するペトリネットの拡張は幾つか提案されているが、本研究では、その概念の容易さを考慮し、抑止アークを導入することとした。このアークを用いることにより、可能なモデル化の範囲が広がると同時に、ある物体の状態が、他の物体の状態遷移を抑止する場合（例えば「ナットとボルトで板を壁に固定する状態」）等をモデル化することが容易となる。

参照プレース

機器保修訓練をモデル化する場合、イベントが発生するための前提条件でありながら、イベントが発生しても、それ自身の状態は変化しない場合がある。例えば、

電気機器の操作の訓練を仮想空間内で行う場合、電気機器を操作するためには、電気機器の電源が ON の状態にある必要があるが、電気機器を操作しても、電源の状態は変化することがない。このような訓練操作をモデル化すると、一つのトランジションに対して、同じプレースが、入力と出力の両方にできることとなる。本研究では特に、このプレースを参照プレースと呼んで区別することにする。

プールプレース

仮想空間内の机の上に、同じペンが 2 本置いてある場合、計算機上でこの 2 本に対して行う処理は、全く同じである^{**}。従って 2 本の物体は同じ状態をとることとなり、一つのプレースに同時に 2 つ以上のトークンが配置される場合もありうる。このように同時に 2 つ以上の物体がとりうる状態に対応するプレースをプールプレースと呼んで、一つしかトークンが入らないシングルプレースと区別することにする。

カラートークン

机の上に複数のペンが置いてある場合には、適用する物理法則が同じであり、どのペンも手で握ることができるので、上述のプールプレースの中に、その対応したトークンが複数入ることになる。しかし、その内の一本を訓練生が把持しようとした時は、訓練生が触っているペンを「手に握られている状態」に遷移させ、同じ状態にある他のペンは状態は変化しない必要がある。このように、計算機上では同じ処理を行う物体でも、状態遷移するときには、他の物体と区別される必要がある。そこで、本ペトリネットではトークンにカラーを導入し、同じ性質を持つ物体であってもペトリネット上では区別して取り扱うこととする。

制御プレース

表 4.3 に示すような訓練環境のモデル化を考える。この作業では Step4 により、「印が付いているナットがボルトにはまった状態」と「印が付いていないナットがボルトにはまった状態」、「印が付いていないナットが手で握られている状態」と「印が付いているナットが手で握られている状態」等、「印が付いている状態」と「印が付いていない状態」をそれぞれ作成する必要があることを意味する。しかしこれらの状態は、ナットのテクスチャデータが異なるだけであり、その他の計算機

^{**} ただし、右側にあるペンは右手だけが握ることができ、左側にあるペンは左手だけが握ることができるようにする場合は、計算機上での処理が異なり、それぞれに対してプレースを作成する必要がある。

での処理は同じであり、無駄があることが分かる。そこで本研究では、制御プレースを導入する。つまり、「ペンに印が付いていない」、「ペンに印が付いている」ことを表す制御プレースをそれぞれ準備し、他のプレースは「ボルトにはまった状態」、「手で握られている状態」のように作成する。そして、計算機上での処理では、「ペンに印が付いていない」、「ペンに印が付いている」ことを表すプレースのトークン配置を参照して、テクスチャを選択し、その他の処理は、他のプレースのトークン配置を参照することにより、処理を行う。制御プレースを導入することにより最大で、プレースの数を約半分に減らすことができる。

4.3.3 実際のモデル化の例

前項で述べたモデル化法により、逆止弁の分解作業を紙面上でモデル化した結果を図 4.7～図 4.17 に示す。モデル化した作業は、「ペンで弁箱に印を付ける作業」から「手で蓋を外す」までで、弁蓋を固定するボルトの個数は、簡単化の為に、実際の逆止弁にあるボルトの個数（16 個）を 2 個に減らしている。モデル化はペンやナットの操作は、右手、左手、どちらでも同様に行えるようにした。また、ナットはスパナと手の両方で外すことができるが、一度スパナで緩めなければ、ナットは手で回すことができない。さらに一度外したナットを別のボルトにはめることもできる。逆止弁を分解する際に操作する対象となるオブジェクトの数は、4 種類 8 個で、作業を終えるまでに最短の手順で分解を行った場合、作業工程数は 41 工程である。プレースの総数は、ノーマルプレース 86 個、制御プレース 12 個、プールプレース 2 個、トランジション総数 132 個であった。

4.3.4 ペトリネットと仮想空間の関係

実際に構築したペトリネットを訓練に使用するためには、ペトリネット上でトランジションが発火し、トークンがプレースを移動していくようにペトリネットシミュレータが動作し、ペトリネットシミュレータと訓練作業を行う仮想空間の管理プログラムとの間で、データをやりとりせねばならない。具体的には、図 4.18 のように、訓練の開始時に、ペトリネットシミュレータにペトリネットの構造とトークンの初期配置をデータとして送る。その後、ペトリネットシミュレータは発火可能なトランジションを検索し、その結果を仮想空間管理プログラムに送る。ここで、仮想空間管理プログラムは、発火可能なトランジションのデータから、トランジションの発火を起こさせる訓練生のジェスチャを予め調べておく。このようにすることによって、仮想空間管理プログラムは、イベントが起こるかどうかを、予め調べておいたジェスチャに関係するものについてのみ、監視すればよいことになり、計算機の

負担を減らすことができる。例えば、「手でペンを握る」トランジションのみが発火可能である場合、仮想空間管理プログラムは、訓練生が手を握るジェスチャを行うかどうかのみ監視すればよく、その他のイベント、例えば、「ペンが床と接触するかどうか」や「訓練生が手を開くかどうか」はこの時点では監視する必要がない。

ペトリネットシミュレータと仮想空間管理プログラムとの間でやりとりされるデータは、トークン配置や発火可能なトランジションに関するデータのみであるので、送るべき情報の量としては、それほど多くなならない。従って、これらの情報をネットワークを通して転送することも十分可能であり、ペトリネットシミュレータと仮想空間管理プログラムとを別々の計算機で処理することにより、処理を分散させることも可能である。

4.4 構築したペトリネットの自動実演機能への応用

一般に機器保修作業は、数多くの道具と部品を複雑な手順で操作する作業であるため、初心者訓練生が初めて作業を行う場合には、覚えるべき事柄が非常に多く、何らかの支援が必要である。その対策の1つとして、訓練環境に自動実演機能を構築することが考えられる。自動実演機能によって、これまでの作業の状況に応じた適切な作業手順を、3次元映像として訓練生に提示することが可能になれば、インストラクタが常に付き添わなくても、正しい作業手順が効率的に学習できるものと期待される。

自動実演機能を実現するための方法として、構築したペトリネットを利用することが考えられる。すなわち、自動実演機能が呼ばれた時のペトリネットのトークン配置を調べ、これまでの作業状況を調べて、どこまで自動的に手順を実演すべきかを判断して、目的とすべきトークン配置を決定し、現在のトークン配置から目的のトークン配置に至るまでのトランジションの発火系列を検索し、得られた発火系列を次の手順として実演する。このような方法で自動実演機能を実現しようとした場合、以下のような項目が問題となる。

1. 作業の正しい手順の情報をどのように記述するか
4.2節で述べたように、機器保修の正しい作業手順は、必ずしも一通りではなく、状況によっては同じ優先順位の作業が多数存在する場合がある。このような多様な作業手順の候補を効率的に、正確に記述する方法を考案する必要がある。
2. 巨大なペトリネットの発火系列をどのように検索するか

自動実演機能が利用された時点で、仮想空間内の機器の状態が正しい手順から逸脱していて、1での記述と異なっている場合、各機器の状態を1での記述に適合させる作業手順を検索する必要がある。目的とする各機器の状態が決定している場合、この問題は、ペトリネット上では、現在のトークン配置を、目的のトークン配置に変化させるトランジションの発火系列を検索する問題に相当する。ペトリネットの発火系列の検索法としては、(1)接続行列を用いて行列方程式を解くことにより各トランジションの発火総数を求める方法(2)可達木を用いて全ての場合について調べる方法等が提案されている¹⁵⁾。(1)の方法では抑止アークや自己ループが存在する場合は、そのままでは適用することができない。また、求めることのできる解は、目的のトークン配置に至るまでの各トランジションの発火回数の総数だけであり、どのトランジションがどの順番で発火するのかについての情報は得ることができないという問題がある。一方(2)の方法では、ペトリネットが並列動作可能な対象をモデル化している場合、発火系列の検索に、プレースとトランジションの総数の指数関数に比例した時間と記憶領域が必要であることが分かっており¹⁶⁾巨大なペトリネットに対してそのまま適用するのは、非常に困難であるという問題もある。

3. 発火させるべきトランジションの情報からどのように3次元動画像を構成するか

次に発火させるべきトランジションが決定すると、今度は、その情報をもとに3次元動画像を合成する方法を考案せねばならない。このような動画像を合成するためには、かなりの量のデータベースを作成する必要がある。例えば、「ペンを手で掴む」といった作業を合成する際に必要な情報と、「ナットをスパナで緩める」といった作業を合成する際に必要な情報とでは、用いる情報が全く異なることが予想され、このような情報を効率的に作成する方法を考案する必要がある。

本研究では、特に2の問題に対して検討を行った。その詳細は多岐に渡るため、付録Bにおいて述べることにする。

4.5 カラーペトリネットを用いたモデル化の方法の検討

4.32項で述べたモデル化法を用いてモデル化したペトリネットを見ると、良く似た構造を持ったネットが数多く存在している。例えば、「ペンを握って放す作業」を表すペトリネットと「ナットを握って放す作業」のペトリネットを取り出して見る

と図 4.19 のようになり, ネットの構造としては全く同じであり, 持つ対象が「ペン」と「ナット」と違っているだけである. このようなペトリネットは, 2 つ以上の属性で特徴づけられるカラートークンを用いれば, その構造を単純化することができる. 例えば, 図 4.19 のペトリネットをカラーペトリネットを用いて書き換えると, 図 4.20 のようにネットサイズを半分以下にすることができる. ただし, 図中の表記で, プレース「 y は x を握っている」にトークン〈ナット, 右手〉が存在している時は, 「右手がナットを握っている」ことを表している. またトランジション「 y を x で握る」が発火するためには, プレース「 y は x を触っている」にトークン〈 x, y 〉が存在している必要があり, このトランジションが発火することにより, プレース「 y は x を握っている」にトークン〈 x, y 〉が遷移することになる. このように, 2 つ以上の属性を持つカラートークンを使用するペトリネットを用いることにより, ペトリネットの構造を単純にすることができるが, このペトリネットを構築する際には, アークにラベル付けを行ったり, トランジションの発火規則を詳細に定義するなど, 別の面ではモデルが複雑になる. 従って, 本研究で構築する支援システムでは採用しなかったが, ペトリネットに関して熟知した者が, 支援システムを利用することを前提とする場合は, こちらのペトリネットを採用する方が効率的であると思われる.

以上, 本章では, 仮想空間内での機器の分解・組立に関わる訓練作業のペトリネットによるモデル化について検討した結果を述べた. なお, 次章に述べる訓練環境の構築支援システムの構築では, 本章の 4.4 節および 4.5 節に述べた, ペトリネットの自動実演機能とカラーペトリネットによるモデル化は対象外として, 今後の発展課題としている.

第 5 章 訓練環境構築支援システムの構築

本章では、本研究で作成した訓練環境構築支援システムの構成とシステムの使用方法について述べた後、構築した支援システムと訓練環境を第三者に使用してもらうことによって、その評価を行った結果について述べる。

5.1 構築支援システムのソフトウェアとハードウェア構成

本研究で作成した訓練環境構築支援システムは、訓練に必要な情報を入力する環境と同時に、訓練を実行する環境も提供する。情報を入力する者（以下オペレータと呼ぶ）は必要に応じて両環境を切り替えて利用することができる。訓練に必要な情報を入力する場合に用いるハードウェア、ソフトウェアと、訓練を実行する場合に用いるハードウェア、ソフトウェアが若干異なるため、以下では、そのそれぞれについて説明する。支援システムの構築で用いたハードウェア、ソフトウェアの一覧を、訓練環境の構築と訓練実行の各場合に分けて、表 5.1 に示す。

5.1.1 訓練に必要な情報を入力する場合のシステム構成

訓練に必要な情報を入力する場合のシステムの構成を図 5.1 に示す。システムは、計測部、情報処理部、情報提示部に分けることができる。計測部は 3 次元マウスと 2 次元マウス、キーボードで構成され、情報提示部は Display で構成される。情報処理部は SGI 社のワークステーション (Maxim um Impact) で構成され、ソフトウェアとしては、OpenGL ライブラリ¹⁷⁾と Motif ライブラリ¹⁸⁾を使用した OpenGL ライブラリは、3 次元画像処理のために開発された C 言語ライブラリであり、用意されている関数は 120 種類にのぼる。Motif ライブラリは X (X Window System Software Foundation) が提唱している X ウィンドウ上の高レベル GUI を提供するためのライブラリであり、様々な「部品」(ポップアップメニューやプッシュボタンなど)を作成するためのライブラリコールからなる。オペレータは OpenGL ライブラリを用いてレンダリングされた仮想空間をディスプレイを介して眺めながら、3 次元マウスと 2 次元マウスを操作し必要な情報を入力していく。2 次元マウスとキーボードは、主に情報の選択や、数値や名称の入力に用い、3 次元マウスは、主に仮想空間内の

3次元位置を指定する場合に用いられるGUI環境の詳細は5.4節で述べる。

5.1.2 訓練を実行する場合のシステム構成

訓練を実行する場合のシステム構成を図5.2に示す。システムは、一般の人工現実感技術を用いたシステムと同様に、計測部、シミュレーション部、情報提示部で構成される⁴⁾。計測部は3次元マウスで構成される。訓練生は、3次元マウスのパックとボタンを操作することによって、手の把持・開放等のジェスチャに対応する動作を行うことができる。シミュレーション部は、ハードウェアとしてSG社のワークステーション、ソフトウェアとしてOpenGLライブラリを用いる。情報提示部は、大型ディスプレイとCrystalEyesとで構成され、訓練生はOpenGLライブラリによって描かれた図5.3のような3次元画像をCrystalEyesを通して立体映像として見ながら訓練を行う。

5.2 仮想空間内で模擬する必要がある物体の運動と物体間の拘束

本節では、訓練環境を構築する際に、仮想空間内の物体の状態が定まった後に、どのような物体の運動と物体間の拘束関係を模擬する必要があるかを述べる。なお本節での検討においては、具体例として原子力発電所のプラントの構成機器である、逆止弁と横置きポンプの分解組立作業を参考にして、本システムで少なくとも模擬すべきものと考えられる運動と拘束関係を抽出した。

1. 訓練生が手に物体を持っている場合

この場合、手は訓練生の意志に従って自由に運動することができ、握られている物体は、訓練生の手の動きに追従する。つまり、訓練生が手を平行移動させると、握られている物体も同じ速さで平行移動し、訓練生が手を回転させると、握られている物体も、手を中心にして回転する。従って、仮想空間内でこれらの運動を模擬するためには、訓練生のジェスチャに従った自由な運動と、手を中心にした回転を表現する必要がある。

2 物体が自由落下する場合

訓練生が握っていた物体を手放した場合、物体は、実世界では自由落下（等加速度運動）する。仮想空間内でも同様の現象を模擬する必要があるが、仮想空間内ではこの運動は必ずしも等加速度である必要はない。それは、例えば、等速

運動であっても、真下に向かって運動していれば、物体が落下していると知覚することができるからである。以上から、等加速度運動もしくは等速度運動を再現できるようにする必要がある。

3. スパナでボルトにはまったナットを回転させる場合

この場合、訓練生の手、スパナ、ナットは、全て、ナットがはまっているボルトを中心軸とした回転運動であり、回転の方向に応じた並進運動も行う。従って、仮想空間内でこれらの運動を模擬するためには、同時に複数の物体が、ある一つの軸を中心にした回転運動を行うように設定できる必要がある。また、ボルトとナットのように、回転角度に応じて並進運動する場合を模擬するためには、回転角度をもとに、平行移動量を設定できる必要もある。

4. ボルトにはまった蓋を外す場合

複数のボルトに蓋がはまっていて、その蓋を手で外そうとする場合、この蓋と訓練生の手は、ボルトの軸方向に平行な運動に限られることになる。従って、拘束関係を模擬するためには、物体の運動が、ある他の物体の軸に沿った平行移動に限定できる必要がある。

5. モータの回転を模擬する場合

モータの回転等は、訓練生の手動きに関係なく、常にある一つの軸を中心にした回転運動である。従って、この運動を、仮想空間内に再現するためには、時間の推移をパラメータにして、物体の回転角度を指示できる必要がある。

6. 蓋の付いた道具入れを模擬する場合

蓋の付いた道具入れは、訓練生が蓋を握ることによって、蓋は、蓋の付け根を中心にした、回転運動を行う。さらにこの回転運動は、蓋が完全に閉まる位置にきた場合は、もうそれ以上に回転することはできない。このような回転と拘束関係を模擬するためには、物体が、ある軸を中心にして、範囲制限のある回転運動を行うことができるように設定できる必要がある。

5.3 支援システムで入力可能なパラメータ

ここでは前節での検討をもとに、本システムで入力可能としたパラメータの種類とそのパラメータを設定することによって構築可能とする訓練環境について述べる。パラメータは物体の各状態ごとに設定することができ、物体の動き（付録 C 参照）に関する情報と、物体の外見に関する情報に大別することができる。

- 物体の外見に関する情報

物体の外見に関しては表 5.2 の情報を設定可能とした。以下にそれぞれについて説明する。

- － 3次元形状データ

物体の表面形状を格納したファイル名を指定する。この情報を各状態に対して設定できるようにすることにより、例えば「開いた手」と「握った手」「正常なコップ」と「割れたコップ」等も表現することができる。

- － テクスチャデータ

物体の表面に張り付けることのできる画像ファイル名を指定する。これにより、臨場感のある仮想空間を構築することが可能になる。また、手に触られている物体のテクスチャを、特別な色（例えば赤一色等）に設定することにより、訓練生に対する力覚フィードバックの欠如を補完する等の工夫を行うことも可能になる。

- － 描画方法

実際の描画の方法として、図 5.4 のように物体の表面を表す面を塗りつぶして描画する方法（ポリゴン）、図 5.5 のように物体の表面を線だけで描画する方法（ワイヤフレーム）、何も描かない（透明）の3つを選択できるようにした。このようにすることにより、訓練生に物体の現在の状態を視覚的に示したり、機器の内部の構造を示すことも可能になる。

- 物体の動きに関する情報

物体の動き（位置と姿勢の変化）は、時間の推移に応じて変化する場合と、3次元マウスからの入力に応じて変化する場合のどちらかを選択できるものとした。表 5.3 に、物体の動きの入力情報の種類と指定方法を示し、以下にそれぞれについて説明する。

- － 時間の推移に応じて変化する場合

この場合は、物体の位置 (X,Y,Z) と姿勢 (A,B,C) を時間の関数として設定できるものとした。例えば、モータの回転は、姿勢の1つのパラメータのみを時間の1次関数で表現し、他のパラメータは変化しないように設定することで実現される。同様に、物体の自由落下も表現することができる。

- － 3次元マウスからの入力に応じて変化する場合

3次元マウスからの入力は、X軸、Y軸、Z軸の平行移動変位 (x,y,z) と、それぞれの軸を中心とした回転変位 (a,b,c) となる。オペレータはこれらの

入力と仮想空間内の物体の位置と姿勢を表すパラメータとの関係を設定することにより、物体の動きを制御する。例えば、仮想空間内のある物体の位置を (X,Y,Z) 、姿勢を (A,B,C) とすると、式 5.1 の場合は、マウスの入力に従って自由に動く手を表す。

$$\begin{cases} X = X + x & A = A + a \\ Y = Y + y & B = B + b \\ Z = Z + z & C = C + c \end{cases} \quad (5.1)$$

ボルトにはまり、手に握られているナット等表現する場合は、式 5.2 のようになる。

$$\begin{cases} X = X & A = A + a \\ Y = Y + ha & B = B \\ Z = Z & C = C \end{cases} \quad (5.2)$$

ただし、 A の値は、ボルトの軸方向を中心にした回転角度を表し、 h は設定可能な定数である。訓練環境を構築する場合は、上記の関係を、項目の選択と数値の入力で行う。本システムではさらに、物体の可動範囲を数値で設定することにより、接触の判定を行わなくても、2 つの物体が仮想空間内で重なり合わないようにすることもできる。

– 物体の各状態における初期位置

仮想空間内で物体の状態が変化する際に、物体の位置や姿勢をある決まった値に変更したい場合が多々ある。例えば、ペンを手で握る場合、ペンの握り方は図 5.6 に示すようにペンの先を下にして握るのが自然である。また、スパナやペンチも握る箇所が決まっている。このような場合に、ペンが手に触られている状態から手で握られている状態に遷移するときに、ペンの位置が、手に握られる位置に必ず変化するように設定できると便利である。そこで本システムでは、状態変化が起きたときに、位置が変化する場合と変化しない場合を選択できるようにし、さらに位置が変化する場合は、位置を指定する基準となる物体と、それを基準とした位置と姿勢を数値で指定できるようにした。

汎用性を高めるために、入力可能なパラメータを数多く準備すると、パラメータの設定作業が煩雑になり、かえって訓練環境を構築する上での効率を低下させる可能性がある。そこで本研究では、設定できる範囲を予め制限してその範囲内でのみ設定可能とした。また、支援環境に GUI を採用し、各パラメータの設定を $\mathbb{C}$ を介して行うことが可能となるようにシステムを作成した。

また、仮想空間内の物体の状態では、良く似た物理法則を適用すべき状態が数多く存在することに注目し、良く使用するパラメータの値の組み合わせを、「状態テンプレート」として登録し、これも GUI を介して選択することができるようにすることにより、さらに訓練環境の構築の効率をあげる試みを行った。

5.4 訓練環境構築支援システムのインタフェース

以下ではこれまでに述べた情報を、支援システムに入力する手順とその際のインタフェースについて説明する。本システムの訓練環境を構築する機能を利用する時の画面例を図 5.7 に示す。本システムでは、図 5.8 に示すようなアイコンを 2 次元マウスを用いてクリックすることによって、各情報の入力画面を表示することができる。入力の手順は大きく次の 4 段階に分けることができ、全ての段階で、 環境を通して情報を入力することができる。

1. 訓練に必要な物体名の登録

本支援システムで訓練環境を構築する場合、訓練に用いる物体名を最初に登録する必要がある。例えば、「右手を用いてペンでふたに印を付ける」訓練（以後「印を付ける訓練」と呼ぶ）を構築する場合、表 5.4 に示す 4 つの物体名を図 5.9 に示すインタフェースを通してキーボードで入力する。

2. 各物体が取りうる状態の入力

前段階で登録した物体のそれぞれに対して、訓練の途中で取りうる状態の登録を行う。例えば印を付ける訓練を構築する場合、ペンは表 5.5 に示すような 5 つの状態を取る。そして、それぞれの状態について、図 5.10 ~ 図 5.13 に示すようなインタフェースを通して、ペンの形状データや、各状態における物理的性質を決定する情報を入力する。この際、本システムでは、頻繁に入力が必要となる状態を表現する情報を、状態テンプレートとして登録することができる機能を設けており、図 5.14 に示すように、状態テンプレートから、必要なテンプレート名を選択することにより、必要な情報を容易に設定することができる。例えば、新たな状態を作成する際に、テンプレートから「手を開いた状態（自由に動く）」を選択すると、パラメータの各項目に表 5.6 に示すような情報が自動的に入力される。

3 イベントの発生に応じた各物体の状態遷移の方法の入力

ペトリネットを構築することによって、イベントの発生に応じた各物体の状態遷移の方法を設定する。第 4 章で述べたように、ペトリネットにおけるプレー

ス 1 つに対して物体の 1 状態を対応させ、訓練生のジェスチャや他のイベントに応じて物体の状態が変化することをトランジションの発火で表現する。具体的には、ユーザは図 5.15 に示すようなインタフェースを通して、物体の変化前の状態と物体の変化後の状態（第 4 章で述べたようにそれぞれ入力プレース、出力プレースに対応する）を入力し、さらに図 5.16 に示すインタフェースを通して、状態遷移を起こさせる原因となるイベントの設定を行う。本システムではさらに、抑止アークと参照プレースを設定することもできる。構築したペトリネットは、図 5.17 に示すように自動的に配置され、図 5.18 に示すような拡大・縮小ボタンやスライダーを操作することによって、全体を眺めることもできる。本システムではさらに、物体の状態遷移が起こるときに発する効果音を個々のトランジションに対して設定することができる。

4. 物体の初期状態の登録

訓練を実際に行う為には、各物体の初期状態を指定する必要がある。本システムでは、ペトリネットにトークンを配置し、各トークンに対して、3 次元の初期位置に関する情報を付加することによって、各物体の初期状態を指定する。ここで、プールプレースに複数のトークンを配置することによって、同じ状態にある物体を複数の仮想空間内に配置することができる。例えば、ペンの状態を表す、「床の上にある」プレースに 2 つのトークンを置き、それぞれに 3 次元初期位置を設定することによって、2 本のペンを仮想空間内に容易に配置することができる。

5. 作成した訓練環境データの保存

これまでに作成した訓練環境のデータを、図 5.19 に示すファイルメニューを用いて保存する。また、このメニューを用いることにより、訓練環境データの読み込みや、その修正、新規作成も行うことができる。

本システムでは、上記の機能の他に特徴的な機能として、図 5.20 に示すような環境設定画面がある。これらのボタンは、上から、影を付けるか否かの選択、物体にテクスチャを貼るか否かの選択、物体をポリゴンで描くか、ワイヤーフレームで描くかの選択、床にテクスチャを貼るか否か、の選択を行うためのボタンである。それぞれのボタンの選択により、使用するシステムの性能に応じた 3 次元画像の表現方法を利用することができる。また、本支援システムでは、仮想空間内で手が一旦視界から外れると、再び視界内に戻すことが困難であるという問題に対処するために、図 5.21 に示すように手の可動範囲を制限する機能を付加することとした。

5.5 構築支援システムでの訓練実行時の情報の流れ

本支援システムにおける訓練実行時の情報の流れを図5.22を用いて説明する。訓練実行時は、支援システムと同時にペトリネットシミュレータを走らせ、発火トランジションと、トークン配置に関する情報をやりとりする。以下、訓練を実行する場合の各情報の流れと処理の順序を説明する。

I. 発火可能なトランジションの判定

ペトリネットシミュレータから、現在発火可能なトランジションと、そのトランジションを発火させる原因となる事象に関するデータを取り出す。

II. 訓練生からのジェスチャの入力

まず訓練生からの3次元マウスの入力を調べる。訓練生からの入力、ボタンの状態(押されているかどうか)に関するものと、手がどの軸を中心にしてどのくらい回転させようとしているか、手がどの軸方向にどのくらい移動しようとしているか、に関するものが数値として得られる。そこでこれらのデータはジェスチャ判定で、4.2.項で述べた6つのジェスチャの内の、いずれに相当するかが判定される。

III. 発火トランジションの決定

Iで得られた発火可能トランジションの発火原因となる事象とIIで得られた訓練生からのジェスチャとを比較し、訓練生のジェスチャが、このトランジションの発火原因となるものに一致すれば、それを発火させるべきトランジションとしてペトリネットシミュレータに渡す。

IV. トランジションの発火によるトークン遷移の調査

IIIで示したトランジションが発火することにより、トークンが遷移し、遷移後のトークン配置が、仮想空間の状態を管理する部分に送られる。

V. 各物体の状態の決定

IVで得られたトークンの変化と、訓練対象機器の状態データをもとに、仮想空間内の物体の現在の状態を決定する

VI. 3次元画像のレンダリングと訓練生への提示

各物体の現在の状態に関する情報はレンダリング部に送られ、訓練対象機器のテクスチャデータと訓練対象機器の3次元形状データとを用いて3次元画像をレンダリングし、訓練生に提示する。

VII. 物体の物理的性質シミュレーション

各物体の現在の状態に関する情報と、「訓練対象機器状態データ」をもとに各物体の物理的性質をシミュレーションする。

以上の処理を繰り返すことによって、訓練生のジェスチャを仮想空間に反映させている。

5.6 構築支援システムを用いた訓練環境の構築例とその実験による評価

本節では、本研究室の先行の研究で構築したスイング式逆止弁の分解訓練環境を、支援システムを用いて構築した結果について述べる。続いて、開発した支援システムとそれを用いて構築した訓練環境をそれぞれ第三者に実際に使用してもらうことにより、評価を行った結果について述べる。

5.6.1 支援システムの使用例

本項では、スイング式逆止弁の全体の分解作業の内、表 4.1 に示した範囲内の作業工程を行うことができる訓練環境を構築した結果について述べる。ただし、訓練対象の 3 次元形状データとテクスチャデータは既に作成されている状態から作業を行った。訓練環境を構築するために入力したデータを表 5.7 ~ 表 5.18 に示す。ただし、表 5.7 は、支援システムに登録した物体名の一覧であり、表 5.8 ~ 表 5.14 は、物体が取りうる状態として登録した状態名である。また表 5.15 ~ 表 5.16 は、各状態に対して、物体の外見や、運動、他の物体との束縛関係を模擬するために入力したデータの一部であり、表 5.18 は、訓練の初期配置として入力したトークン配置である。

実際にはこれらのデータに、4.3.2 節で作成したペトリネットを変更したものと、3 次元位置に関する情報が加わる。これらの情報は、テンプレート機能を用いることにより、ほとんどの場合、細部までは入力する必要はない。構築した訓練環境の実行画面を図 5.23 に示す。この訓練環境を構築するために必要とした時間は、約 1 2 時間であり、先行研究でほぼ同じ訓練環境を構築する際に約 4 ヶ月の日数が必要であったことと対比させると、本支援システムの使用により作業時間が大幅に短縮され、飛躍的に作業効率が向上することを示している。

5.6.2 訓練環境構築支援システムの操作性の実験による評価

前節で述べたシステムの使用例は、支援システムを構築した本人による作業であるため、支援システムに精通した者による作業の評価であるといえる。そこで本項では支援システムに対しては初心者の第三者にシステムを利用してもらい評価を行った結果について述べる。

実験の目的

本研究で構築した訓練環境の構築支援システムを第三者に使用してもらい、支援システムを用いることによる訓練環境構築の作業能率の改善を確認する。またアンケート調査により支援システムの改良すべき点を見い出す。

実験方法

- 被験者

被験者は2人で、プログラミングにより訓練環境を構築した経験のない20歳代と40歳代の男性とした。2人とも2次元マウスとキーボードの操作には十分慣れており、必要な情報を不自由なく入力できる。また、2人とも支援システムを初めて使用する。

- 評価方法

被験者に人工現実感技術を用いた訓練環境を支援システムを用いて構築させる。3次元形状データとテクスチャデータは予め作成されていることとし、どのファイルにどのデータが格納されているかを予め被験者に知らせておく。被験者には、支援システムの各機能の説明と操作方法を約30分間説明し、例として「手でペンを握ったり放したりできる訓練環境」の構築作業を実際に見学させる。実験は、作業開始から作業終了までにかかった時間を計測し、また、作成した訓練環境データの内容も記録する。さらに、作業終了後、支援システムを使用した感想を述べてもらう。そして予め準備した、訓練環境構築の回答例と比較し、被験者が正しく訓練環境を構築することができたかどうかを検討する。

- 被験者が構築する訓練環境

以下の訓練環境を構築させる

課題 モータとそのスイッチが配置してあり、スイッチを操作することにより、モータを回転させたり、静止させたりすることができる訓練環境を構築

する。準備する 3 次元形状データは、モータの土台、シャフト、スイッチの台、スイッチの 4 つである。

このような訓練環境を構築するために入力する必要がある情報の一例を、表 5.19 ~ 表 5.24 と図 5.24 に示す。この構築作業では、ペトリネットの作成、すなわち、物体の状態遷移の仕方の設定は、比較的容易であるが、シャフトの回転方法の設定は難しい作業である、と予想を立てた。

実験結果

二人の被験者（被験者 A、被験者 B）は共に訓練環境を正しく構築することができた。訓練環境構築に必要とした作業時間は、被験者 A では 41 分、被験者 B では 38 分であった。難しいと予想された、シャフトの回転方法の設定も、支援システムの各機能を説明する際に実演した方法を真似ることで、混乱することなく、設定することができたようである。ただし、被験者 B は、ペトリネットを作成する際に、物体の状態の分け方にすこし戸惑ったが、試行錯誤の後、正しいペトリネットを作成することができた。実験後のアンケートにより、両者から表 5.25 のような感想が得られた。

考察

実験の結果により、本支援システムを初めて利用する場合でも、単純な訓練環境ならば、試行錯誤しながら、最終的には構築できることが確認された。被験者は 2 人とも、これまでに仮想空間を構築した経験はなく、プログラミングで仮想空間を構築させた場合、プログラミングの方法から学ぶ必要があり、非常に時間がかかるものと思われるが、本システムを用いると、多くても数時間の練習を行うだけで、目的の仮想空間を構築できるようになるとと思われる。

一方、アンケートの結果からは、オブジェクトの位置合わせのインタフェースを改良すべきであることが判明した。また、3 次元マウスを用いて、正確なオブジェクトの位置を設定することは、初心者には困難であることも判明した。GUI 環境で、2 次元マウスを用いて位置合わせを行うことができるインタフェースを構築すべきであると思われる。

5.6.3 支援システムで構築できる訓練環境の実験による評価

支援環境で構築することのできる訓練環境は 3.1 節で述べたように、奥行き知覚特性の改善と、力覚フィードバックの補完を目的とした工夫がなされている。本項

ではそれらの工夫に対して実験により評価を行った結果について述べる。

実験の目的

本支援システムでは、作業効率の改善のために、(1) イベント音の提示、(2) CrystalEyes の使用、(3) 影の提示、の 3 種の補助機能を利用することを可能としているが、それらを用いる場合と、用いない場合とでは、作業効率にどのような影響が生じるかについて調べる。また実験後、アンケートを行い、主観的にどの環境が最も容易に作業を行うことができるかについても調べる。

実験方法

- 被験者

被験者は 6 名で、CrystalEyes を用いて 2 眼式立体視が比較的容易にできる 24 ~ 26 歳の男性とした。

- 作業内容

図 5.25 に示すように、仮想空間内に 5 つのナットをそれぞれ視点からの距離が異なるように配置する。ただし、全てのナットは、床が奥行きを知覚する基準として用いられないように空中に配置してある。それらのナットを仮想空間内の右手を 3 次元マウスを操作してつかみ、手前の板に載せる作業を表 5.26 に示す 5 種類の異った環境において行う。作業の途中でナットを板の上以外の場所に落とした場合は、それを拾い上げ、再び板の上に置くこととする。全てのナットが板の上に載った時点で作業終了とし、作業開始から終了までの時間を計測する。実験は、同じ環境を 5 回続けて環境 1 から環境 5 まで実験した後、再び環境 1 について 5 回実験を行い、計 30 試行実験を行うグループ (I グループ：被験者 A,B,C) と、環境 1 から環境 5 までを順番に 1 回ずつ実験を行い、それを 5 回繰り返すことにより、計 25 試行実験を行うグループ (II グループ：被験者 D,E,F) の 2 つに分けて行った。実験の前に、それぞれの被験者に 3 次元マウスの操作に慣れてもらうため、5 分間ずつ各環境に対して練習を行った。また、CrystalEyes を用いた環境は、被験者が全ての距離のナットに対して立体視ができるように、立体視のパラメータ (両目の瞳孔間隔) を個々に調整した。

実験結果

各被験者の作業時間を測定した結果を、表 5.27～表 5.32と図 5.26～図 5.31に示す。グラフは横軸が各環境の実験回数、縦軸が作業時間（単位は秒）である。またそれぞれの被験者に対して、各環境ごとの作業時間の平均を比較したグラフを図 5.32に示す。このグラフは、横軸が環境、縦軸が作業時間の平均（単位は秒）である。さらに、各被験者に、各環境の使いやすさに順位を付けてもらった結果を表 5.33に、実験に対する感想をまとめたものを表 5.34に示す。なお、被験者 E からは、感想を得ることができなかった。

考察

- 同一被験者の同一環境における作業時間のばらつきについて

同じ環境の間でも、その実験が何回目であるかによって、結果に大きな差がある被験者があるが、アンケートの結果によると、時間がかかった試行は「一度失敗したら、焦って連続して失敗してしまった」とのことである。また、時間がかからなかった試行は「カンで掴んだら、たまたま上手く掴めた」とのことである。このことを考慮し、全被験者の各環境における、作業時間の最大と最小を取り除いて平均を比較するグラフを作成した結果を図 5.33に示す。このグラフと、全ての試行について単に平均を取ったグラフ（図 5.32）とを比較すると、各環境間の関係はほぼ同じ傾向が出ており、実験結果にさほど影響を与えていないと思われる。

- 被験者の 3 次元マウスの操作の慣れについて

I グループの実験結果をしてみる。全ての被験者において、最初に実験を行った環境 1 の結果と、最後に実験を行った環境 1 の結果を比較すると、最後に行った環境 1 の方が、平均の作業時間が小さい。これは実験前の練習が十分でなく、3 次元マウスの操作に実験中に慣れていったためであると思われる。しかし II グループでは、全ての環境を一度ずつ順番に実験を行っているため、慣れが全ての環境に対して同じように影響するものと期待される。

- 支援機能の効果について

図 5.32を見ると、全ての被験者について、機能を全く用いない環境（環境 1）よりも、機能を一つでも用いた環境（環境 2, 3, 4）の方が、作業終了までの時間が短い。このことより、影、音、立体視それぞれに効果があることが確認された。しかし、被験者 D, F に関しては、立体視が他の被験者に比べて、効果が少ない。これは、アンケートでも、「立体視は見づらい」という意見が出ていたこ

とも考慮すると、D、F 両被験者が、立体視が苦手であることが原因であると思われる。主観的な使いやすさに関しては、アンケートによると、影が最も使いやすく、音がもっとも使いにくいという結果が出ている。

- 作業を行う際の支援機能の利用傾向について

作業終了までに要した時間の平均（図 5.32）を見てみると、全ての被験者で、環境 1 が最も時間がかかり、環境 5 が最も時間がかからないという結果が出た。アンケートの結果でも、環境 1 は非常に難しく、環境 5 が最も容易であったという感想を得ている。このことは、本訓練環境の支援機能が効果的に働いていることを示している。なお、環境 2、3、4 よりも、環境 5 が最も容易であるという結果より、補助機能を複合して用いることの効果が確認された。

なお、アンケートの結果では、立体視は物体の大体の場所を知るのには有効であるが、微妙な位置まで知るのには向いていないという感想があり、また影は微妙な位置を知りたい時と、物体を板に落とすときに有効であったと述べている。さらに「影と立体視を同時に使うのが最も作業が行い易いであろう」という感想を述べている被験者も 2 名いた。

結論

本実験により、作業の支援機能を用いた場合の方が、用いない場合より効率的に作業ができ、複数の補助機能を複合する方が有効なことが確認された。また、立体視の機能に関しては、立体視が得意な者と不得意な者がいることを考慮すると、立体視のみを用いた従来の訓練環境は、訓練生に物体の前後関係を知覚させるのには不十分であることが分かる。

なお、今回の実験では、影を用いた訓練環境は、訓練生の作業効率を改善する有効な手段であることを確認したが、この場合は仮想空間が比較的単純で、物体の数も少なかった。しかし、対象が複雑になってくると、影は他の物体に遮られる可能性があり、影のみを用いた訓練環境は、問題があると思われる。

第 6 章 結論

本研究では、人工現実感技術を用いた機器保修の訓練環境を効率的に構築するための、訓練環境の構築支援システムの開発を行った。

第 2 章では、人工現実感技術を用いた訓練環境の利点と、その構築を行う際の問題点について述べ、本研究の方針について述べた。第 3 章では、支援システムの開発に際して、人工現実感技術を用いた訓練環境が具備すべき特徴について検討し、それらの特徴を持つ訓練環境を構築するために作成する必要がある情報についてまとめた。さらに、支援システムが具備すべき機能について検討し、GUI 環境で訓練環境を構築できる支援システムを設計した。第 4 章では、仮想空間内の数多くの物体の状態変化の仕方を、ペトリネットを用いてモデル化する方法を提案した。ペトリネットを用いることにより、効率的に各物体の状態を管理できるだけでなく、ユーザーが視覚的に物体の状態変化の仕方を設定することができるようになる。第 5 章では、本研究で開発した、訓練環境構築支援システムの全体構成について述べ、訓練環境の具体的な構築手順について述べた。また、訓練環境を不便なく使用できるように、奥行き知覚特性の改善と、力覚フィードバックの補完の 2 点に注目し、幾つかの工夫を行い、その効果を実験により確認した。また、本研究で開発した支援システムの第三者による使用実験により、訓練環境の構築における作業効率が飛躍的に向上することを確認した。

本研究で開発した支援システムを用いることにより、訓練環境を構築する際の労力を軽減することは可能となったが、人工現実感技術を用いた訓練環境を実際の訓練に用いるためには、まだまだ改善すべき点が多々ある。例えば、訓練生に、作業手順を効果的に教示するための自動実演機能の構築は今後の課題である。また、訓練履歴を記録し、その結果を訓練生にフィードバックする環境もまだ構築されていない。また、訓練を行う際のデバイスは、まだ実用的なレベルに達しておらず、ある程度、人がその操作に慣れるまで努力を要するので、デバイスの改良についても今後の研究が必要である。

人工現実感とは、その登場以来、多くの人の努力によって飛躍的に発展してきた。今後も、今以上の速度で発展していくことが予想される。この技術が人類の究極の夢をかなえるものなのか、それとも文明に対するもっとも恐るべき悪夢となるのか。いずれにせよ、幻想と現実の境界が消える日もそう遠くないと思われる。

謝 辞

本論文の執筆にあたり、適切に御指導いただいた吉川榮和教授に深謝致します。

本研究をすすめる上で御指導をいただき、数々の助言をいただいた手塚哲央助教授に深謝致します。

また、本研究を暖かく見守って下さり、助けていただくことも多かった下田宏助手に深謝致します。

研究中何かとお世話いただき、実験の被験者にもなっていただいた吉川研究室の学生の皆様にも心から感謝致します。

最後に、新婚であるにも関わらず、毎日夜遅くに家に帰ってくる私に対して、泉のように深い愛情と支援を与えてくれた、私の妻令乃奈に心から感謝します。

参 考 文 献

- 1) Loftin, R.B. and Kenney, P.J.: Training the Hubble Space Telescope Flight Team. IEEE Computer Graphics and Applications, 15, 5, (1995), pp.21-37.
- 2) 三谷 拓也:仮想現実感による機器の分解組立シミュレーションに関する研究, 京都大学工学部電気工学第二学科学士論文, (1995).
- 3) 柏 健一郎:仮想空間における機器保修訓練システムの構築, 京都大学大学院工学研究科電気工学第二修士論文, (1996).
- 4) 舘 日章, 他:バーチャル・テック・ラボ, (工業調査会,1992).
- 5) 森 健策, 他:3次元胸部CT像に基づく気管支内視鏡シミュレーションシステム(仮想化気管支内視鏡システム), 医学電子と生体工学(日本エム・イー学会論文誌), 33, 4, (1995), pp. 343-351
- 6) 塚田 敏彦, 他:表面反射率が不均一な物体の3次元形状計測, 電気学会, 電子・情報・システム部門誌, Vol.117-C, (1997), pp. 1355-1361
- 7) 中島 豊四郎, 伊藤 宏隆, 南里 孝幸:ペトリネットによる伸線プラントシステムの解析, 電気学会, 電子・情報・システム部門誌, Vol. 144-C(1994), pp. 881-887.
- 8) IEXER RESEARCH Inc.: aWORDTM Reference Manual, (1997).
- 9) 後藤 善裕, 他:2台の傾斜対向配置EDカメラによる手の形状認識, 日本バーチャルリアリティ学会第2回大会論文集, (1997), pp. 216-217
- 10) SPACE CONTROL & MAINTENANCE SPACE NOISE User's Manual, (1997).
- 11) C Petri: Kommunikation mit Automaten, Ph.D dissertation, University of Bonn, Bonn, West Germany, (1962).
- 12) 村田 忠夫:ペトリネットの解析と応用, (近代科学社,1992).
- 13) K Jensen & David Petri Nets. Basic Concepts, Analysis Models and Practical Use. Volume1:Basic Concepts, (1992). Volume2:Analysis Models,

(1994). **Volume3**: Practical Use, (1997). Monographs in Theoretical Computer Science, Springer-Verlag.

- 14) Committee Draft ISO/IEC 15909: High-level Petri Nets - Concepts, Definition and Graphical Notation, **Version 3.4** (1997).
- 15) J. L. ピーターソン: ペトリネット入門, (シュプリンガー・フェアラーク東京株式会社, 1992).
- 16) R. Lipton: The Reachability Problem Requires Exponential Space. Research Report 62, Department of Computer Science, Yale University, (1976).
- 17) 相川 恭寛: OpenGL プログラミング・ガイドブック, (技術評論社, 1995).
- 18) 木下 凌一, 他: X-Window OSF/Motif プログラミング, (日刊工業新聞社, 1994).

目 次

3.1	仮想空間内の物体に定規を付加する方法	49
3.2	仮想空間内に提示する影の例	49
3.3	システムの概念的構成	50
4.1	スイング式逆止弁の構造	51
4.2	手とペンの状態の遷移	52
4.3	ペトリネットの定義一覧	53
4.4	ノーマルアーク	53
4.5	抑止アーク	53
4.6	参照アーク	53
4.7	手でナットを外す(その1)	54
4.8	手でナットを外す(その2)	54
4.9	手に握ったスパナをナットに当てる(その1)	55
4.10	手に握ったスパナをナットに当てる(その2)	55
4.11	スパナでナットを締める, 緩める, 外す(その1)	56
4.12	スパナでナットを締める, 緩める, 外す(その2)	56
4.13	スパナでナットを締める, 緩める, 外す(その3)	57
4.14	スパナでナットを締める, 緩める, 外す(その4)	57
4.15	蓋を握る, スパナを握る	58
4.16	ペンで蓋に印を付ける	58
4.17	内圧を確認する, ナットを握る	59
4.18	ペトリネットと仮想空間の関係	59
4.19	ペンとナットを手でつかむ作業のペトリネット	60
4.20	カラーペトリネットを用いたモデル化	60
5.1	訓練環境構築時のシステム構成	61
5.2	訓練実行時のシステム構成	62
5.3	訓練生に提示される映像の例	63
5.4	Polygon による描画	64
5.5	Wireframe による描画	64
5.6	ペンの握り方	65
5.7	訓練環境の構築時の画面例	66

5.8 アイコンの例	67
5.9 新たに物体を登録する際のインタフェース	67
5.10 状態を作成する際のインタフェース(外見に関する情報)	68
5.11 状態を作成する際のインタフェース(初期位置に関する情報)	69
5.12 状態を作成する際のインタフェース(平行移動に関する情報)	70
5.13 状態を作成する際のインタフェース(時間に応じた移動法に関する 情報)	71
5.14 テンプレート機能	72
5.15 新規トランジション作成画面	73
5.16 イベント設定画面	73
5.17 ペトリネット表示画面	74
5.18 ペトリネット拡大縮小ボタンとスライドバー	74
5.19 ファイル操作選択画面	74
5.20 環境設定画面	75
5.21 両手の可動範囲	75
5.22 訓練実行時の情報の流れ	76
5.23 訓練環境の構築例	81
5.24 構築すべきペトリネット	83
5.25 5つのナットを空中に配置した仮想空間	84
5.26 被験者 A	87
5.27 被験者 B	87
5.28 被験者 C	88
5.29 被験者 D	88
5.30 被験者 E	89
5.31 被験者 F	89
5.32 各環境における平均値	90
5.33 各環境における平均値(最大値、最小値を除外)	91
A 1 Magell an.	93
A 2 Crystal Ey es による立体視の原理	93
B 1 冗長なトランジション	100
B 2 融合前のトランジション	100
B 3 融合後のトランジション	100
B 4 融合前のトランジション	101
B 5 融合後のトランジション	101

B.6 分割可能なペトリネット	102
B. 7 階層化が可能なペトリネット	102
B. 8 モデル化の対象とする作業空間	103
B. 9 モデル化したペトリネット	104
B. 1 分割したペトリネット (その 1)	104
B. 1 分割したペトリネット (その 2)	105
B. 1 階層化した後の 2 つのトランジション	105
B. 1 階層化した後のペトリネット	105
C. 1 絶対座標系とオブジェクト座標系	106

表 目 次

4.1	逆止弁の分解作業の手順（一部）	51
4.2	「ペンでふたに印を付ける」作業のジェスチャによる表現	51
4.3	制御スペースにより効果的にモデル化できる訓練環境の例	52
5.1	支援システムの構成	61
5.2	訓練環境を構築するために入力する物体の外見に関する情報一覧	63
5.3	訓練環境を構築するために入力する物体の動きに関する情報一覧	65
5.4	「印を付ける訓練」を作成する場合に入力する物体名	65
5.5	ペンが取り得る状態	66
5.6	自動的に入力されるパラメータ	72
5.7	登録する物体名	76
5.8	登録する状態（右手）	77
5.9	登録する状態（ボルト 1 ～ 4 ）	77
5.10	登録する状態（蓋）	77
5.11	登録する状態（ナット）	78
5.12	登録する状態（ペン）	78
5.13	登録する状態（スパナ）	78
5.14	登録する状態（印）	78
5.15	状態におけるパラメータ（手に握られたペン）	79
5.16	状態におけるパラメータ（ボルトを中心に回転するナット）	79
5.17	状態におけるパラメータ（落下するナット）	80
5.18	初期トークンの配置	80
5.19	登録する物体名	82
5.20	登録する状態（hand,motor-body,switch-body,with）	82
5.21	登録する状態（shaft）	82
5.22	状態におけるパラメータ（何もしていない）	82
5.23	状態におけるパラメータ（回転するシャフト）	82
5.24	状態におけるパラメータ（静止しているシャフト）	83
5.25	支援環境を利用した感想	84
5.26	実験の環境	84
5.27	実験 2 の結果（被験者 A ）(単位は秒)	85

5.28 実験 2 の結果 (被験者 B)(単位は秒)	85
5.29 実験 2 の結果 (被験者 C)(単位は秒)	85
5.30 実験 2 の結果 (被験者 D)(単位は秒)	86
5.31 実験 2 の結果 (被験者 E)(単位は秒)	86
5.32 実験 2 の結果 (被験者 F)(単位は秒)	86
5.33 各環境の主観的な使いやすさの順位	90
5.34 訓練環境を使用した感想	91
B 1 ペンで蓋に印を付ける作業に対応する発火系列 (一部)	95

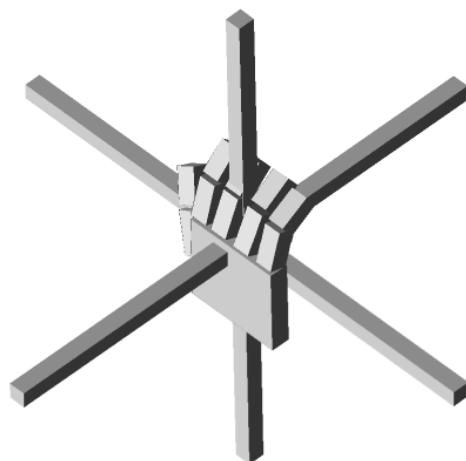


図 3.1: 仮想空間内の物体に定規を付加する方法

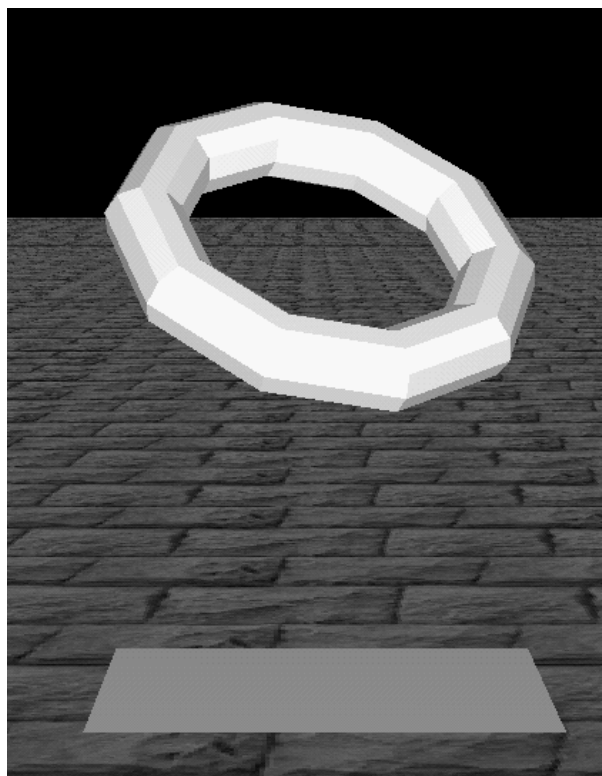


図 3.2: 仮想空間内に提示する影の例

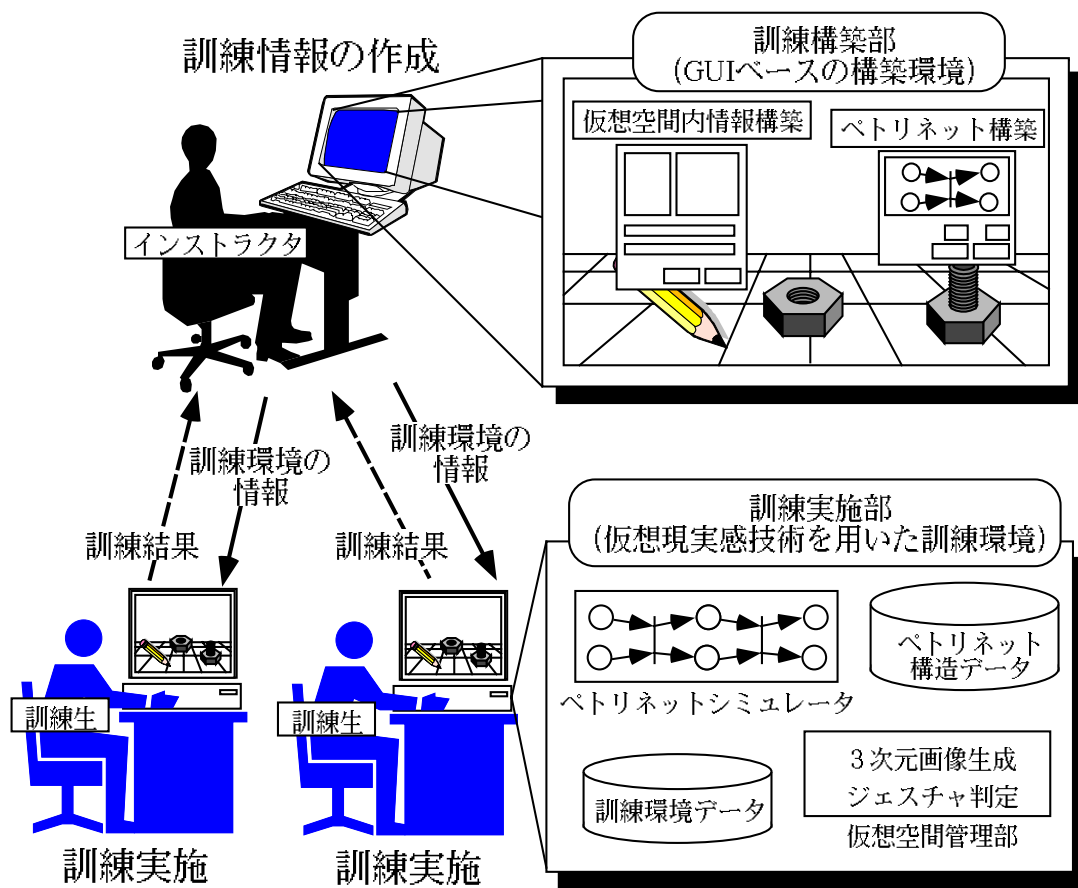


図 3.3: システムの概念的構成

表 4.1: 逆止弁の分解作業の手順（一部）

作業工程	作業内容
工程 1	ペンで弁箱に合いマークを付ける
工程 2	ふたボルトのナットをレンチでゆるめる
工程 3	ふたを少し持ち上げ、内圧がかかっていないことを確認する
工程 4	ふたのボルトのナットをすべて取り外す
工程 5	ふたを取り外す

表 4.2: 「ペンでふたに印を付ける」作業のジェスチャによる表現

作業内容	対応するジェスチャ
右手とペンを接触させる	ジェスチャ 1
右手でペンを握る	ジェスチャ 3
ペンでふたに印を付ける	ジェスチャ 1
ペンをふたから離す	ジェスチャ 2
ペンを手放す	ジェスチャ 4

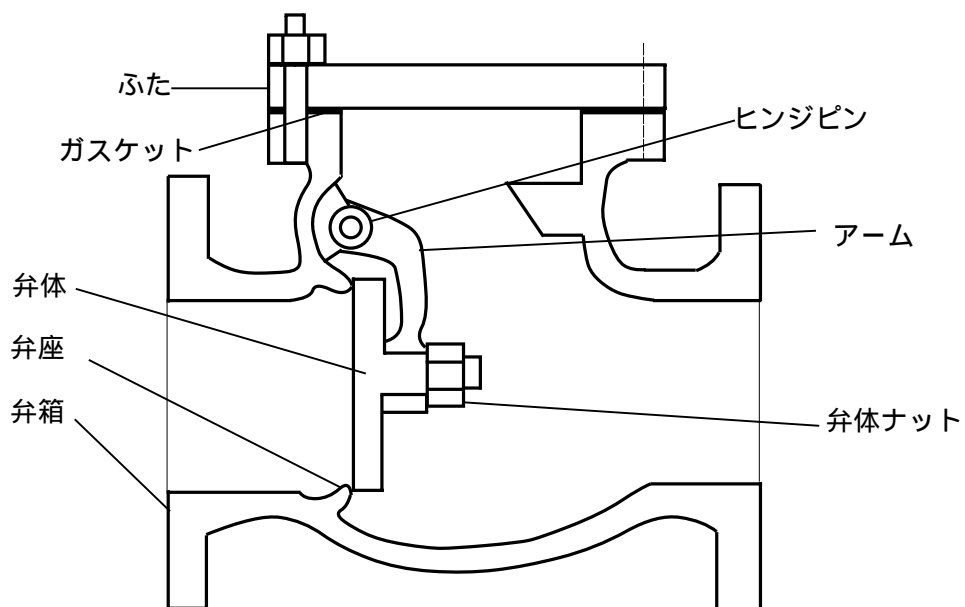


図 4.1: スイング式逆止弁の構造

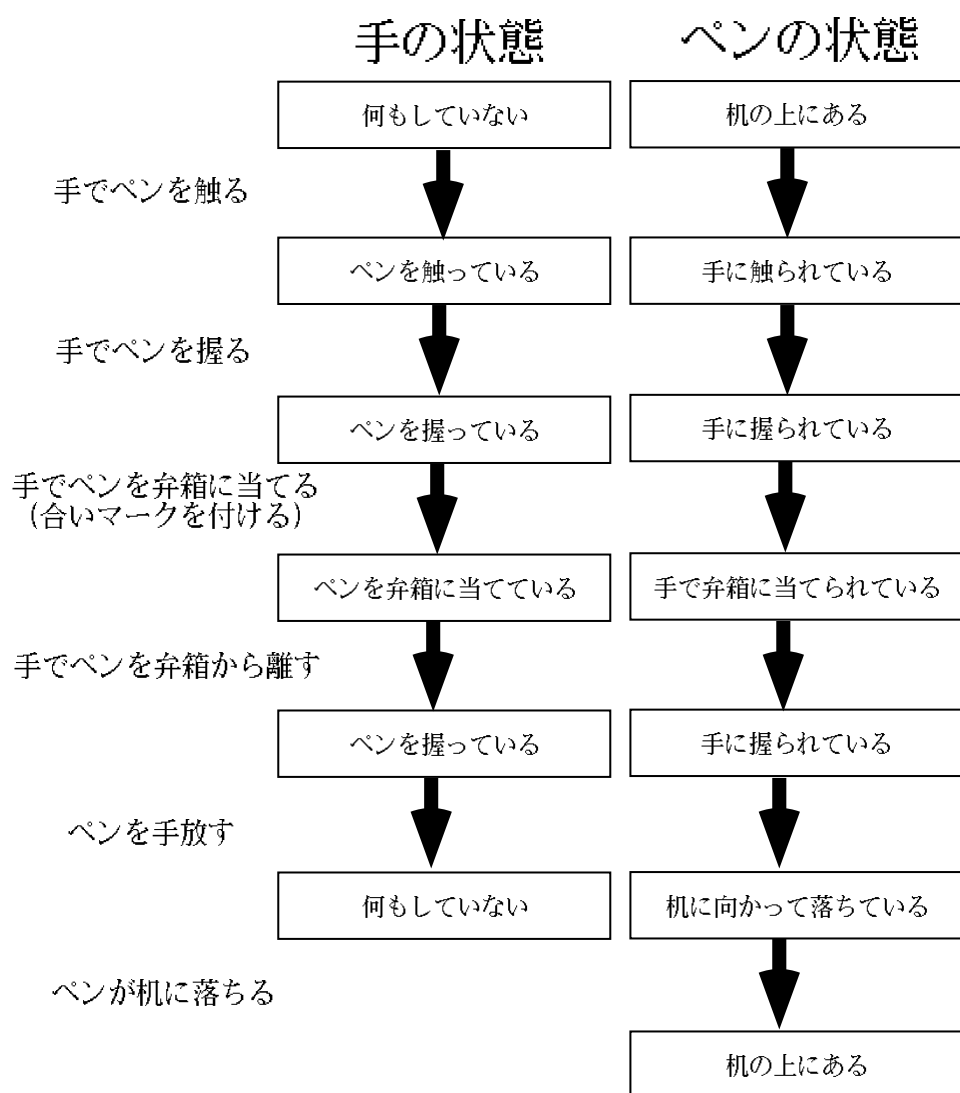


図 4.2: 手とペンの状態の遷移

表 4.3: 制御プレースにより効果的にモデル化できる訓練環境の例

1	ナットは手で握ることができる
2	ナットはボルトにはまることができる
3	ナットはスパナで締め付けることができる
4	ナットはペンで印を付けることができる

Terminology	Symbol	Terminology	Symbol
Single Place		Auto Transition	
Pool Place		Normal Arc	
Control Place		Inhibitor Arc	
Normal Transition		Reference Arc	
		Token	

図 4.3: ペトリネットの定義一覧

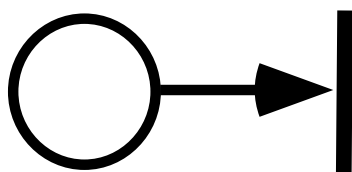


図 4.4: ノーマルアーク

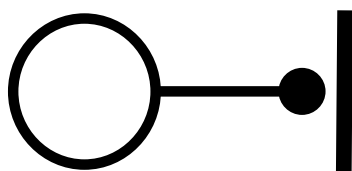


図 4.5: 抑止アーク

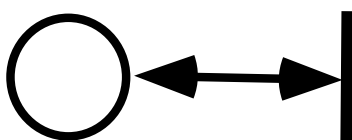


図 4.6: 参照アーク

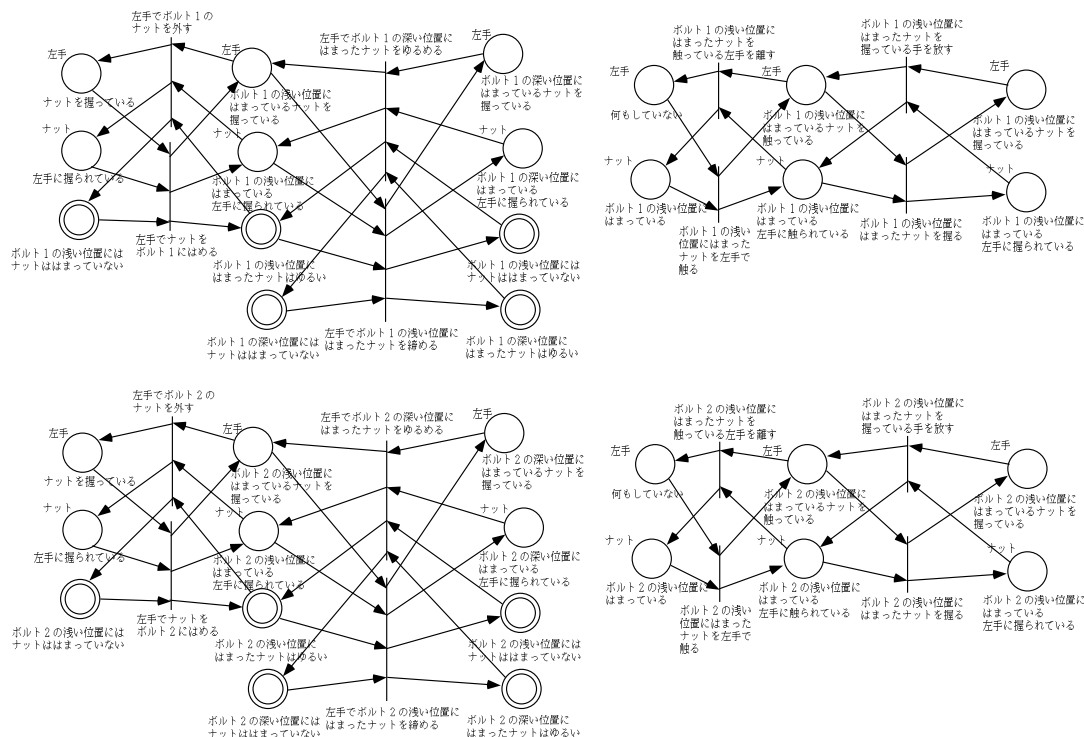


図 4.7: 手でナットを外す (その 1)

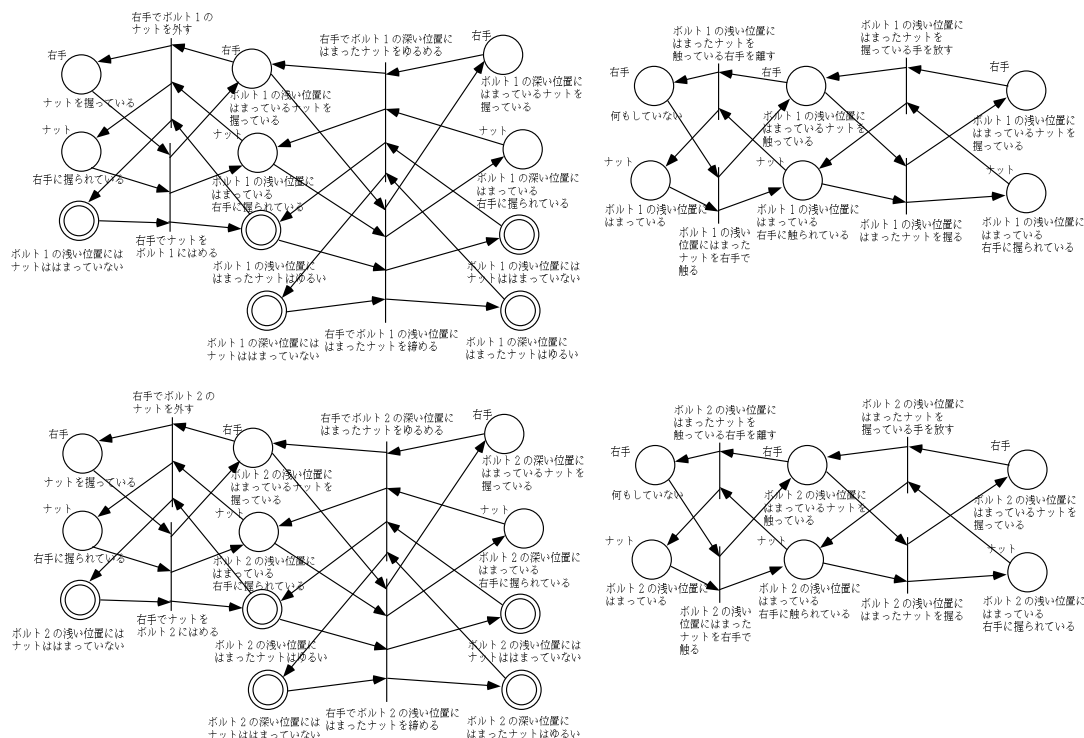


図 4.8: 手でナットを外す (その 2)

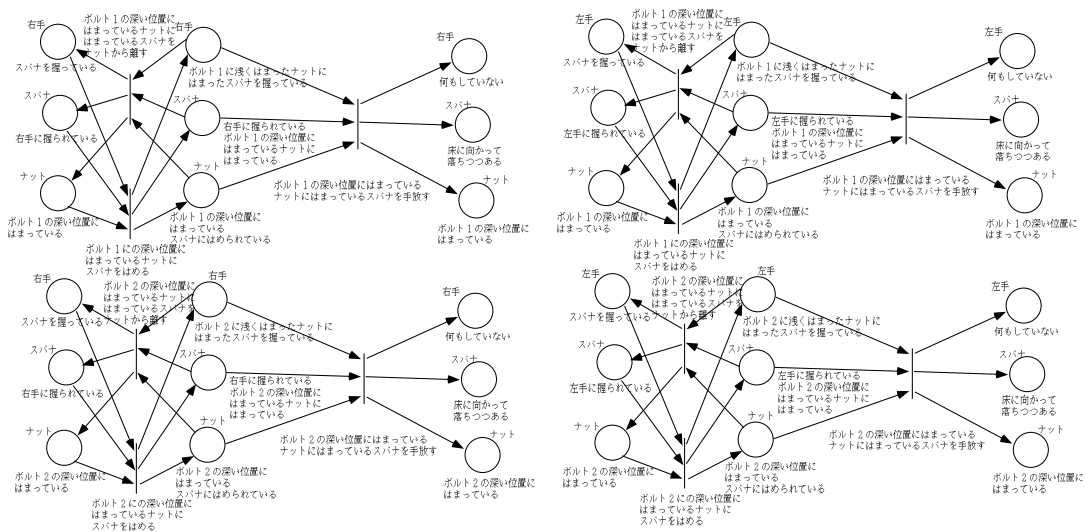


図 4.9: 手に握ったスパナをナットに当てる (その 1)

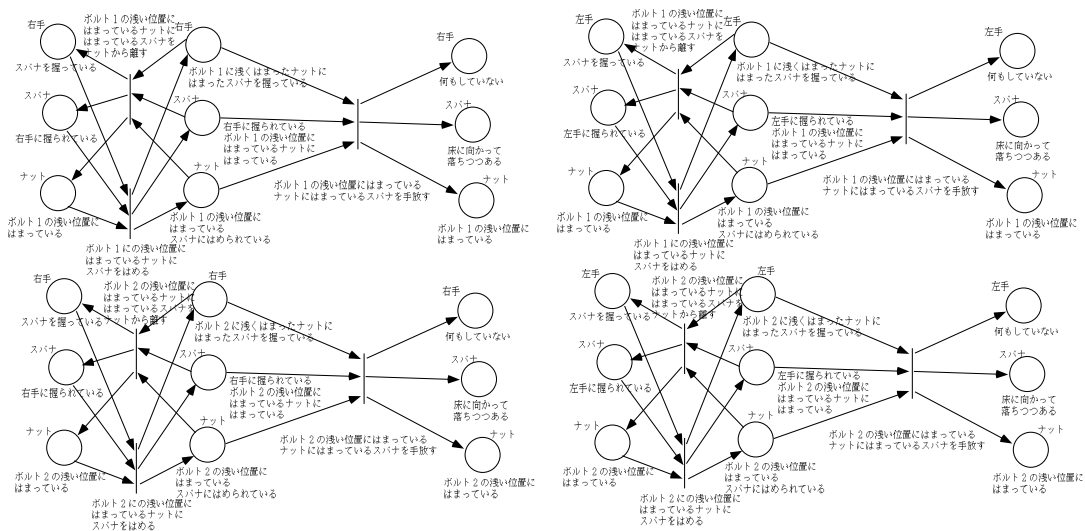


図 4.10: 手に握ったスパナをナットに当てる (その 2)

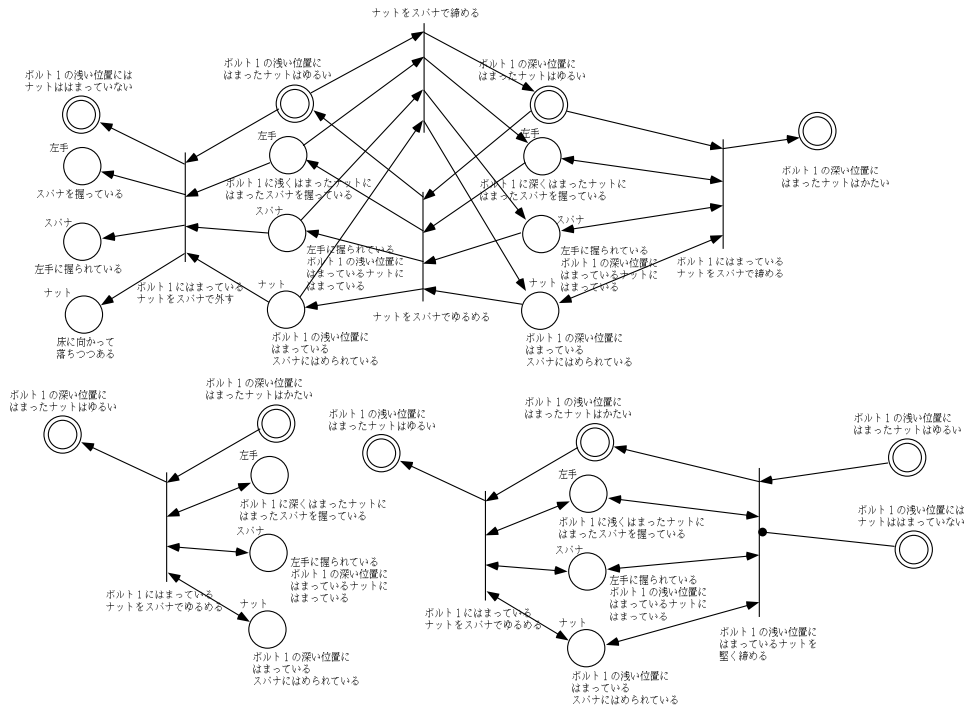


図 4.11: スパナでナットを締める, 緩める, 外す (その 1)

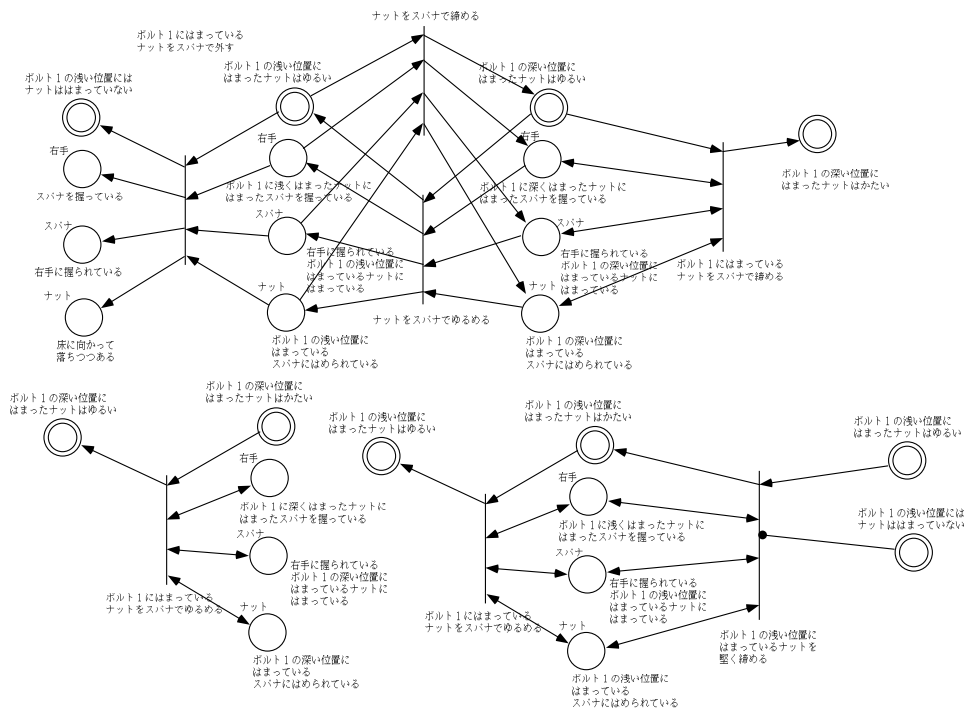


図 4.12: スパナでナットを締める, 緩める, 外す (その 2)

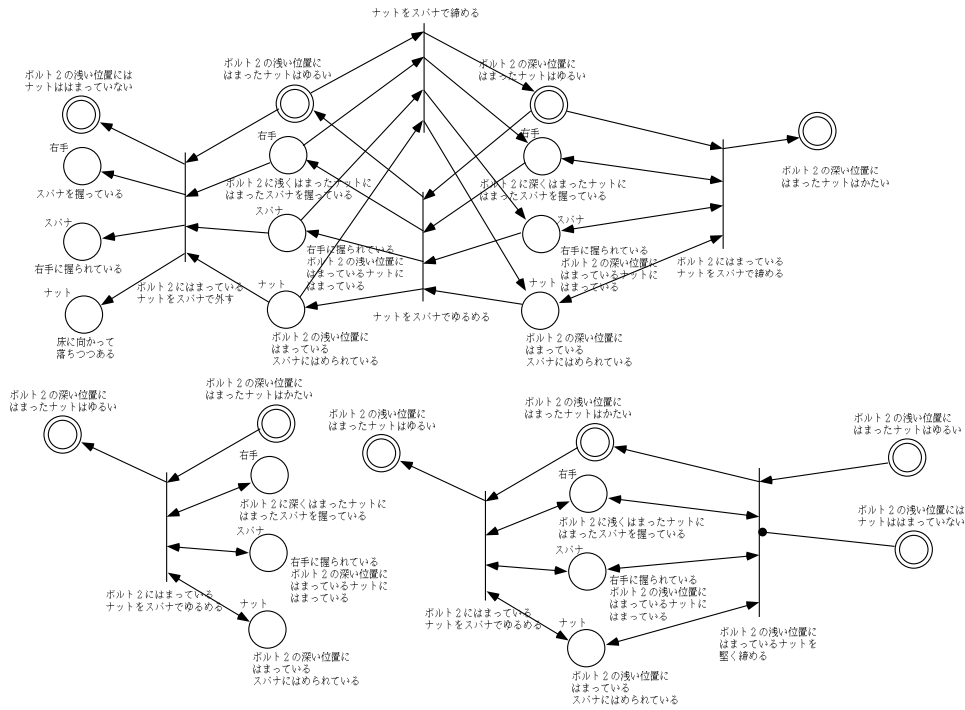


図 4.13: スパナでナットを締める, 緩める, 外す (その 3)

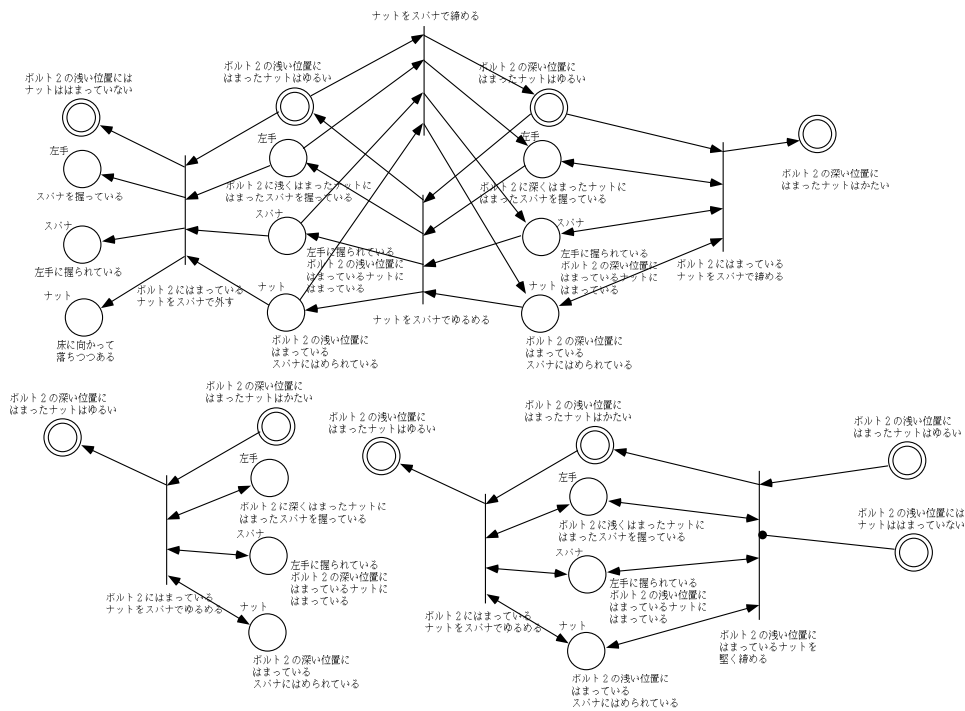


図 4.14 スパナでナットを締める, 緩める, 外す (その 4)

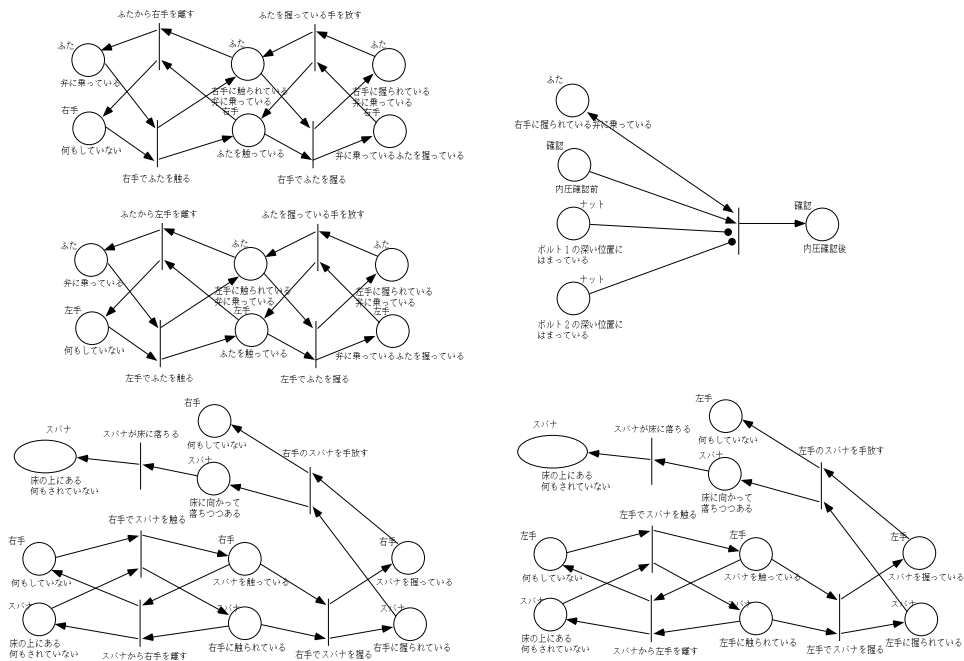


図 4.15: 蓋を握る, スパナを握る

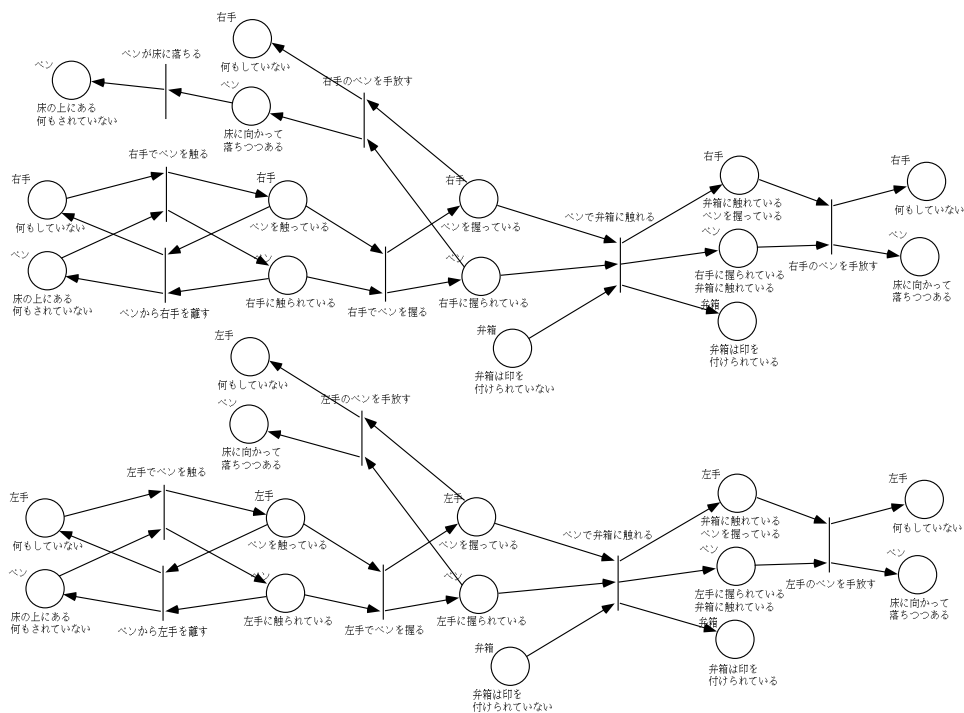


図 4.16: ペンで蓋に印を付ける

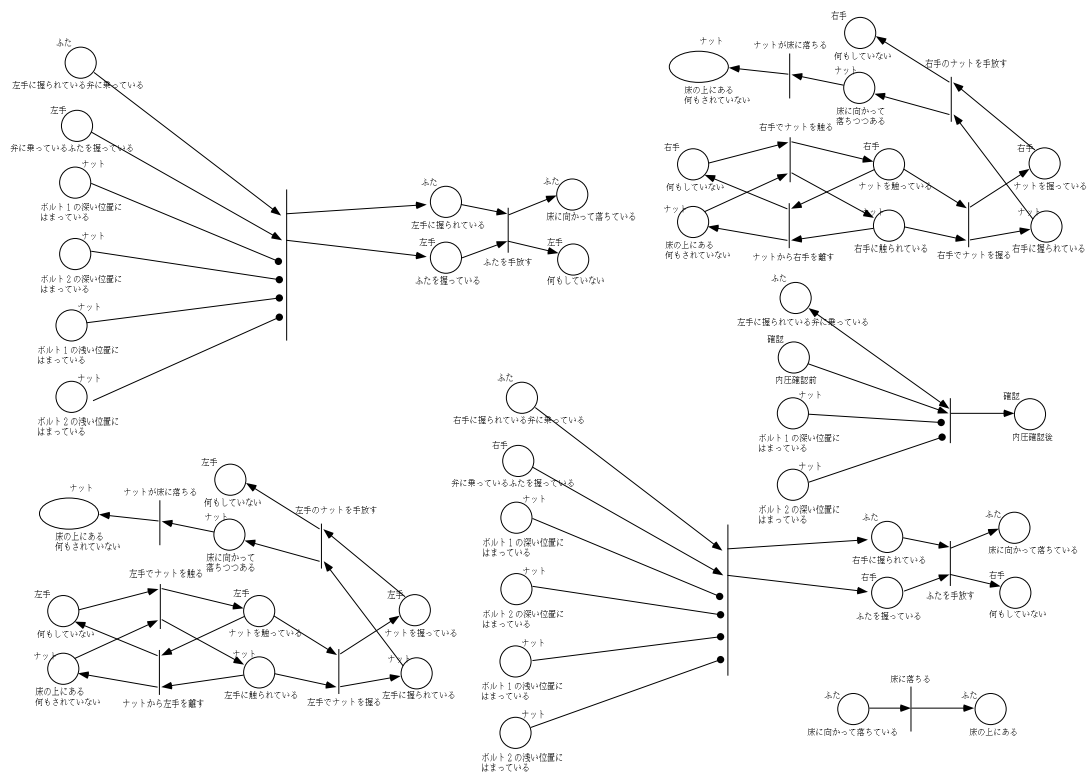


図 4.17: 内圧を確認する, ナットを握る

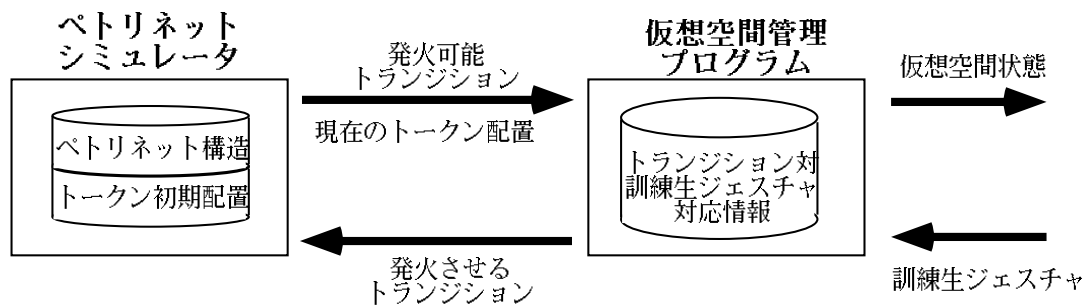


図 4.18: ペトリネットと仮想空間の関係

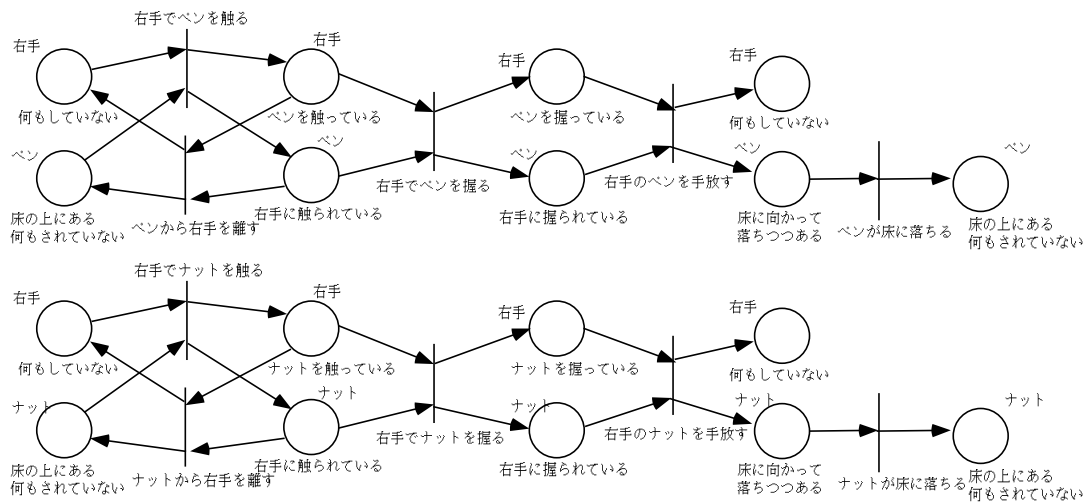


図 4.19: ペンとナットを手でつかむ作業のペトリネット

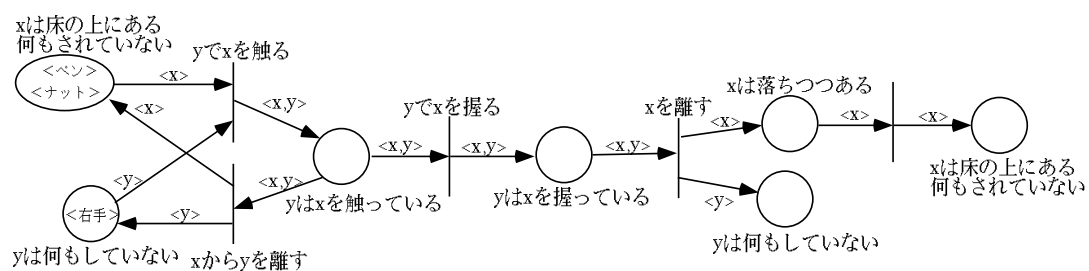


図 4.20: カラーペトリネットを用いたモデル化

表 5.1: 支援システムの構成

	計測部	情報提示部	情報処理部
訓練環境構築	3D Mouse 2D Mouse Key board	Dis play	Wo rks ta t i o n Ope n GL Mo t i f
	計測部	情報提示部	シミュレーション部
訓練実行	3 D Mo u s e	Di s p l a Cr y s t a l E y e s	Wo r k s t a t i o n Ope n GL

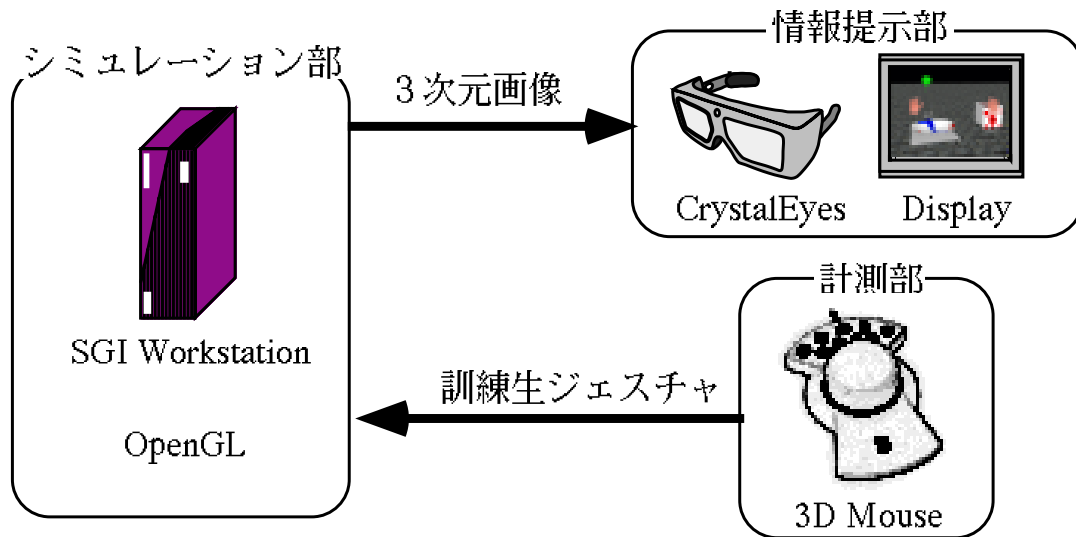


図 5.1 訓練環境構築時のシステム構成

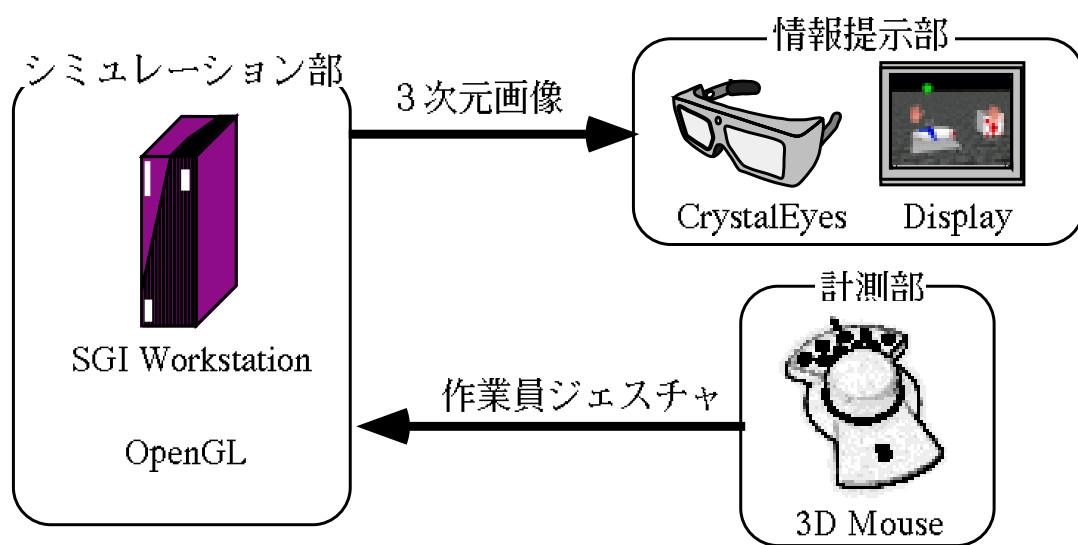


図 5.2: 訓練実行時のシステム構成

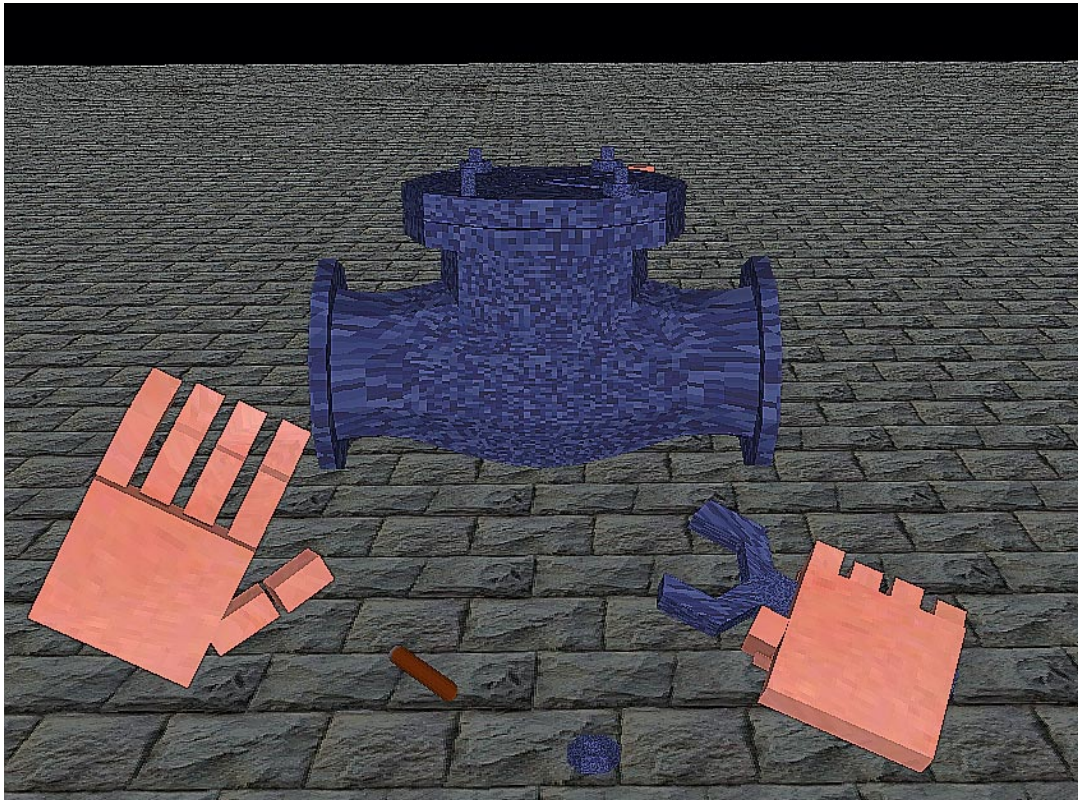


図 5.3: 訓練生に提示される映像の例

表 5.2: 訓練環境を構築するために入力する物体の外見に関する情報一覧

入力する情報の種類	指定方法
3次元形状データ	データを格納したファイル名を指定する
テクスチャデータ	データを格納したファイル名を指定する
描画方法	ポリゴン・ワイヤーフレーム・透明の 3つの項目から1つを選択する

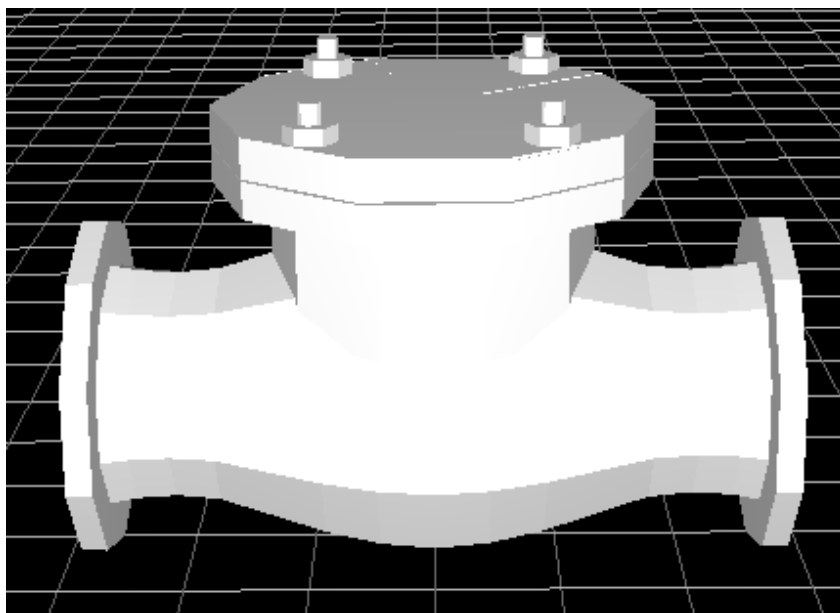


図 5.4: Polygon による描画

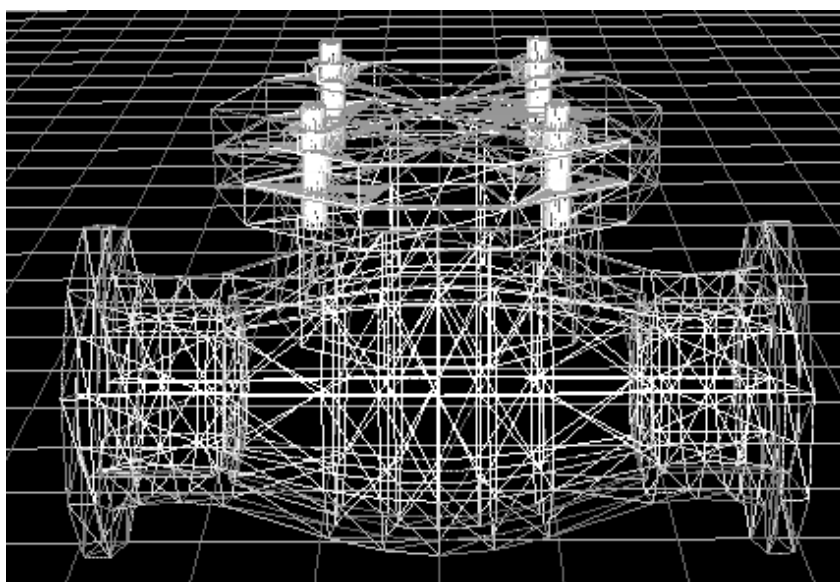


図 5.5: Wireframe による描画

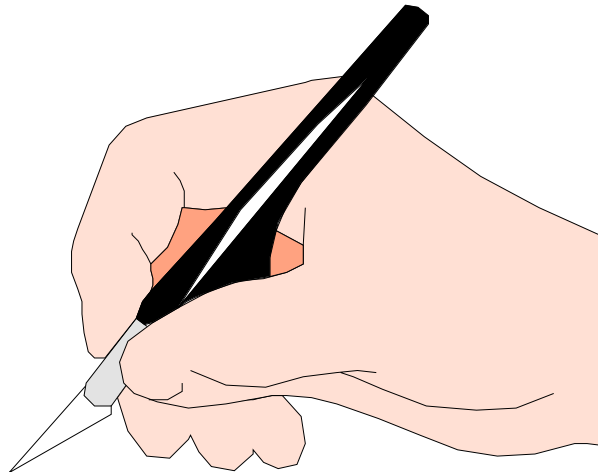


図 5.6: ペンの握り方

表 5.3: 訓練環境を構築するために入力する物体の動きに関する情報一覧

入力する情報の種類	指定方法
時間推移に従った位置 ・姿勢変化	位置・姿勢のパラメータを時間の関数として 表した式の計数を数値として入力する
マウスの入力に従った位置 ・姿勢変化	マウスからの入力値と、物体の位置・姿勢の パラメータとの関係を項目選択と数値入力で指示する
各状態における初期位置	物体の状態が変化したときの物体の位置と姿勢の 変化を項目選択と数値入力によって指定する

表 5.4: 「印を付ける訓練」を作成する場合に入力する物体名

入力する物体名
右手
ペン
ふた
印

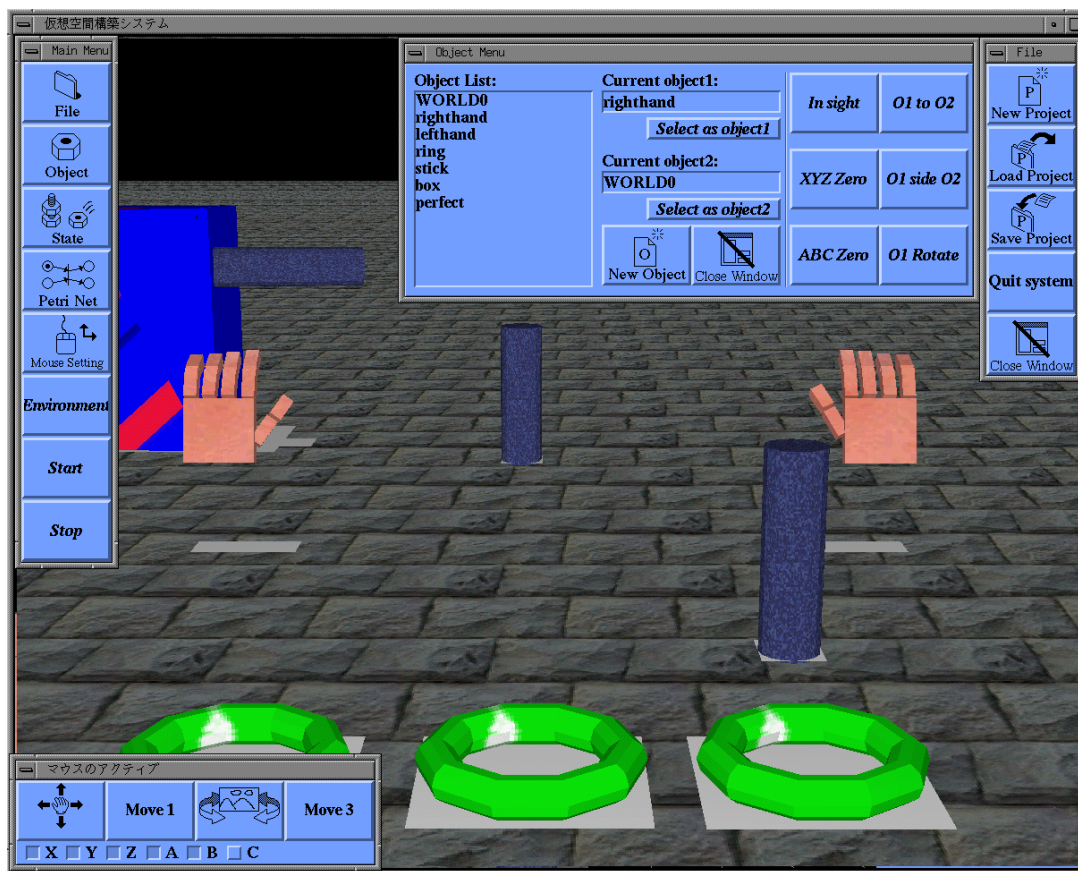


図 5.7: 訓練環境の構築時の画面例

表 5.5: ペンが取り得る状態

作成する状態名
何もされていない
右手に触られている
右手に握られている
蓋に接触している
落ちつつある

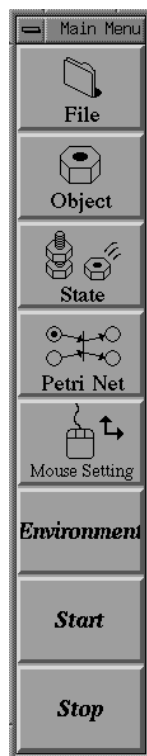


図 5.8: アイコンの例

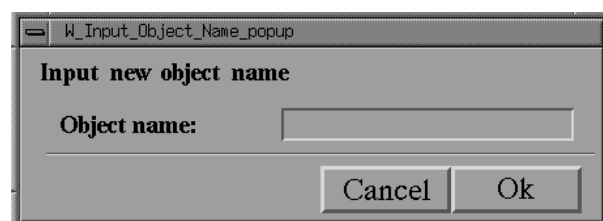


図 5.9: 新たに物体を登録する際のインタフェース



図 5.10: 状態を作成する際のインタフェース（外見に関する情報）

W_Input_Feature_Custom_popup

General	Initial Location	Define_Pararel	Define_Rotate
Initial locate type: ☒ New location ☐ Same as pre-state			
New location:		Use 3D mouse	
X-coordinate:		-1.4000	
Y-coordinate:		+1.2000	
Z-coordinate:		+6.0000	
A-coordinate:		+0.0000	
B-coordinate:		+3.1416	
C-coordinate:		+0.0000	
Base object:		WORLD0 Browse object lists	
☐ Use X ☐ Use Y ☐ Use Z ☐ Use A-B-C			
		Cancel Ok	

図 5.11: 状態を作成する際のインタフェース (初期位置に関する情報)

W_Input_Feature_Custom_popup

General	Initial Location	Define_Pararel	Define_Rotate
---------	------------------	----------------	---------------

Pararel movement definition

Coordinate base object:

☐ Move as time
 ☐ Move as other object

☐ X-Coordinate

☐ Min
 ☐ Max

Parameter Synchronize:

☐ Converted mouse input
 ☐ Right
 ☐ Left

Rate:

☐ Self parameter

☐ Y-Coordinate

☐ Min
 ☐ Max

Parameter Synchronize:

☐ Converted mouse input
 ☐ Right
 ☐ Left

Rate:

☐ Self parameter

☐ Z-Coordinate

☐ Min
 ☐ Max

Parameter Synchronize:

☐ Converted mouse input
 ☐ Right
 ☐ Left

Rate:

☐ Self parameter

When parameter will be out of definition,

☐ Fix all synchronized parameter
 ☐ Fix only self parameter

図 5.12: 状態を作成する際のインタフェース (平行移動に関する情報)

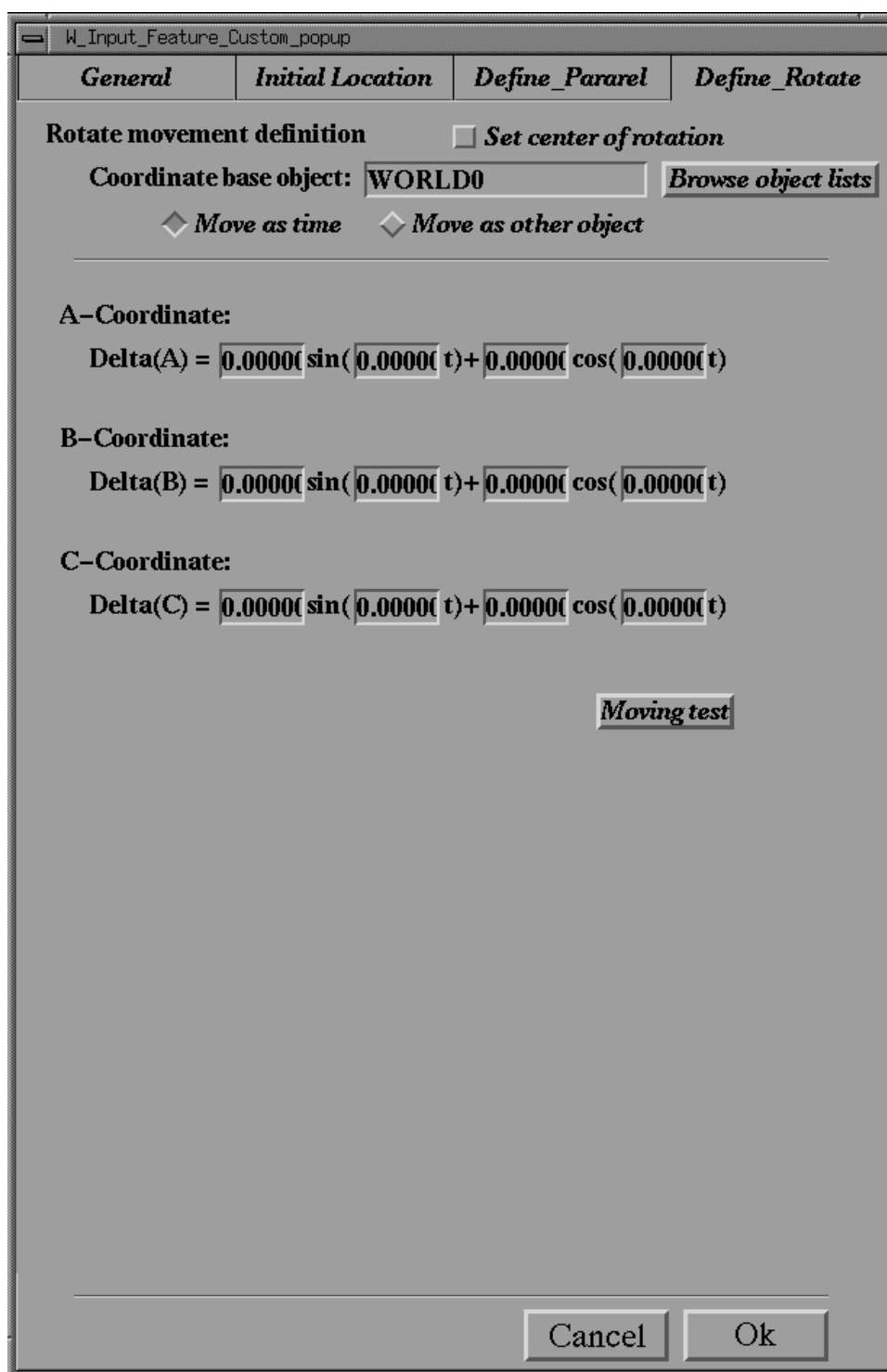


図 5.13: 状態を作成する際のインタフェース (時間に応じた移動法に関する情報)

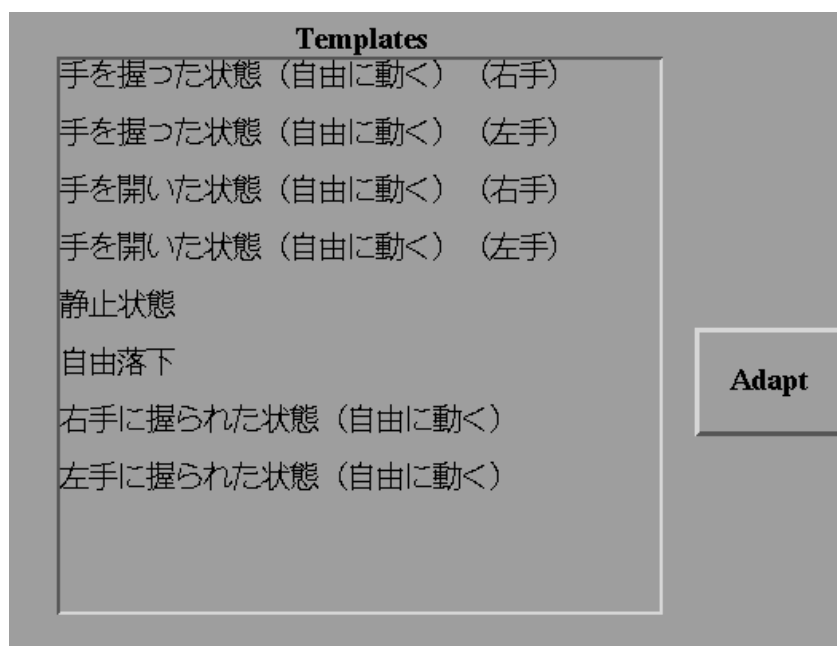


図 5.14: テンプレート機能

表 5.6: 自動的に入力されるパラメータ

パラメータ名	値	パラメータ名	値
状態名		形状ファイル	right-open. dxf
テクスチャファイル	hand. rgb	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
平行移動	有り	基準物体名	絶対座標
X 軸方向	移動可能	マウス入力比率	1.000
Y 軸方向	移動可能	マウス入力比率	1.000
Z 軸方向	移動可能	マウス入力比率	1.000
回転移動	有り	基準物体名	絶対座標
A 軸方向	移動可能	マウス入力比率	1.000
B 軸方向	移動可能	マウス入力比率	1.000
C 軸方向	移動可能	マウス入力比率	1.000

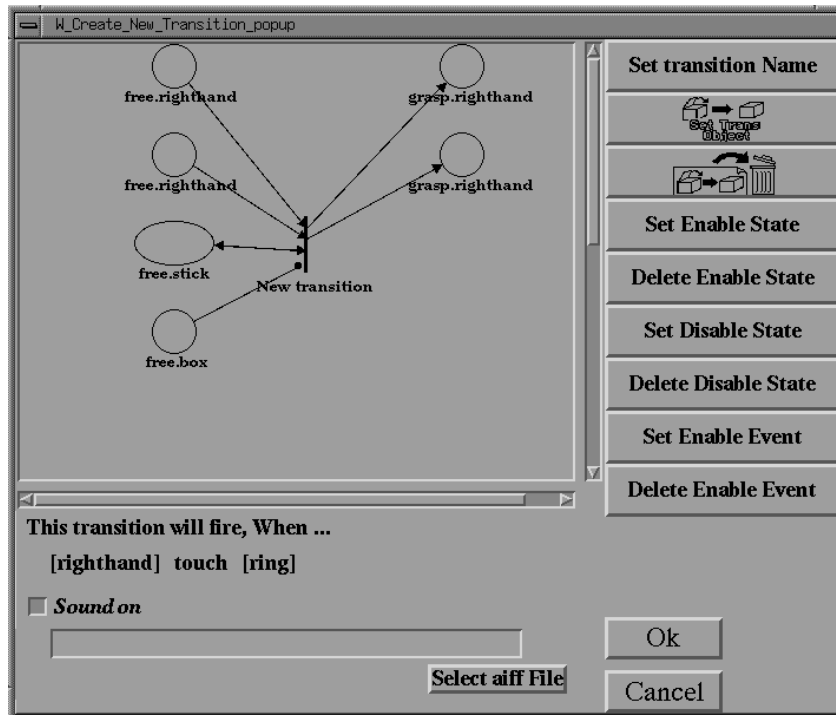


図 5.15: 新規トランジション作成画面

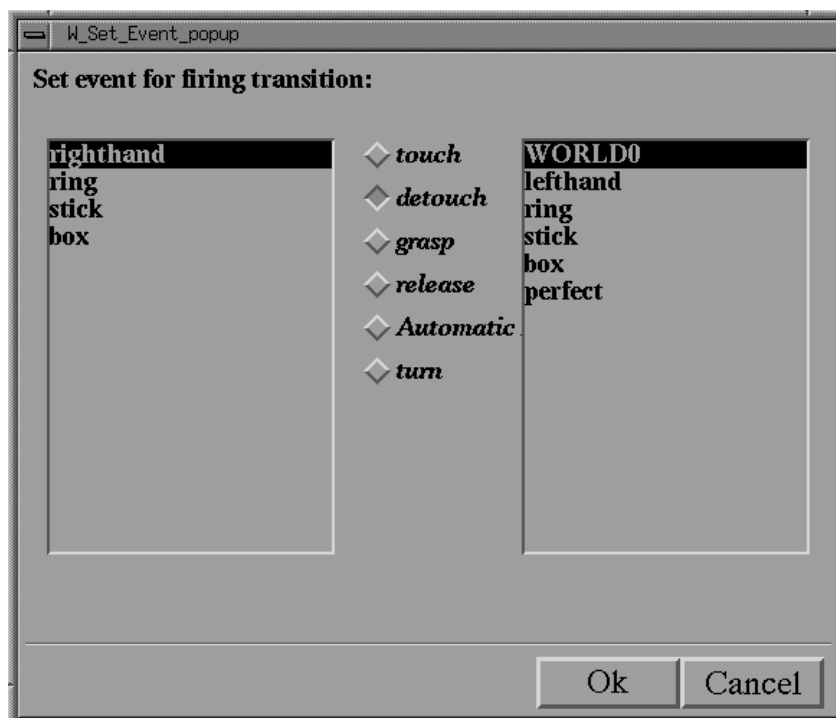


図 5.16: イベント設定画面

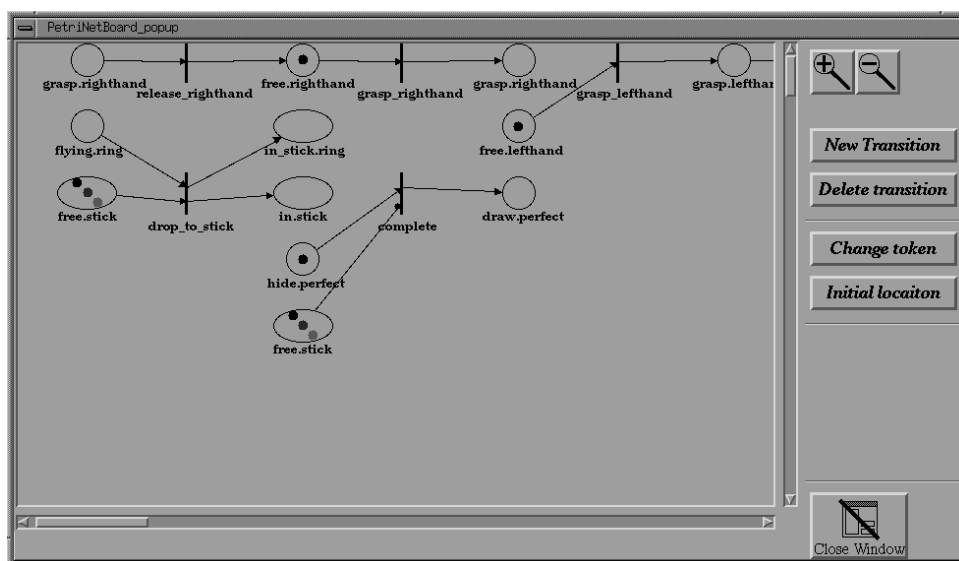


図 5.17: ペトリネット表示画面



図 5.18: ペトリネット拡大縮小ボタンとスライドバー

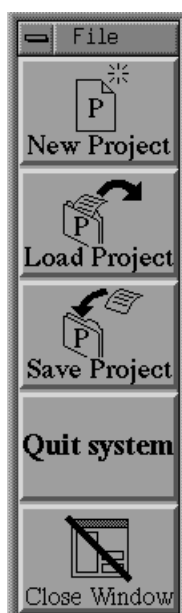


図 5.19: ファイル操作選択画面

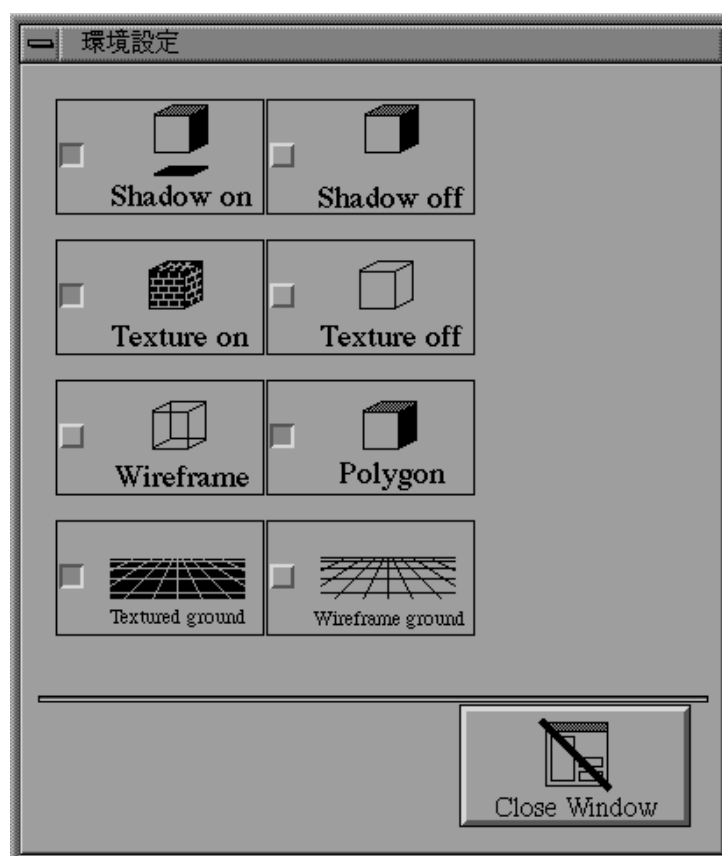


図 5.20: 環境設定画面

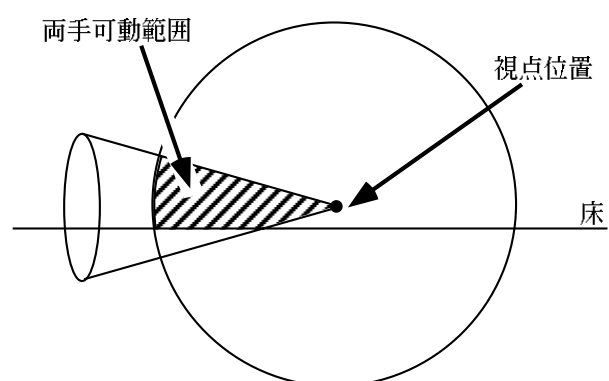


図 5.21: 両手の可動範囲

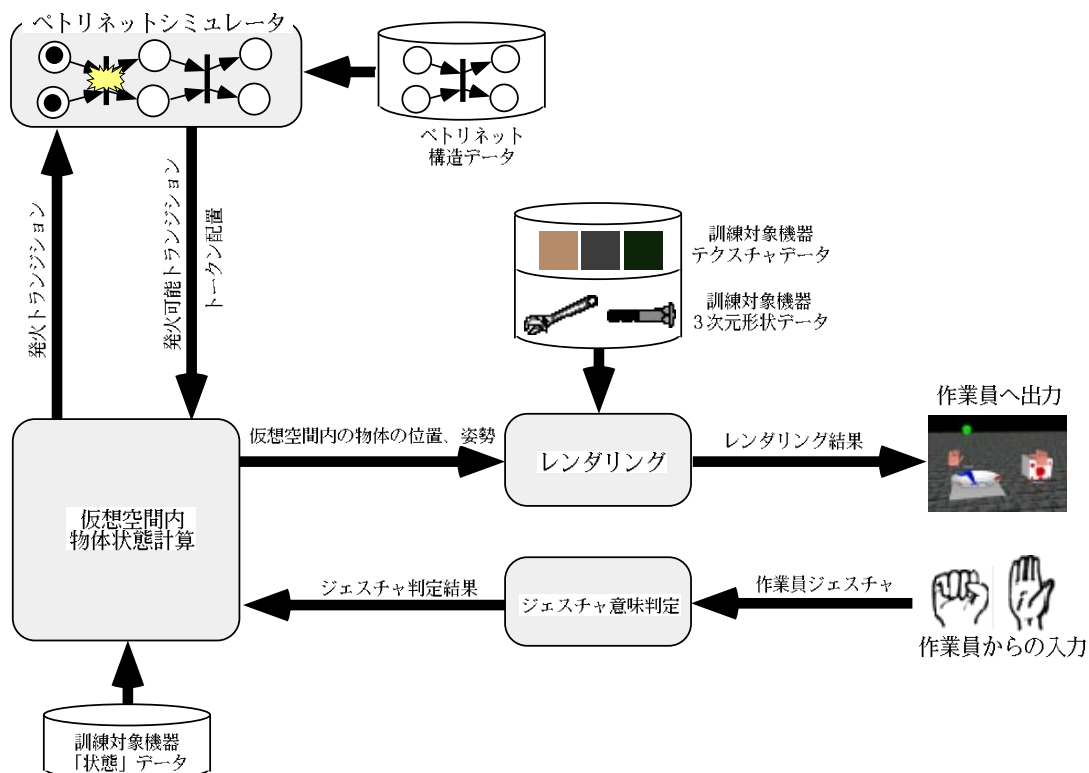


図 5.22: 訓練実行時の情報の流れ

表 5.7: 登録する物体名

righthand	left hand
VOLT1-BOTTOM	VOL T2- BOTTOM
VOL T3- BOTTOM	VOL T4- BOTTOM
VOL T1- MIDDLE	VOL T2- MIDDLE
VOL T3- MIDDLE	VOL T4- MIDDLE
VOL T1- UPPER	VOL T2- UPPER
VOL T3- UPPER	VOL T4- UPPER
VOL T1- TOP	VOL T2- TOP
VOL T3- TOP	VOL T4- TOP
BO D Y	LI D
NUT	PEN
SPANNA	MARK

表 5.8: 登録する状態 (右手)

free	grasp	touch-lid
grasp-lid	grasp-lid-in-m	grasp-lid-in-t
grasp-lid-in-space	touch-lid-in-space	touch-pen
grasp-pen	touch-sspana	grasp-sspana
sspana-volt1-up-tight	sspana-volt1-up-loose	sspana-volt1-top-loss
sspana-volt2-up-tight	sspana-volt2-up-loose	sspana-volt2-top-loss
sspana-volt3-up-tight	sspana-volt3-up-loose	sspana-volt3-top-loss
sspana-volt4-up-tight	sspana-volt4-up-loose	sspana-volt4-top-loss
touch-nut	grasp-nut	grasp-nut-volt1-top
grasp-nut-volt1-up	touch-nut-volt1-top	touch-nut-volt1-up
grasp-nut-volt2-top	grasp-nut-volt2-up	touch-nut-volt2-top
touch-nut-volt2-up	grasp-nut-volt3-top	grasp-nut-volt3-up
touch-nut-volt3-top	touch-nut-volt3-up	grasp-nut-volt4-top
grasp-nut-volt4-up	touch-nut-volt4-top	touch-nut-volt4-up

表 5.9: 登録する状態 (ボルト 1 ~ 4)

on-body

表 5.10: 登録する状態 (蓋)

on-body	on-body-touched
on-body-grasped	on-body-grasped-in-m
on-body-grasped-in-t	in-space-grasped
in-space-touched	on-floor
falling	dummy-on-body

表 5.11: 登録する状態 (ナット)

in-volt1-up-tigh	in-volt2-up-tigh	in-volt3-up-tigh
in-volt4-up-tigh	in-volt1-up-lose	in-volt2-up-lose
in-volt3-up-lose	in-volt4-up-lose	in-volt1-top-tigh
in-volt2-top-tigh	in-volt3-top-tigh	in-volt4-top-tigh
in-volt1-top-lose	in-volt2-top-lose	in-volt3-top-lose
in-volt4-top-lose	in-volt1-up-ti-spa	in-volt1-up-lo-spa
in-volt1-top-lo-spa	in-volt2-up-ti-spa	in-volt2-up-lo-spa
in-volt2-top-lo-spa	in-volt3-up-ti-spa	in-volt3-up-lo-spa
in-volt3-top-lo-spa	in-volt4-up-ti-spa	in-volt4-up-lo-spa
in-volt4-top-lo-spa	falling	on-floor
touched	grasped	in-volt1-top-lo-hand
in-volt1-up-lo-hand	in-volt2-top-lo-hand	in-volt2-up-lo-hand
in-volt3-top-lo-hand	in-volt3-up-lo-hand	in-volt4-top-lo-hand
in-volt4-up-lo-hand		

表 5.12: 登録する状態 (ペン)

free	touched	grasped
falling		

表 5.13: 登録する状態 (スパナ)

free	touched	grasped
falling	fit-volt1-up-tigh	fit-volt1-up-lose
fit-volt1-top-lose	fit-volt2-up-tigh	fit-volt2-up-lose
fit-volt2-top-lose	fit-volt3-up-tigh	fit-volt3-up-lose
fit-volt3-top-lose	fit-volt4-up-tigh	fit-volt4-up-lose
fit-volt4-top-lose		

表 5.14: 登録する状態 (印)

un-DRAW	DRAW
---------	------

表 5.15: 状態におけるパラメータ (手に握られたペン)

パラメータ名	値	パラメータ名	値
状態名	grasped	形状ファイル	PEN.dxf
テクスチャファイル	PEN.rgb	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
初期位置	有り	基準物体名	righthand
位置	-0.075, 0.089, 0.000	姿勢	0.937, 6.017, 1.744
平行移動	有り	基準物体名	righthand
X 軸方向	移動可能	マウス入力比率	1.000
Y 軸方向	移動可能	マウス入力比率	1.000
Z 軸方向	移動可能	マウス入力比率	1.000
回転移動	有り	基準物体名	righthand
A 軸方向	移動可能	マウス入力比率	1.000
B 軸方向	移動可能	マウス入力比率	1.000
C 軸方向	移動可能	マウス入力比率	1.000

表 5.16: 状態におけるパラメータ (ボルトを中心に回転するナット)

パラメータ名	値	パラメータ名	値
状態名	in-volt1-top	形状ファイル	NUT.dxf
テクスチャファイル	NUT.rgb	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
初期位置	有り	基準物体名	VOLT1-TOP
位置	0.000, 0.000, 0.000	姿勢	0.000, 0.000, 0.000
平行移動	有り	基準物体名	VOLT1-TOP
X 軸方向	移動不可能	自己パラメータ B 軸比率	0.050
Y 軸方向	移動可能		
Z 軸方向	移動不可能		
回転移動	有り	基準物体名	VOLT1-TOP
A 軸方向	移動不可能		
B 軸方向	移動可能	マウス入力比率	1.000
C 軸方向	移動不可能		

表 5.17: 状態におけるパラメータ（落下するナット）

パラメータ名	値	パラメータ名	値
状態名	falling	形状ファイル	NUT.dxf
テクスチャファイル	NTrgb	状態タイプ	Normal
描き方	Pdygn	サイズ	0.3
初期位置	無し	基準物体名	絶対座標
平行移動	時間推移	式の係数	50,0,0,0,0,0
X 軸方向	移動不可		
Y 軸方向	移動可能		
Z 軸方向	移動不可		
回転移動	不可		

表 5.18: 初期トークンの配置

プレース名	数	プレース名	数	プレース名	数
right hardfree	1	left hardfree	1	BODYfree	1
VOL T-BOTTOMfree	1	VOL T-BOTTOMfree	1	VOL T-BOTTOMfree	1
VOL T-BOTTOMfree	1	VOL T-MIDDLEfree	1	VOL T-MIDDLEfree	1
VOL T-MIDDLEfree	1	VOL T-MIDDLEfree	1	VOL T-UPPERfree	1
VOL T-UPPERfree	1	VOL T-UPPERfree	1	VOL T-UPPERfree	1
VOL T-TOP .free	1	VOL T-TOP .free	1	VOL T-TOP .free	1
VOL T-TOP .free	1	HEADbody	1	NUTinv dt1-up-tight	1
NUTinv dt2-up-tight	1	NUTinv dt3-up-tight	1	NUTinv dt4-up-tight	1
BNfree	1	SPANNfree	1	MK.u+DA W	1

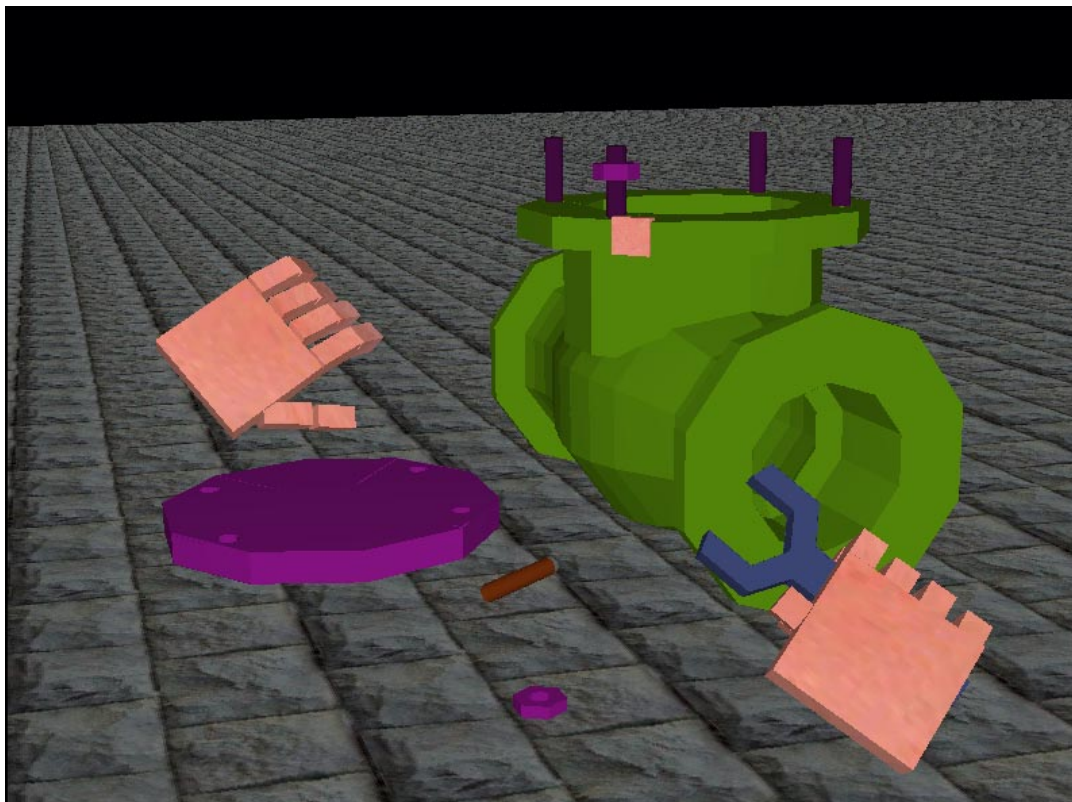


図 5.23: 訓練環境の構築例

表 5.19: 登録する物体名

righthand	left hand
motor-body	motor-shaft
switch1	switch2
switch-body	

表 5.20:登録する状態 (hand,motor-body, switch-body, switch)

何もしていない

表 5.21:登録する状態 (shaft)

回転している	回転していない
--------	---------

表 5.22:状態におけるパラメータ (何もしていない)

パラメータ名	値	パラメータ名	値
状態名	free	形状ファイル	各オブジェクト指定
テクスチャファイル	各オブジェクト指定	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
初期位置	無し	回転移動	不可
平行移動	不可		

表 5.23:状態におけるパラメータ (回転するシャフト)

パラメータ名	値	パラメータ名	値
状態名	rotate	形状ファイル	SHAFT.dxf
テクスチャファイル	SHAFT.rgb	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
初期位置	無し	平行移動	無し
回転移動	有り	基準物体名	SHAFT
A 軸方向	移動不可		
B 軸方向	移動不可		
C 軸方向	移動可能	時間移動比率	100

表 5.24: 状態におけるパラメータ（静止しているシャフト）

パラメータ名	値	パラメータ名	値
状態名	rotate	形状ファイル	SHAFT.dxf
テクスチャファイル	SHAFT.rgb	状態タイプ	Normal
描き方	Polygon	サイズ	0.3
初期位置	無し	平行移動	無し
回転移動	無し		

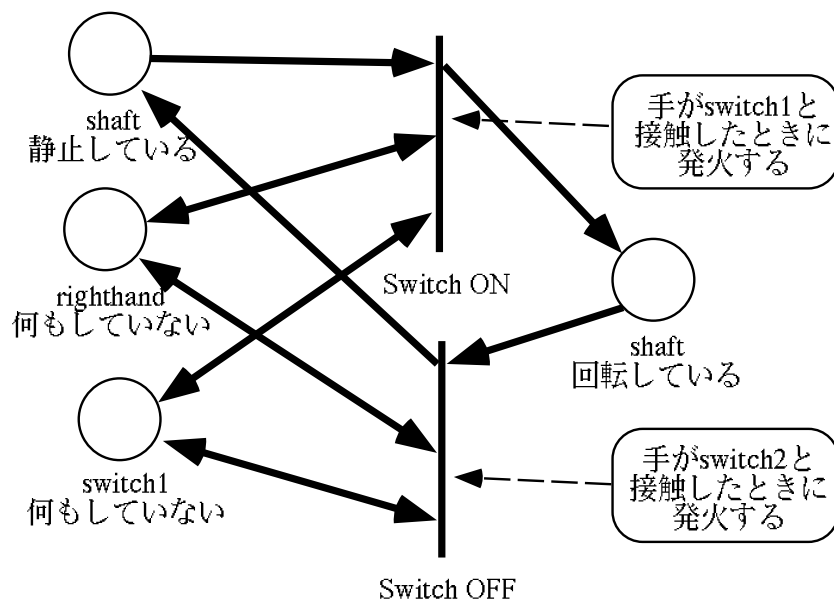


図 5.24: 構築すべきペトリネット

表 5.25: 支援環境を利用した感想

支援システムを利用して、最初はどのボタンが、どの機能に対応しているのかが分からなかったが、使っている内に分かるようになった。
オブジェクトの位置を合わせる作業が難しかった。
使っている内に、もっと凝った仮想空間を作りたいと思うようになった。
システムを使い込むと、もっと自由に仮想空間を構築できるようになると思う。

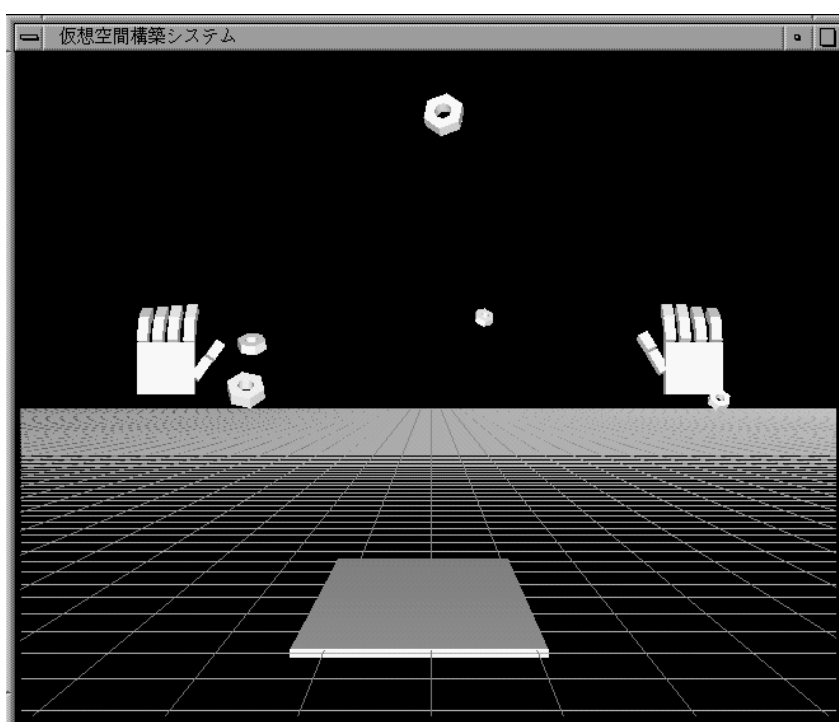


図 5.25: 5つのナットを空中に配置した仮想空間

表 5.26: 実験の環境

環境 1	3つの補助機能のいずれも用いない
環境 2	影のみ用いる
環境 3	効果音のみ用いる
環境 4	立体視のみ用いる
環境 5	全ての補助機能を同時に用いる

表 5.27: 実験 2 の結果 (被験者 A)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5	環境 1
1 回目	135.3	61.2	83.0	55.9	36.9	143.2
2 回目	96.6	37.0	50.9	62.6	31.3	104.9
3 回目	108.2	70.7	43.9	46.2	37.9	74.7
4 回目	89.4	42.5	61.2	37.4	47.3	68.9
5 回目	97.7	51.8	80.6	35.7	32.1	59.8
平均	105.4	52.7	63.9	47.6	37.1	90.3

表 5.28: 実験 2 の結果 (被験者 B)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5	環境 1
1 回目	156.7	50.2	56.9	59.7	30.6	84.2
2 回目	86.6	59.5	48.8	42.3	31.2	51.5
3 回目	72.5	44.5	55.3	44.2	33.9	80.6
4 回目	72.4	64.0	51.3	43.4	55.2	72.2
5 回目	54.5	41.3	62.0	36.5	33.0	69.2
平均	88.5	51.9	54.9	45.2	36.8	71.5

表 5.29: 実験 2 の結果 (被験者 C)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5	環境 1
1 回目	63.1	38.8	42.3	51.7	21.4	25.9
2 回目	34.4	27.3	42.9	36.2	37.3	34.2
3 回目	48.8	37.3	32.1	27.9	27.7	35.9
4 回目	43.6	23.8	32.3	33.9	20.0	53.7
5 回目	55.0	27.0	31.3	34.4	31.2	43.4
平均	49.0	30.8	36.2	36.8	27.5	38.6

表 5.30: 実験 2 の結果 (被験者 D)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5
1 回目	43.7	22.0	25.2	28.7	14.1
2 回目	34.9	15.9	23.8	22.4	17.4
3 回目	27.7	20.0	22.7	22.8	18.5
4 回目	33.0	19.2	23.3	29.1	17.7
5 回目	22.9	22.1	25.2	20.6	16.3
平均	32.4	19.8	24.0	24.7	16.8

表 5.31: 実験 2 の結果 (被験者 E)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5
1 回目	37.4	27.2	31.1	32.0	26.3
2 回目	46.3	27.3	40.3	31.3	20.1
3 回目	25.2	27.3	35.7	30.8	19.2
4 回目	29.0	27.4	26.9	24.2	21.0
5 回目	26.3	25.2	28.4	26.2	22.0
平均	32.8	26.9	32.5	28.9	21.7

表 5.32: 実験 2 の結果 (被験者 F)(単位は秒)

実験回数	環境 1	環境 2	環境 3	環境 4	環境 5
1 回目	49.6	41.9	41.6	43.5	35.7
2 回目	50.2	38.1	28.8	35.2	23.6
3 回目	95.6	35.2	25.5	51.2	20.7
4 回目	34.5	32.5	34.4	45.0	26.0
5 回目	50.3	21.6	31.2	32.2	18.9
平均	56.1	33.9	32.3	41.4	25.0

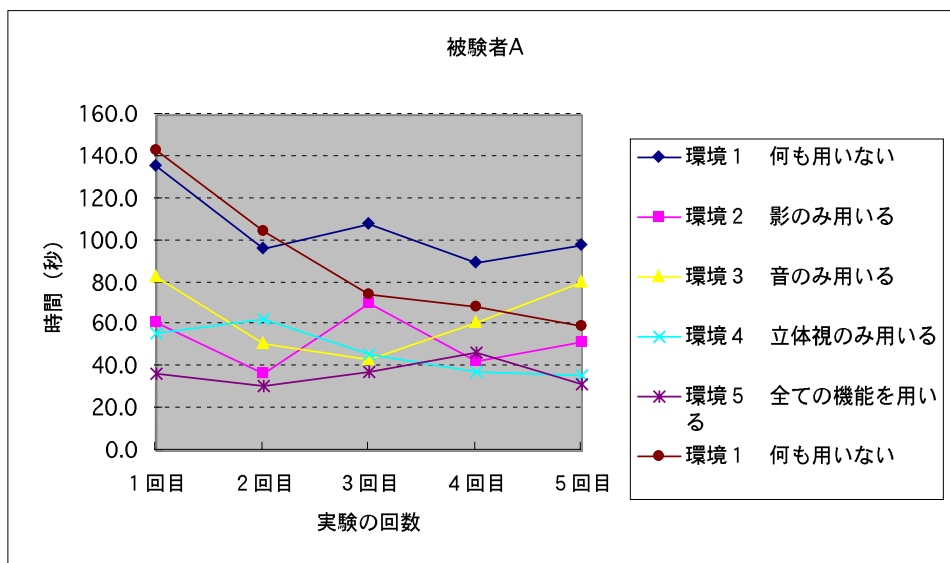


図 5.26: 被験者 A

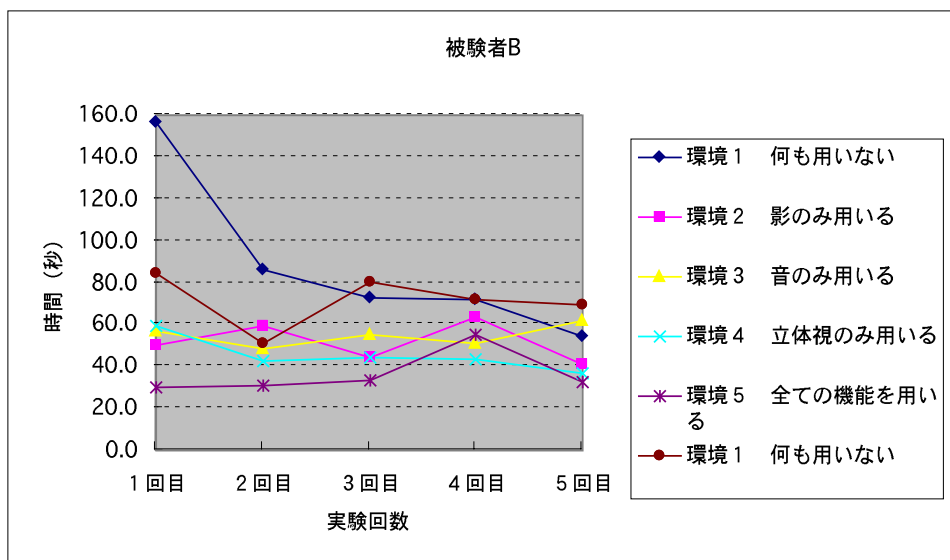


図 5.27: 被験者 B

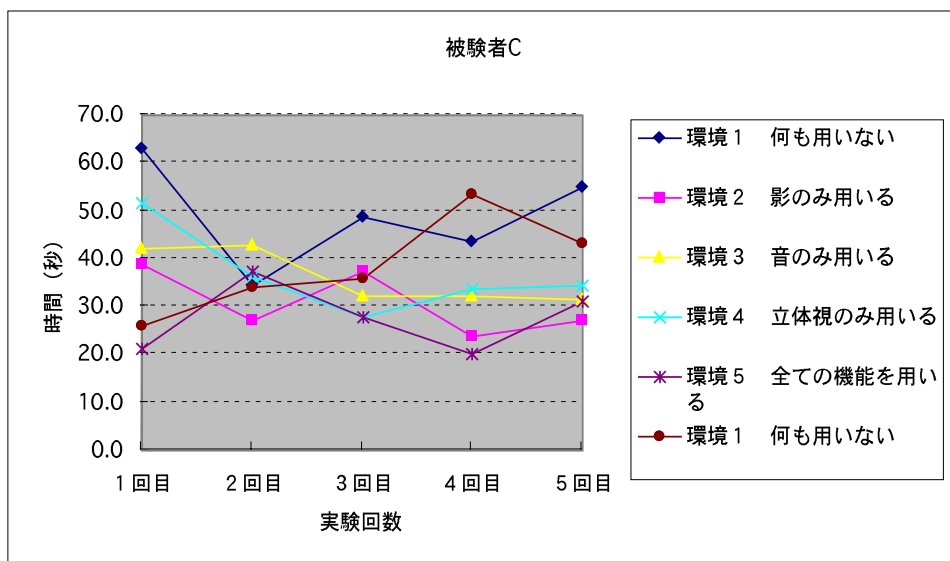


図 5.28: 被験者 C

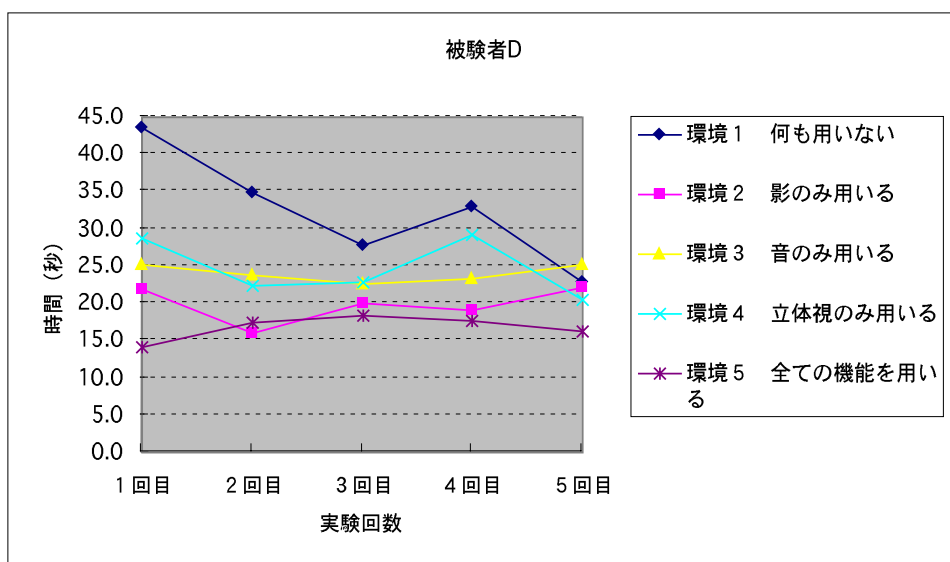


図 5.29: 被験者 D

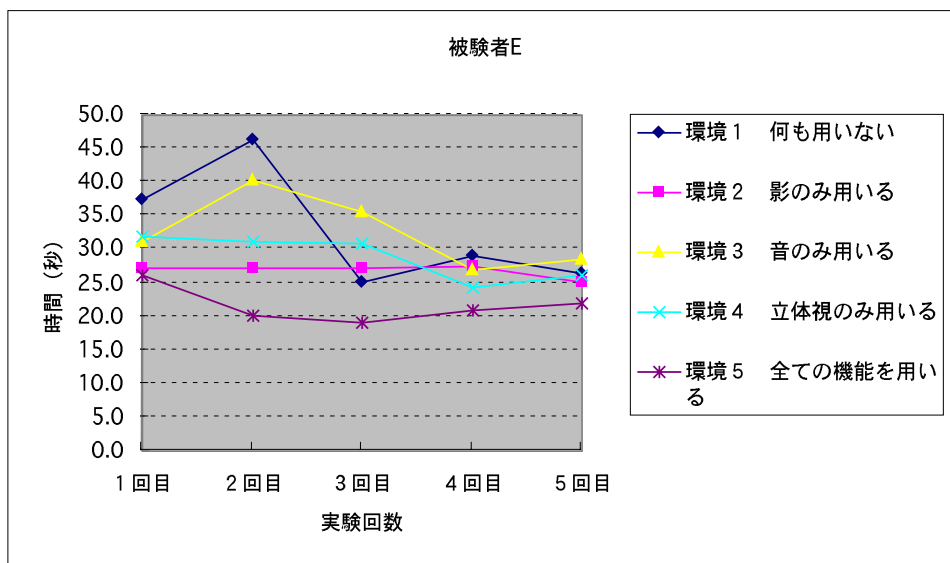


図 5.30: 被験者 E

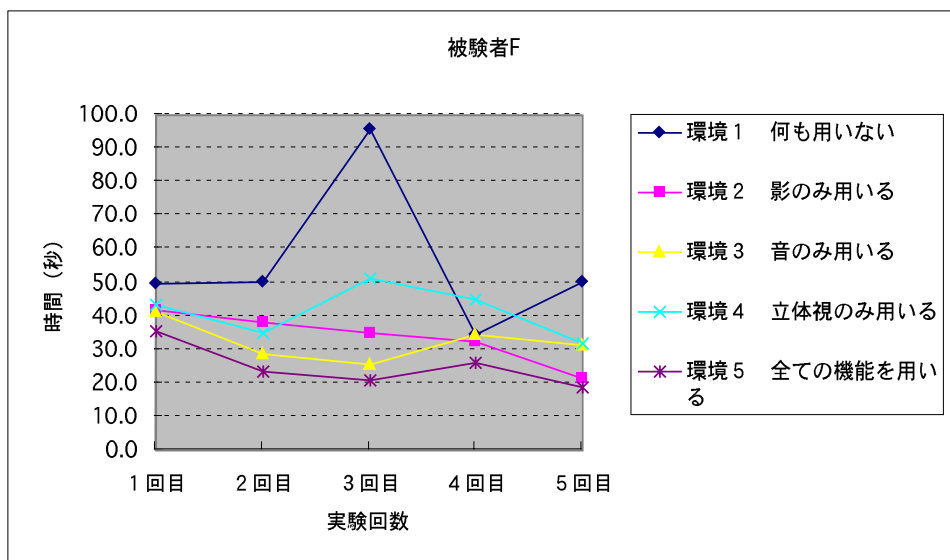


図 5.31: 被験者 F

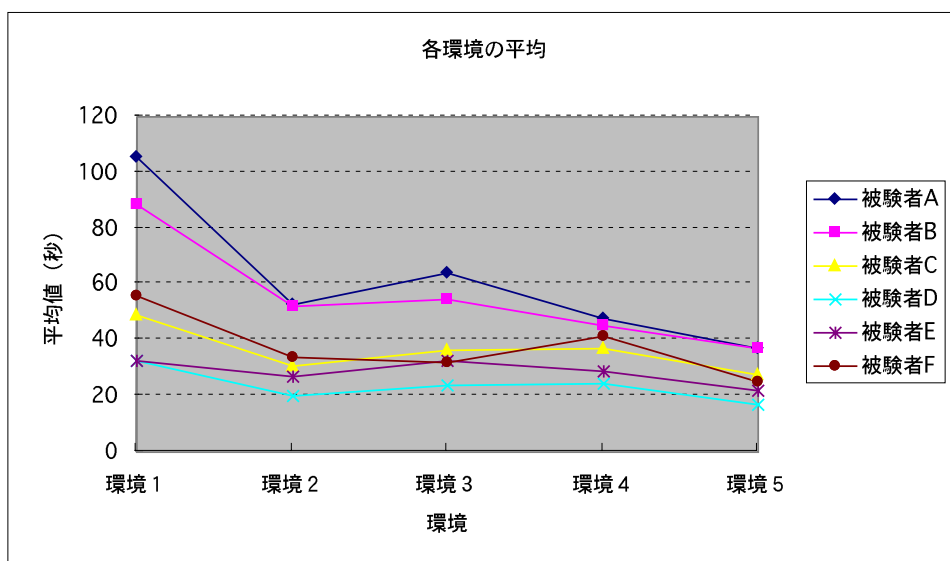


図 5.32: 各環境における平均値

表 5.33: 各環境の主観的な使いやすさの順位

順位	被験者 A	被験者 B	被験者 C	被験者 D	被験者 E	被験者 F
1	環境 5	環境 5	環境 5	環境 5	環境 5	環境 5
2	環境 2	環境 2	環境 3	環境 2	環境 4	環境 3
3	環境 3	環境 4	環境 2	環境 4	環境 2	環境 2
4	環境 4	環境 3	環境 4	環境 3	環境 3	環境 4
5	環境 1	環境 1	環境 1	環境 1	環境 1	環境 1

表 5.34: 訓練環境を使用した感想

被験者	感想
被験者 A	物を落とすときは影がすごく便利 ゲーム感覚で焦ってしまう 全部の機能を用いた物は、最初は困惑するが後になればやりやすい
被験者 B	立体視と影の両方を利用するのがやりやすい 3次元マウスを使い続けると手がすりそうになる
被験者 C	何もなしはかなり難しい 遠くは影が見にくい 音よりは絵に頼ってしまう 立体視では大体の位置は分かるが、微妙なことは判断できない 立体視と影の両方を利用するのが有効である 立体視で大体の位置を知り、影で確信する
被験者 D	立体視は目が非常に疲れる
被験者 F	急に基準がなくなると困惑する かんが当たった時が何回もあった

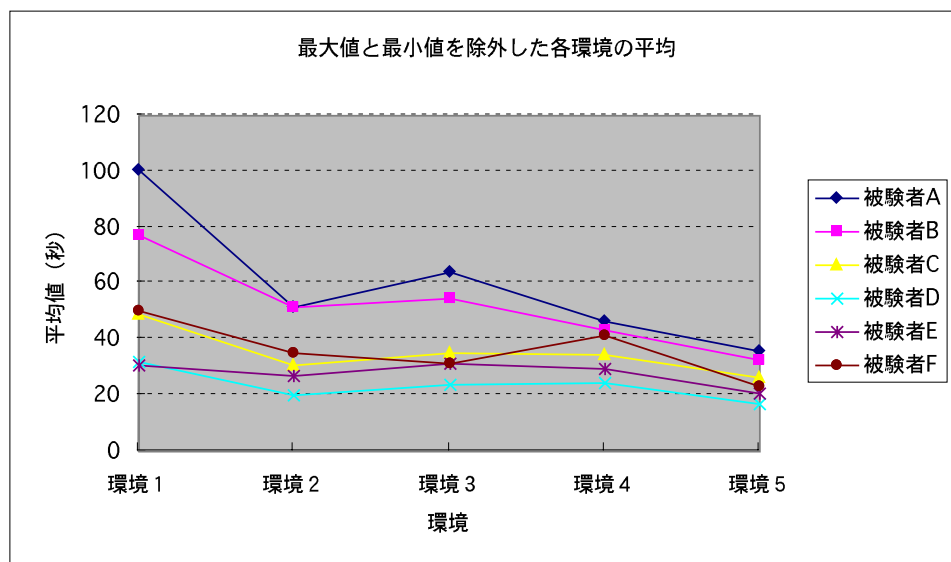


図 5.33 各環境における平均値 (最大値、最小値を除外)

付録 A ハードウェア詳細

A.1 Magellan

Magellan は図 A.1に示すように 1 つのパックと 9 つのボタンで構成される。パックはバネにより制御され、位置と回転の 6 自由度 ($X, Y, Z, Yaw, Roll, Pitch$) をコントロールすることができる。このデバイスは、RS-232C インタフェースに接続することができるため、ワークステーションに限らず、パーソナルコンピュータにも接続して使うことができる。また、デバイスを制御するためのコマンド類も公開されており、容易にアプリケーションに利用することができる。

A2 CrystalEyes

CrystalEyes は計算機によって生成された画像やビデオ映像の立体視を可能にする製品である。ここでその立体視の原理を簡単に説明する。立体視の原理を図 A2 に示す。CrystalEyes のレンズの部分には液晶シャッターが付いていて、その液晶シャッターが 10Hz の速さで、閉じたり開いたりしている。もちろんこの速度では、人間の目には目の前が真っ暗になる感じはない。むしろまばたきと同じことである。これを左右の眼に、交互に点滅させる。これに合わせて、映像も左眼、右眼用と切り替えながら映す。すると本人には気づかないうちに、片目ごとにそれぞれの映像を見て、それらを同時に見ているものと脳に感じ、立体的に見える。

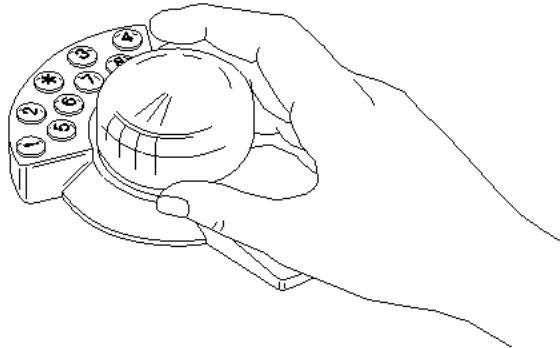


図 A.1: Magellan

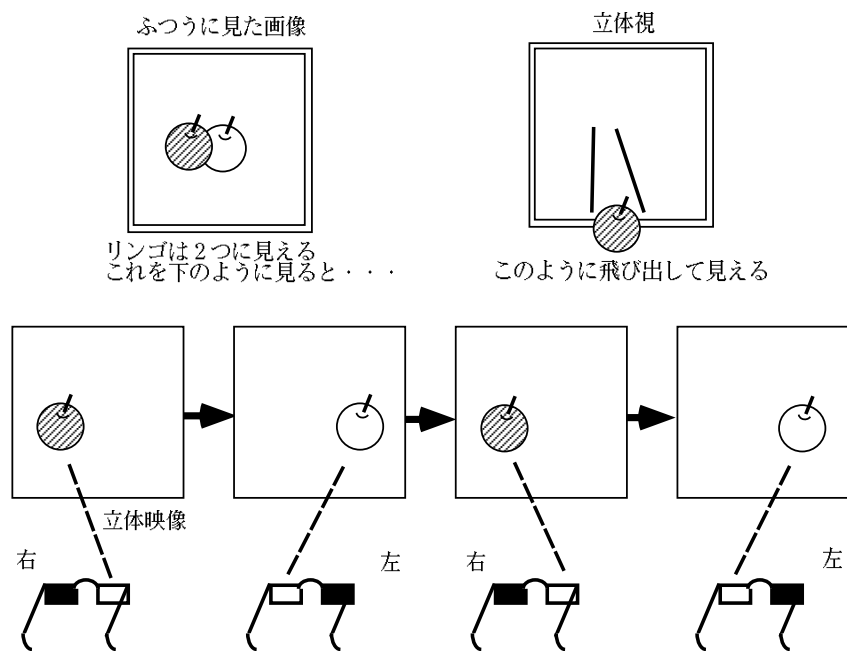


図 A2: CrystalEyes による立体視の原理

付録 B ペトリネットの自動実演機能への適用

本章では、自動実演機能を実現するために解決すべき問題の一つである、巨大なペトリネットの発火系列の検索法について検討する。ここで自動実演機能とは以下の特徴を持った機能であるとする。

- 特徴 1 自動実演の機能は、両手に何も持っていない時に限り利用することができる。
- 特徴 2 正しい手順で作業を行っている場合に自動実演機能を選ぶと、その次の訓練を進めるための作業を自動的に実演する。
- 特徴 3 間違った手順で作業を行っている場合に自動実演機能を選ぶと、元に戻って、その間違いを修正する作業を自動的に実演する。

特徴 1 により、発火系列の検索を行う時の初期状態には、必ず、「手に何も持っていない状態」を表すスペースにトークンが存在することになる。

B.1 発火系列を検索する方針

4.4節で述べたように、ペトリネットの発火系列の検索法には、接続行列を用いる方法と、可達木を作成する方法が提案されている。しかし、接続行列を用いる方法では、トランジションの発火回数に関する情報は得ることができるが、発火の順番に関する情報は得ることができないので、本研究では可達木を用いて発火系列の検索を行う方法を採用することにする。しかし、巨大なペトリネットの可達木を完全に作成することは、計算量の関係から非常に困難である。また、作成したペトリネットの完全な可達木を作成し、それを基に発火系列を求めると、自動実演を行う為には必要でない発火系列も求めることになり無駄である。自動実演を行う為に必要な発火系列は、最も工程数の少ない（すなわち発火させるトランジションの総数が最も小さい）発火系列である場合がほとんどである。例えば、逆止弁の分解作業をモデル化したペトリネットの発火系列を、完全な可達木を作成して検索した場合、「ペンで蓋に印を付ける作業」を行う発火系列だけを取り出すと、表 B.1に示すような作業に相当する発火系列が検索されることが予想される。これら求められた発火系列の内、「ペンで蓋に印を付ける作業」の自動実演を行うために利用する発火系列は、「発火系列 1」だけであり、その他の発火系列は作業に無駄があるため、用いないと考えるのが自然である。このように無駄な発火系列を検索してしまうのは、作業の目的に直接関係のない物体であっても右手と左手で同時に操作す

表 B.1: ペンで蓋に印を付ける作業に対応する発火系列 (一部)

発火系列 1	ペンを右手でさわる・ペンを右手で握る・ペンで蓋に印を付ける ペンを手放す
発火系列 2	ペンを右手でさわる・ペンを右手で握る・ナットを左手で触る ナットから左手を離す・ペンで蓋に印を付ける・ペンを手放す
発火系列 3	ペンを右手でさわる・ペンを右手で握る・スパナを左手で触る ペンで蓋に印を付ける・スパナから左手を離す・ペンを手放す
発火系列 4	ペンを右手でさわる・スパナを左手で触る・ペンを右手で握る ペンで蓋に印を付ける・スパナから左手を離す・ペンを手放す
発火系列 5	スパナを左手で触る・ペンを右手でさわる・ペンを右手で握る ペンで蓋に印を付ける・スパナから左手を離す・ペンを手放す

ることを可能にしているためである。つまり、表 B1 に示した発火系列は、「ペンで蓋に印を付ける作業」を目的にしているにも関わらず、目的の作業とは関係のない「ナット」や「スパナ」を操作しているために、作業工程に無駄が生じている。このような無駄な操作を排除するようにペトリネットの発火系列を求めることによって、検索の効率を上げることができるものと期待される。

B.2 発火系列の検索法

具体的には以下の手順で、発火系列の検索を行う。

手順 1 同一の入出力を持つ 2 つのトランジションを 1 つに融合する。

図 B1 に示すペトリネットを考える。このペトリネットのプレース p_1 に入っているトークンをプレース p_4 に遷移させるトランジションの発火系列を求めると $t_1 \cdot t_2 \cdot t_4$ と $t_1 \cdot t_3 \cdot t_4$ の 2 つの系列を求めることができる。しかし、自動実演を行う為には、どちらか一方の発火系列を求めるだけで十分である。そこで、2 つのトランジションを 1 つにまとめる。

手順 2 連続して発火させることになるトランジションの融合

「ペンを右手で握り、ナットを左手で握る作業」を考えた場合、自動実演を行う時は、必ず「ペンを右手で触る」、「ペンを右手で握る」、「ナットを左手で触る」、「ナットを左手で握る」といった順番で作業を行うことが普通であり、「ペンを右手で触る」、「ナットを左手で触る」、「ペンを右手で握る」、「ナットを左手で握る」といった順番で作業を行うことは、可能であるが、自動実演としては適していない。従ってこのようなトランジションの発火系列は必要が

ないため、検索を行う際に排除する。すなわち、他のトランジションの発火に影響を受けることのない、連続した作業のトランジションの組を1つに融合することで、検索すべきトランジションの総数を減らすことができる。すなわち、図 B.2 に示す2つのトランジションは、プレース p_3 と p_4 が、トランジション t_1 と t_2 以外に接続されていなければ、図 B3 に示すように、一つのトランジション $t_1 t_2$ に融合する。

手順3 前トランジションの融合

手順2と同様の考え方で、図 B4 に示すペトリネットは、図 B5 に示すようにトランジションの融合が可能である。すなわち、図 B4 において、プレース p で接続されたトランジション h およびトランジション集合 F が以下の条件を満たすとき、トランジション h と F のトランジション f の各組を融合して1つのトランジションにする。

1. $h \cdot = \{p\} \wedge W_0(p, h) = 0 \wedge W_0(h, p) = 1 \wedge W_1(p, h) = 0$
2. $\forall f \in F : W_0(p, f) = 1 \wedge W_0(f, p) = 0 \wedge W_1(p, f) = 0$
3. p は h, F 以外のトランジションと接続していない
4. $|M_0(p)| = 0$
5. $(\cdot h) \cdot = h$

ただし、 $|M_0(p)|$ はプレース p の初期トークンの数である。

手順4 互いの発火可能性に影響を与えないペトリネットの分割

図 B6 に示すようなペトリネットを考える。このペトリネットは、2つのサブネットが「手に何も持っていない」状態を表すプレース（以下 HandFree プレースと呼ぶ）で結合されている形になっている。この HandFree プレースには、発火系列の検索を行う際に初期配置として必ずトークンが存在するので、発火系列を検索する際に、お互いのサブネット間で、発火可能性に影響を与えることはない。従って、ペトリネットを2つのサブネットに分割し、それぞれに対して、可達木を作成することにより、発火系列検索の計算量を減らすことができる。

手順5 外部から発火可能性に影響を受けないサブネットの抽出とペトリネットの階層化

図 B7 のようなペトリネットを考える。このペトリネットの斜線で示された部分は、他の部分と、FreeHand プレース以外からは、発火可能性に影響を受けることはない。発火系列の検索の際に FreeHand プレースに必ずトークンが存在していることから、この部分のみに関する可達木を作成し、発火系列の検索を

行うことが可能である。発火系列の検索の結果を FreeHand プレースにトークンが存在するトークン配置で区切ることにより、発火系列をその変化させるプレースの数に応じて、以下の様に分類することができる。

系列 1 発火の前後で、トークン配置に変化がない

系列 2 発火の前後で、トークン配置を 1 つだけ変化させる

系列 3 発火の前後で、複数のトークン配置を変化させる

検索された発火系列の結果によっては、ペトリネットを階層化できる場合がある。例えば、上記 2 の発火系列を 1 つ持ち、しかもその発火系列によって変化するプレースのみが外部から参照されている場合（図 B. 7 のような場合）は、発火によってトークンが取り去られるプレースを入力プレースに、トークンが増やされるプレースを出力プレースにし、その発火系列が発火可能となるための条件を入出力プレースに配置することにより、発火系列 1 つに対して、トランジション 1 つを作成し、ペトリネットの階層化を行うことができる。このように、発火系列の結果によっては、ペトリネットを階層化することができるが、どのような発火系列を持つサブペトリネットが階層化可能であるのかについての詳細な研究は、今後の課題である。

手順 6 可達木を作成する

これまでの操作により単純化されたペトリネットの可達木を作成する。4. 3. 項で述べたモデル化法で作成したペトリネットは、トランジションの発火によってトークンの総数が変化しないため、作業の初期状態におけるトークンの総数が有限であることから、作成される可達木も有限である。

手順 7 発火系列の検索

現在のトークン配置と、目的のトークン配置が示されているとする。この目的のトークン配置に至るまでの発火系列の検索は、以下の手順で行う。

Step1 手順 4 で分割したペトリネットごとに発火させるべきトランジションの系列を可達木から求める。

Step2 手順 5 により階層化したペトリネットを展開する。

Step3 手順 2, 手順 3 で融合したトランジションを展開する。

Step4 手順 1 で得たトランジションは、その何れかを選択する。

以上のような手順により、目的のトークン配置に至るまでの発火系列を得ることができる。

B.3 ペトリネットの簡素化と発火系列の検索の例

本節では、B.2節で述べた方法で、実際にペトリネットの簡素化を行い、ペトリネットの発火系列の検索を行う。モデル化の対象としては図 B8 に示すような作業空間を考える。すなわち、作業空間にペン、箱、板、ボルト、ナットが存在する。訓練生は、ペンを用いて箱に印を付けることができる。また、ナットと板は、ボルトに填めることができるが、ナットがボルトに漬っている間は、板の詰め外しができないものとする。このような作業空間をペトリネットでモデル化すると、図 B9 のようになる。このペトリネットは、トランジションの総数が 30 個、プレースの総数が 33 個である。トークンの初期配置（作業開始の段階の各物体の状態）は、プレース $P1, P2, P3, P4, P9$ にそれぞれトークンが 1 つずつ存在することにする。

このようなペトリネットの発火系列を検索する場合、そのままの形で可達木を作成すると、非常に巨大になる。そこで、B2 節で述べた方法で、ペトリネットの簡素化を行う。

- 手順 1 によるトランジションの融合

今回のペトリネットでは、同一の入出力を持つトランジションは存在しないので、融合できるトランジションはない。

- 手順 2 によるトランジションの融合

図 B9 のペトリネットは、自動実演を行う時には、トランジション T_{10} と T_{11}, T_{11} と T_{12}, T_{18} と T_{10}, T_{19} と T_{10} を、必ず連続して発火させることになるので、これらを融合する。

- 手順 3 による前トランジションの融合

図 B9 のペトリネットでは、条件を満たすトランジションが存在しないので、融合できるトランジションはない。

- 手順 4 によるペトリネットの分割

図 B9 のペトリネットでは、トランジション T_1 から T_{12} までと、トランジション T_{13} から T_{10} を分割することができる。

ここまでの作業により、図 B9 のペトリネットは、図 B10 と図 B11 に示すペトリネットに分割されることになる。

- 手順 5 によるペトリネットの階層化

図 B10 のペトリネットを見ると、トランジション T_1 から T_{10}, T_{11} までのサブペトリネットは、外部のネット（トランジション T_{13} から T_{12}, T_{12} まで）から発火可能性の影響を受けないことが分かる。そこでトランジション T_1 から

T10T11までのネットの可達木を作成し、発火系列の検索を行うと、B.で述べた系列2の発火系列が2つ検索されることがわかる。すなわちプレース P2にあるトークンをプレース P11に遷移させる発火系列と、プレース P11にあるトークンをプレース P2に遷移させる発火系列が検索される。このとき、トランジション T1からトランジション T10 T11までのサブペトリネットは、外部のネットから、プレース P11のみが参照されており、手順5で述べたことから、このサブネットは図 B. 1に示す2つのトランジションに階層化することができ、図 B. 1のペトリネットは、図 B. 1に示すペトリネットに階層化できたことになる。

- 手順6による可達木の作成

これまでの作業により、ペトリネットは2つに分割され、1段階の階層化が行われた。ここでは、図 B. 1と図 B. 1の可達木を作成することになる。

- 手順7による発火系列の検索

例えば、作業開始時から、プレース P1, P11, P22, P2にトークンが存在する状態に変化する発火系列を求めると、T13-T14-T15-T17-T1の発火系列が可達木から求めることができる。ここで、T34は階層化されたペトリネットであり、このトランジションと同様の働きをするトランジションの発火系列はT1-T3-T5-T7である。従って、これらの結果から、求めるべき発火系列はT13-T14-T15-T17-T19-T1-T3であることが分かる。

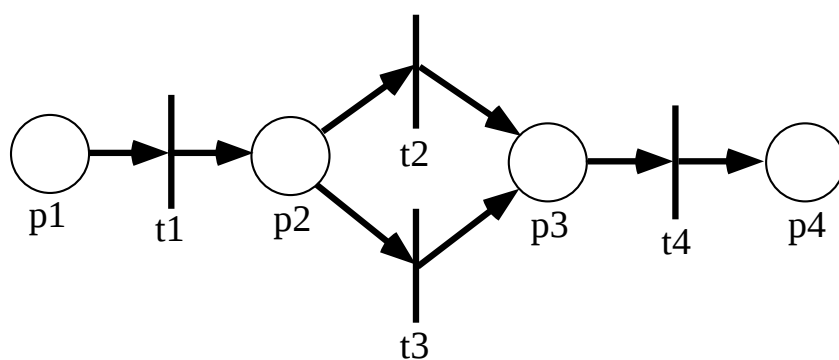


図 B.1: 冗長なトランジション

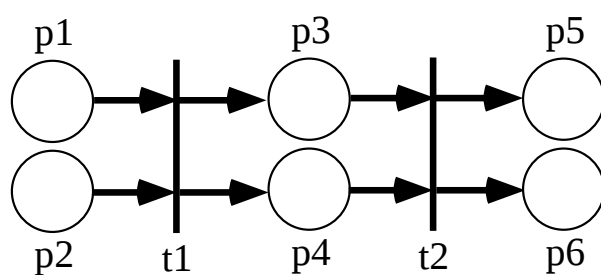


図 B. 2:融合前のトランジション

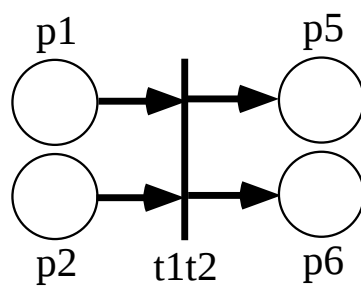


図 B. 3:融合後のトランジション

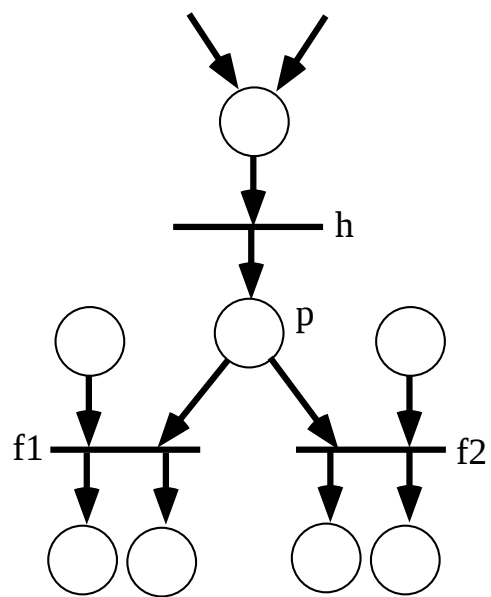


図 B.4: 融合前のトランジション

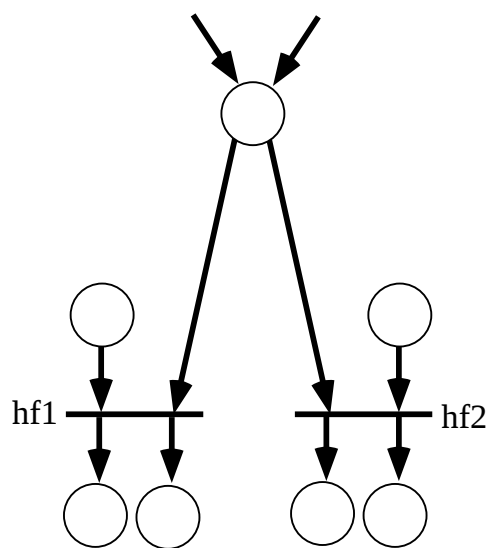


図 B. 5:融合後のトランジション

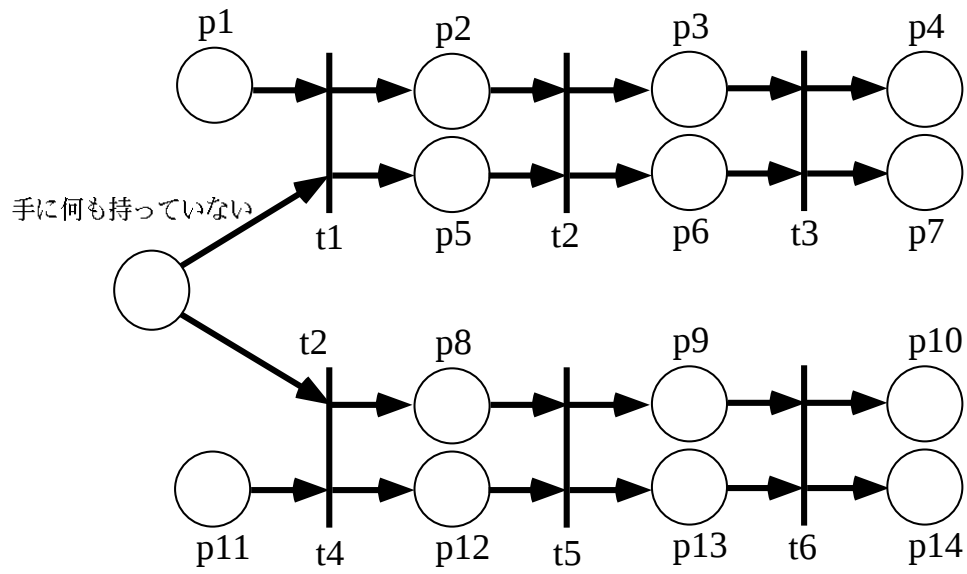


図 B.6: 分割可能なペトリネット

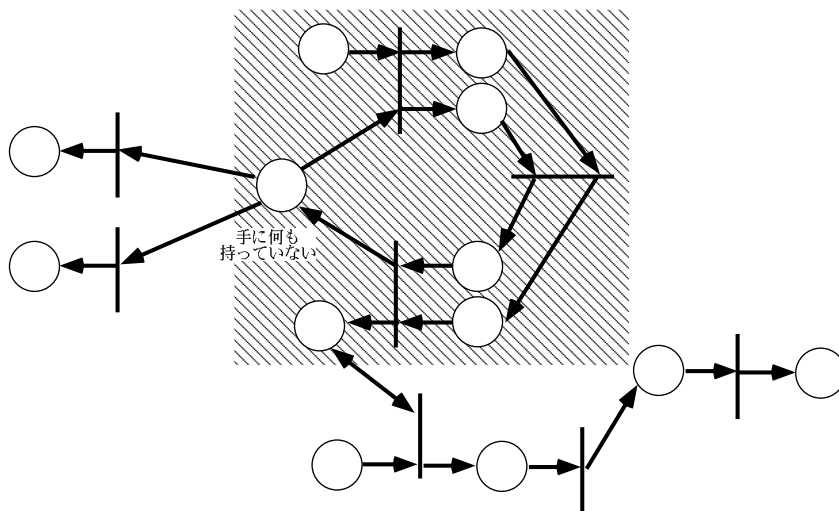


図 B. 7: 階層化が可能なペトリネット

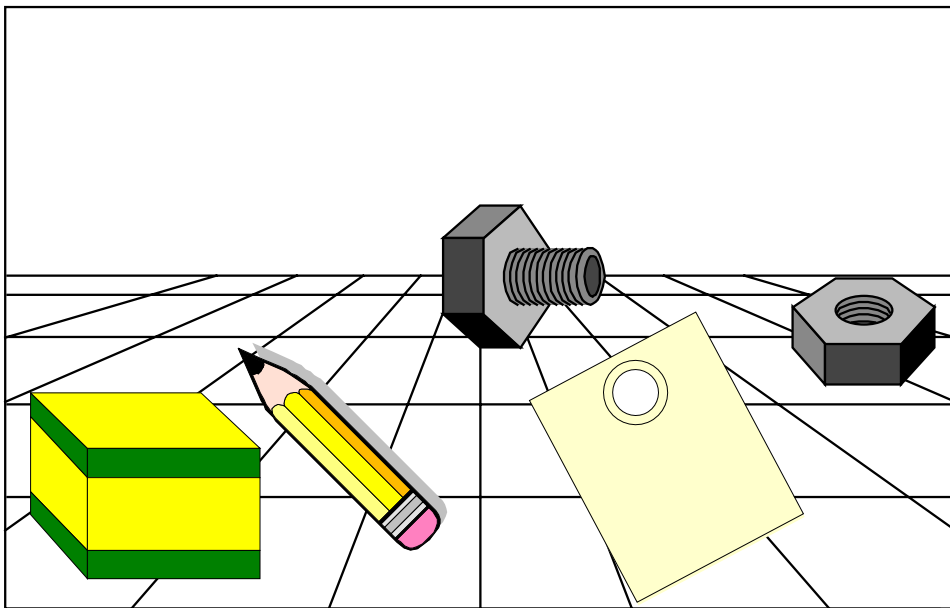


図 B.8: モデル化の対象とする作業空間

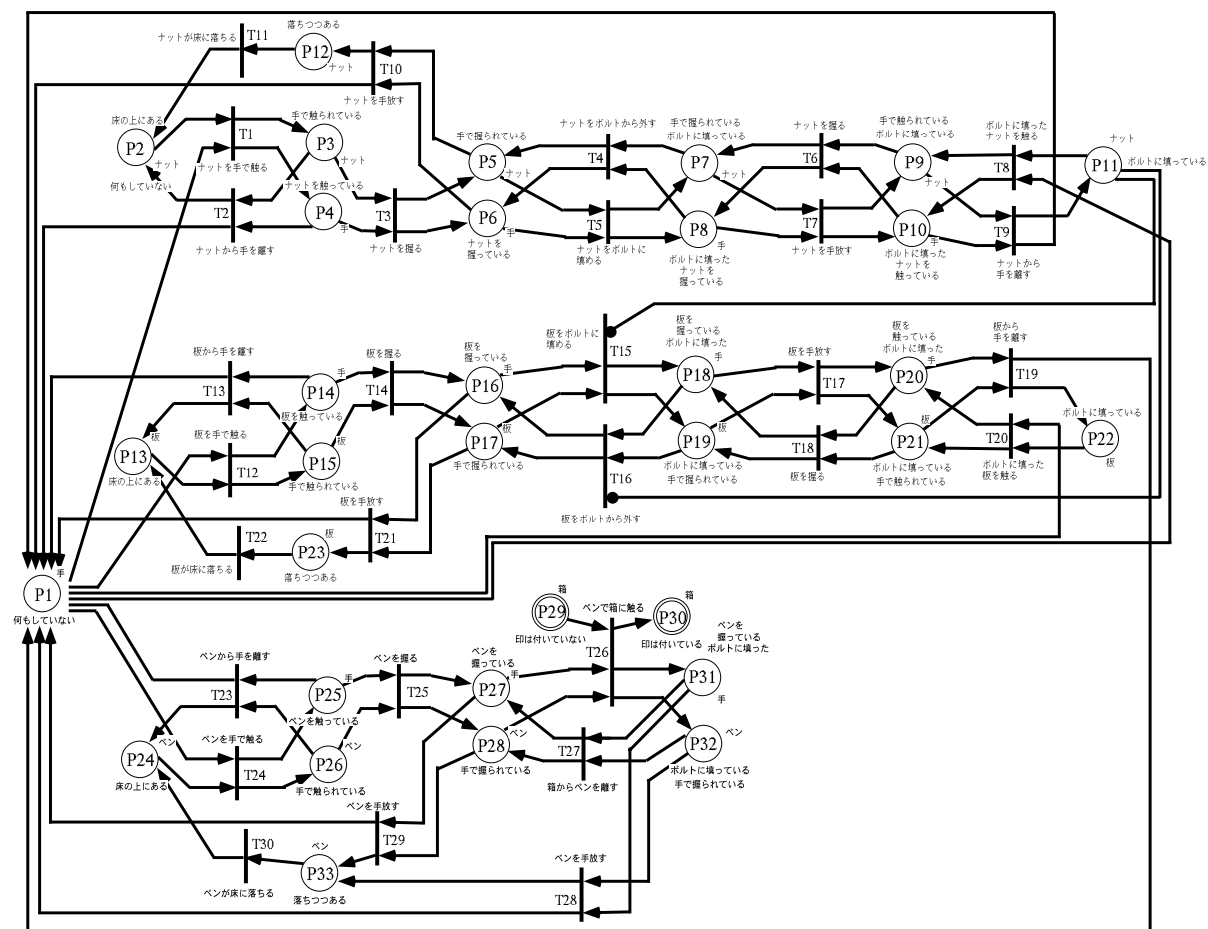


図 B.9: モデル化したペトリネット

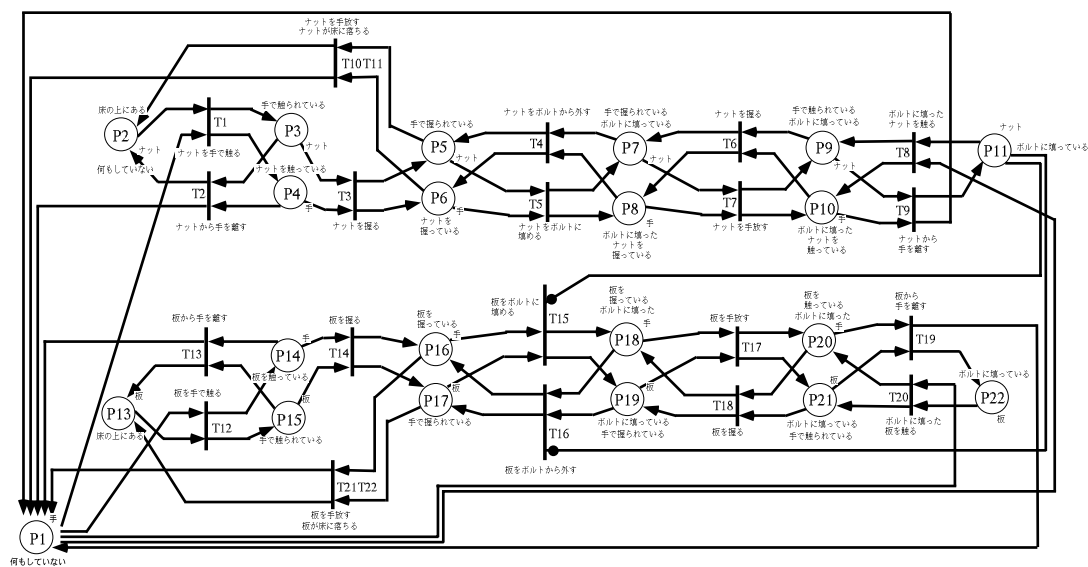


図 B. 10:分割したペトリネット (その1)

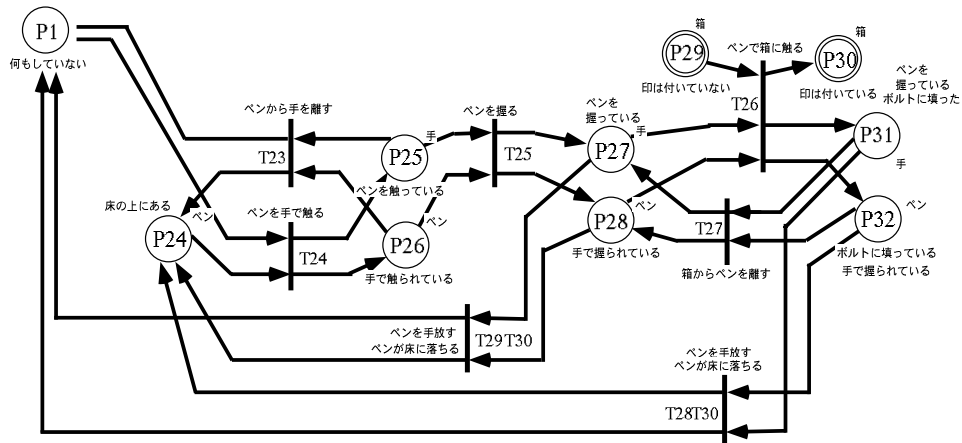


図 B.11: 分割したペトリネット (その2)

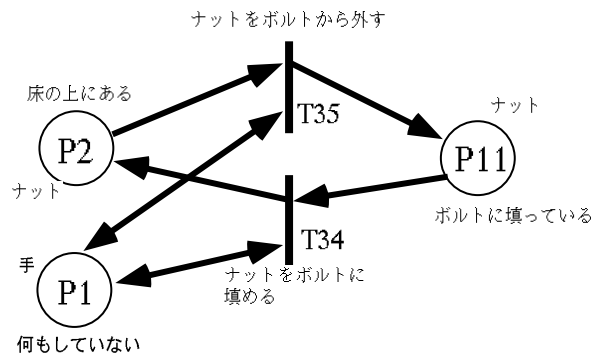


図 B.12: 階層化した後の2つのトランジション

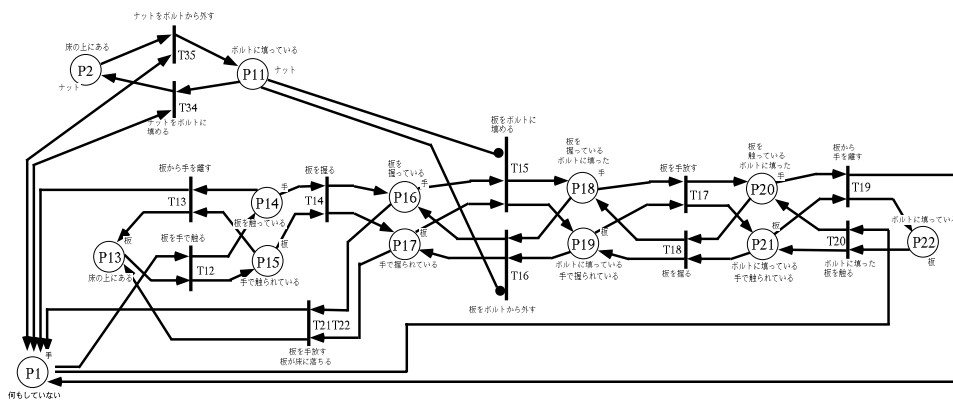


図 B.13: 階層化した後のペトリネット

付録 C 支援システムにおける 3 次元座標の記述法

支援システムにおいては、仮想空間内の物体の場所を指定するために、基準座標系と、位置 (X,Y,Z) 、姿勢 (A,B,C) の 7 自由度の情報を指定する必要がある。このうち、基準座標系は、絶対座標系(床を基準にした座標系)かオブジェクト座標系(他の物体を基準にした座標系)を指定する。また、位置のパラメータである、 XYZ はそれぞれ、基準座標系の原点からの、 X 軸方向の平行移動距離、 Y 軸方向の平行移動距離、 Z 軸方向の平行移動距離を表す。また、姿勢のパラメータである、 ABC は基準座標系の X 軸、 Y 軸、 Z 軸の順番に、それぞれの軸を中心に回転させる角度をラジアンで表す。各パラメータの関係を図 C1 に示す。

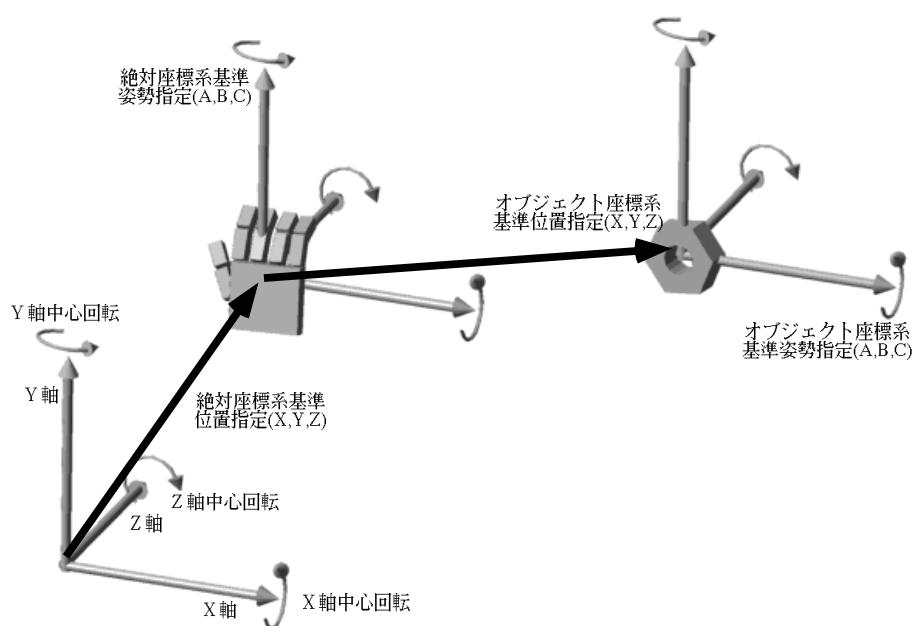


図 C1: 絶対座標系とオブジェクト座標系