

人工現実感技術を用いた 機器保修の訓練環境構築支援システムの開発

石井 裕剛^{*1} 手塚 哲央^{*1} 吉川 榮和^{*1}

Development of a Design Support System for Constructing Machine-Maintenance
Training System based on Virtual Reality Technology

Hirotake Ishii^{*1}, Tetsuo Tezuka^{*1} and Hidekazu Yoshikawa^{*1}

Abstract – A design support system has been developed for constructing VR-based training environment for machine maintenance work without any expertise and programming effort on VR. Using the developed system, users can easily construct various training environments under GUI environment. It was verified through some experiments that the developed system could reduce the working hours remarkably and that novice users who had no prior knowledge on the system could construct a training environment successfully after a few hours of tutorial on the system. In this paper, the basic concept of the system, the way to model the state transition of objects by Petri net and experimental results are described.

Keywords : Design Support System, Training System, Virtual Reality, Petri Net

1. はじめに

近年の人工現実感技術の発展により、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 [1]。人工現実感技術を用いた訓練環境としては、宇宙空間でハッブル宇宙望遠鏡の修理作業の仮想体験ができる環境 [2]、変電所保守員向け体感型シミュレータ [3] 等が構築されている。

筆者らは既に、仮想空間内に原子力プラントの構成機器の 1 つであるスイング式逆止弁の分解作業の訓練環境を構築した [4]。その際、「仮想空間内の物体の状態遷移の仕方をペトリネットを用いてモデル化する手法」を開発し、ペトリネットのモデル化能力の高さを利用することにより、従来の訓練環境 [2] [3] に比べ、作業の自由度が高い訓練環境を効率的に構築できた。

しかし、これらの先行研究ではシステムの殆どがプログラミングにより構築されており、新たに別の対象についての訓練環境を構築するには、プログラムを修正もしくは新規に作り直す必要がある。そのため、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には、膨大な労力が必要となり、多くの時間と人件費が必要となる。また、訓練環境の構築作業は、計算機に精通した専門のプログラマーが行う必要があり、現場のインストラクタが直接訓練環境を構築することができないため、実際のニーズを反映した訓練環境を構築

することが難しい。

そこで本研究では、GUI(Graphical User Interface)を介して必要な情報を入力することにより、プログラミングを全く行わずに、複雑な機器に対する訓練環境を容易に構築することができる構築支援システムの開発を目指すこととした。

GUI を介して仮想空間を構築できるシステムとしては、「RealMaster」や「dVISE」等が開発されている [5]。「RealMaster」は株式会社ソリッドレイ研究所が開発したシステムで、プログラミングを行わなくても物体の操作ができる仮想空間の構築を可能にしている [6]。また、「dVISE」は英国 Division 社が開発したシステムで、メニューによる対話入力で、プログラミングなしに仮想空間の構築、操作を可能にしている [7]。これらのシステムを用いることにより、コンピュータに関する高度な知識のない人でも「ドアのノブを触るとドアが開く」「レバーを操作するとクレーンが動く」といった、単純な仮想空間は容易に構築できる。しかし「RealMaster」は、インタラクティブな仮想空間を構築するために、「イベント」と「アクション」の関係を記述し各物体の運動の仕方等の細かい設定を行う「シーンファイル」と呼ばれるテキストを作成する必要がある。また「dVISE」は、「イベント」と「アクション」の関係を全て GUI を介して設定できる機能を備えているが、設定すべき「イベント」と「アクション」の関係が複雑になった場合に、それらを効率的に扱うための機能は備えていない。そのた

^{*1}: 京都大学大学院エネルギー科学研究科

^{*1}: Graduate School of Energy Science, Kyoto University

め、作業手順が非常に複雑なモータやポンプ等の機器を分解・組立できるような仮想空間を構築する場合、複雑な構築作業を行う必要があり、不可能ではないが多くの時間が必要である。

そこで本研究では、筆者らの先行研究[4]で提案した「仮想空間内の物体の状態遷移の仕方をペトリネットを用いてモデル化する手法」を、ペトリネットに関して初心者であるユーザにも理解し易いように定義し直し、構築支援システムに応用した。ペトリネットは、その概念が非常に単純であり、また、グラフを用いることにより視覚的に記述できることから、モデル化したペトリネットを視覚的に入力するインタフェースを構築することにより、「イベント」と「アクション」の関係を効率的に設定できるものと期待される。

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[8][9]、仮想空間の構築を支援するシステムにペトリネットを応用した例は見あたらない。

本論文では、まず始めに構築支援システムにおける基本概念について述べる。続いてペトリネットによる物体の状態遷移のモデル化の方法について述べた後、構築支援システムを用いた訓練環境の構築例と、実験により構築支援システムの評価を行った結果について述べる。

2. 構築支援システムにおける基本概念

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のを可能にするには、各物体相互間の干渉や物理法則等をシミュレーションする必要がある。例えば、手でペンを握っていれば、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。このようなシミュレーションを行うための1つの方法として、物体の3次元形状データをもと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方法も考えられるが、リアルタイムで処理を行うためには非常に高性能な計算機が必要である。また、電気機器等を操作する訓練環境を構築する場合、「ボタンを押すとモータが回り出す」といった仮想空間を構築するためには、仮想空間内の3次元モデルにイベントを発生させるための仕掛けを設ける必要がある。

そこで本構築支援システムでは、物体に「状態」の概念を導入し、各物体の取りうる状態それぞれに対して、物体の運動の仕方を予め定めておき、さらに物体の状態遷移の仕方とその状態遷移を起こさせるイベントとの関係を記述することによって、物体同士の干渉や、物理法則を考慮に入れた仮想空間を構築するものとした。

例えば「ナットが机の上に置いてあり、手でつかんだり放したりする環境」を構築する場合、ナットの取りうる状態と、各状態における運動の仕方、状態遷移の仕方

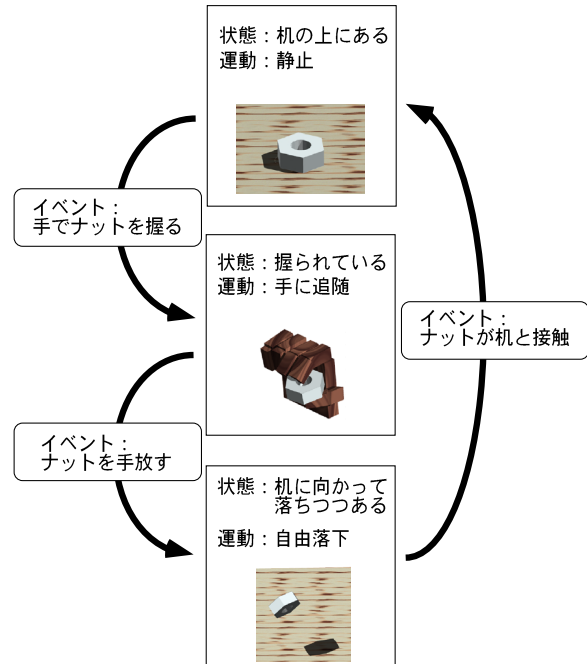


図1 ナットの状態遷移と各状態における運動の仕方

Fig.1 State transition of a nut and its movement at each state

と状態遷移を起こさせるイベントの関係は図1のようになる。

具体的には、訓練環境を構築する際にはユーザは以下の2つの情報を作成する。

- 運動データベース

物体が取りうる状態とその状態における物体の運動の仕方を規定したデータベース

- 状態遷移データベース

イベントの発生に応じた各物体の状態遷移の仕方を規定したデータベース

特に「運動データベース」に関しては、各種の情報の設定を項目選択と数値入力に分類して構成することとした。この時、複雑な訓練環境を柔軟に構築できるためには、数多くのパラメータを設定可能にする必要がある。そこで、本研究では、実際の機器保守の訓練作業の内容をもとに、設定可能とすべきパラメータを抽出して、物体の外見に関するデータ、物体の動きに関するデータ等、約60のパラメータを設定できるようにした。具体的なパラメータの内容に関しては3節において述べる。

また、「状態遷移データベース」の構成に関しては、物体の状態遷移の仕方をペトリネットを用いてモデル化する方法に従って、支援システム上でペトリネットを視覚的に構築することにより、物体の状態遷移の仕方を登録できるようにした。具体的には、支援システム上で、遷移前の状態と遷移後の状態、そして状態遷移を起こさせる原因となるイベントを項目選択で入力することによ

り、ペトリネットを作成するものとした。具体的な内容に関しては4節において述べる。

3. 運動データベース

3.1 物体の運動と物体間の拘束関係

本研究では、複雑な訓練環境を柔軟に構築できるようにするために、実際の機器保修の訓練作業の内容をもとに、構築支援システムで最低限設定できる必要があるパラメータを抽出した。

例えば、「訓練生が手に物体を握っている状態」を表現するためには、握られている物体の運動を「手の動きに追従する」ように設定できる必要がある。また、「スパナでボルトにはまったナットを緩める作業」を構築可能にするためには、「同時に複数の物体が、ある1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。さらに「握っている物体を放すと、その物体が自由落下する」仮想空間を構築するためには、物体の運動を、訓練生のジェスチャに関係なく等加速度運動するように設定できる必要がある。

以上のような検討をもとに、本構築支援システムで入力可能とすべきパラメータについて抽出した。パラメータは物体の外見に関する情報と、物体の運動の仕方に関する情報に大別することができる。以下ではそのそれぞれについて述べる。

3.2 物体の外見に関する情報

物体の外見に関しては以下の3種類のパラメータを設定可能とした。

- 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「手を握っている状態から手を開く」といった、物体の形状に変化が伴う動作を、データを差し替えることにより表現できる。

- 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。

- 描画方法

実際の描画の方法として、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）、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。

3.3 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。以下ではそのそれぞれについて述べる。

- 時間の推移に応じて変化する場合

この場合は、物体の位置と姿勢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。ユーザ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、時刻 t における仮想空間内のある物体の位置を (X_t, Y_t, Z_t) 、姿勢を (A_t, B_t, C_t) とすると、ボルトにはまり、手に握られているナット等を表現する場合は、式(1)のようになる。

$$\begin{cases} X_{t+1} = X_t & A_{t+1} = A_t + a \\ Y_{t+1} = Y_t + ha & B_{t+1} = B_t \\ Z_{t+1} = Z_t & C_{t+1} = C_t \end{cases} \quad (1)$$

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。

上記のパラメータの他に、本構築支援システムでは訓練中に提示する効果音も選択できるようにした。これにより、物体同士が衝突した時に衝突音を提示することも可能である。

3.4 状態テンプレート

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、仮想空間内の多数の物体の状態には、よく似た物理法則を適用すべき場合が多数存在することに注目し、よ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。

4. 状態遷移データベース

4.1 ペトリネットの定義

本構築支援システムで用いるペトリネットは、筆者らの先行研究[4]において提案したペトリネットを基本と

しているが、初心者でも容易に支援システムを利用できるように、一部定義を変更した。以下ではその定義の変更部分のみを述べる。

図2にグラフでの表記法の一覧を示す。以下では、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、訓練シミュレーションの際に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を必要としないことを意味する。

- 制御プレースの廃止

先行研究では、ペトリネットの冗長性を避けるため、手の状態を制御プレースを用いて表していた。そのため「オブジェクト1つがトークン1つに対応する」という規則が手のオブジェクトに関してのみ守られていなかったが、初めてペトリネットを構築する際は、全てのオブジェクトに関してこの規則が守られている方が分かり易いため、制御プレースを廃止し、支援システムでは手の状態もシングルプレースを用いて表すこととした。

- 抑止アークの導入

1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、ペトリネットにプレースのゼロテストを行う能力が無ければ、モデル化することができない[10]。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、新たに抑止アークを導入することとした。グラフではプレースからトランジションへ、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を引くことにより表現する。このアークを用いることにより、モデル化できる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。

- 参照プレースと参照アーク

機器必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、「ダイヤルを回す」等、電気機器を操作しても、電源の状態は常にONのままであり、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、1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ぶ。

用語	グラフ表現	用語	グラフ表現
シングルプレース	○	プールプレース	○
ノーマルトランジション		自動発火トランジション	
ノーマルアーク	→	抑止アーク	→●
参照アーク	↔	トークン	●

図2 ペトリネットのグラフ表記法
Fig.2 Graphical notation of Petri net

また、グラフでは記述を容易にするために、プレースとトランジションの間を双方向の矢印で結ぶことにより表現する。この双方向の矢印を参照アークと呼ぶ。

- カラートークンの導入

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、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1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だけを「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ようにする必要がある。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同一のオブジェクトが複数存在する場合に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。

- トランジションの入出力アークの総数

先行研究では制御プレースを用いていたため、トランジションの入出力ノーマルアークの総数は必ずしも同じではなかった。本支援システムではトランジションの入力ノーマルアークの総数と出力ノーマルアークの総数は必ず等しいものとなる。

4.2 ペトリネットによるモデル化

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プレースに、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。そして、物体がある状態にあることを、その状態に対応するプレース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。具体的には以下の手順でモデル化を行う。

- (1) 訓練環境内に配置する物体を抽出する

ここで、抽出する物体には床や両手も含まれる。例えば、「机の上にあるペンを右手で握って床の上に落

表 1 構築支援システムで選択可能なイベント

Table 1 Events which can be selected in the support system

イベント	例
物体 1 と物体 2 が接触する	スパナと床が接触する
物体 1 と物体 2 が分離する	右手をペンから離す
手を開く	右手を開く
手を握る	左手を握る
条件が成立する	ペンを握っている

とす」作業の場合、抽出すべき物体は、「机」「ペン」「右手」「床」となる。

(2) 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。

(3) 各物体が状態遷移する原因となるイベントを抽出する

本構築支援システムでは表 1 に示す 5 種類の項目を状態遷移を起こさせるイベントとして選択できるようにした。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、「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である。ここで、「右手でペンを触る」イベントは構築支援システム上ではイベントの項目列の中から「物体 1 と物体 2 が接触する」を選択し、「物体 1」に「右手」を、「物体 2」に「ペン」を対応させることにより設定する。

(4) ペトリネットを構築する

(2) で抽出した状態をブレースに対応させ、(3) 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ブレースに、後の状態を出力ブレースに対応させることによって、ペトリネットを構築する。

(5) 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、訓練環境の初期状態を設定する。このとき、個々のトークンに対して、仮想空間内の 3 次元位置を設定することにより、訓練開始時の各物体の初期位置を設定する。

4.3 ペトリネットによるモデル化の例

図 3 に「床の上に 2 本のペンがあり、右手でつかんだり放したりできる環境」をモデル化した例を示す。このモデル化では、ペンが 2 本あるため、ブレース「ペン：床の上にある、何もされていない」にトークンが 2 つ配置してある。このブレースはプールブレースとしてモデル化しているので、床の上には複数のペンを置くことができるが、ブレース「ペン：右手に握られている」はシングルブレースとしてモデル化しているため、訓練生は一度に 1 本のペンしかつかむことができない。

5. 支援システムの概要

5.1 構築支援システムの構成

本研究で開発した構築支援システムは、訓練環境を構築するモード（構築モード）と訓練を実施するモード（訓練モード）を持つ。訓練環境を構築する際には必要に応じてこれら 2 つのモードを切り替えることにより、仮想空間の状態を確認しながら構築作業を行う。構築支援システムの開発は一人のプログラマーが行い、約 6 カ月の開発期間を要した。また、プログラムコードの総数は約 18,000 行となった。構築モードで使用するハードウェア、ソフトウェアと、訓練モードで使用するハードウェア、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。

5.1.1 構築モード

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 4 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、情報提示部はディスプレイで構成される。情報処理部は SGI 社のワークステーション（Indigo2, Maximum IMPACT）で構成され、ソフトウェアとしては、OpenGL ライブラリと Motif ライブラリを使用した。ユーザは 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、必要な情報を入力する。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る。

5.1.2 訓練モード

訓練を実行する場合のシステム構成を図 5 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。計測部は 3 次元マウスで構成される。訓練生は、3 次元マウスのバックとボタンを操作することによって、手の把持・開放・回転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 SGI 社のワークステーション、ソフトウェア

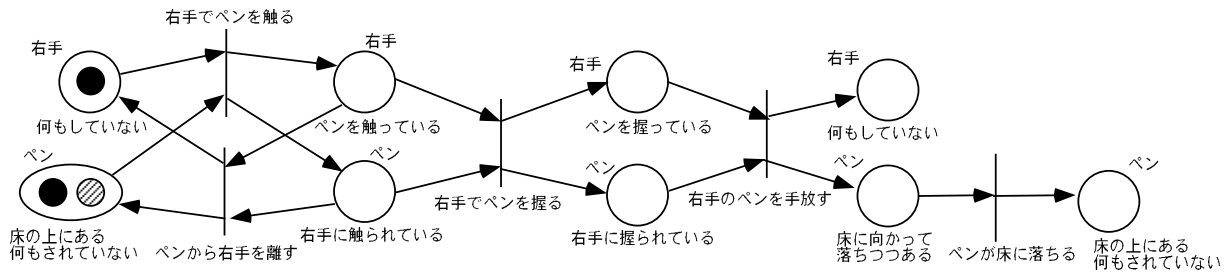


図3 ペトリネットによるモデル化の例
Fig. 3 An example of modeling with Petri net

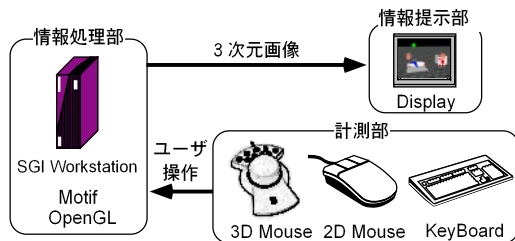


図4 システム構成 (構築モード)
Fig. 4 System configuration (construction mode)

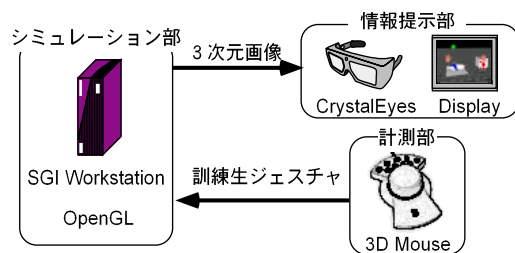


図5 システム構成 (訓練モード)
Fig. 5 System configuration (training mode)

として OpenGL ライブラリを用いた。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 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ことができる。なお、今回開発した構築支援システムでは、訓練生への力覚フィードバックの提示は考慮していない。

5.2 支援システムのインターフェース

訓練環境を構築するために必要な情報の入力、は、トグルボタンによる項目の選択や、スライダーによる数値の設定により行う。支援システムの操作ボタンには図 6 に示すようにアイコンが張り付けてあり、支援システムを利用する場合に、各ボタンの機能が直観的に理解できるように工夫した。具体的には以下の手順で必要な情報を入力する。

1. 訓練に必要な物体名の登録

例えば「右手を用いてペンで蓋に印を付ける」訓練（以後「印を付ける訓練」と呼ぶ）を構築する場合、



図6 アイコンを張り付けたボタンの例
Fig. 6 An example of buttons with icons

「右手」「ペン」「蓋」「印」を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行う状態の登録を行う。例えば、「印を付ける訓練」を構築する場合、「ペン」の状態に関して、「床の上にある」「右手に触られている」「右手に握られている」「自由落下している」を登録する。これらはペトリネットにおいて、プレースの名称となる。そして、図 7 に示すようなインターフェースを介して、3 節で述べた運動データベースを入力する。

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

図 8 に示すようなインターフェースを介し、ペトリネットを構築する。特にトランジションの入出力プレースに関しては、「イベントの発生により状態遷移する物体の名称」「状態遷移前の状態」「状態遷移後の状態」「状態遷移を抑止する物体の状態」等を入力することにより、ペトリネットが構築される。このように「イベントの発生に応じて、どの物体が、どのように状態変化するか」といった直観的な情報入力でペトリネットを構築できることが支援システムの特徴となる。

4. 物体の初期状態の登録

訓練を実際に行うためには、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。

5.3 訓練を実施する際の情報処理

本支援システムにおける訓練実行時の情報の流れを図 9 に示す。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実行し、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

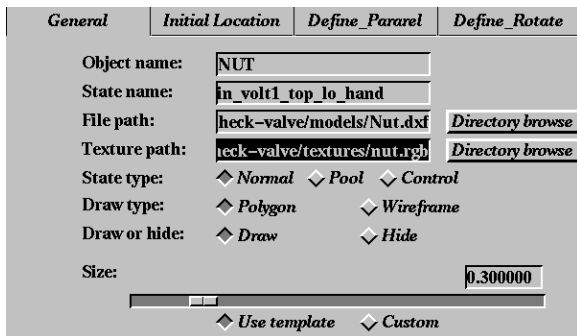


図7 支援システムのインターフェースの画面例

Fig. 7 An example of GUI for setting parameters

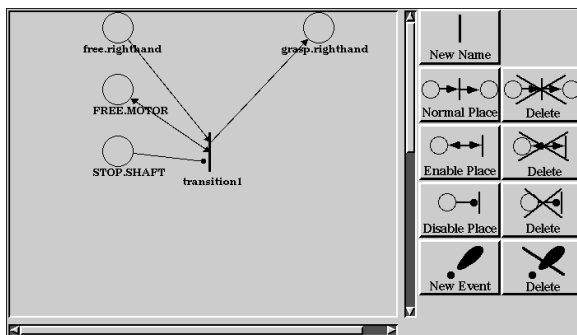


図8 ペトリネット構築の際の画面例

Fig. 8 An example of GUI for constructing Petri net

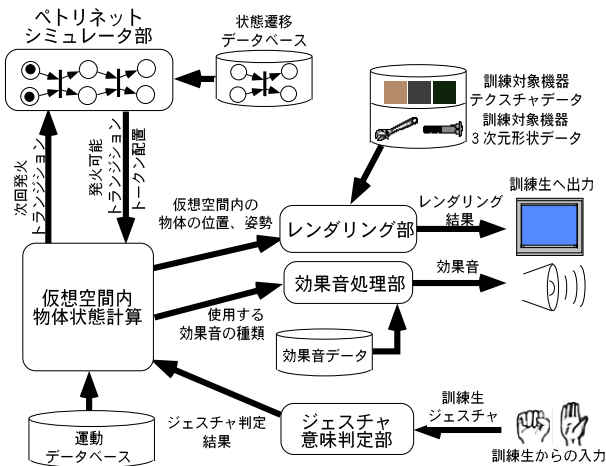


図9 訓練実行時のソフトウェア構成ブロック図

Fig. 9 Block diagram for training

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イベント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

訓練生からの3次元マウスの入力を調べる。訓練生からの入力は、ボタンの状態(押されているかどうか)

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。これらのデータをもとにジェスチャ判定部でどのようなジェスチャ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イベントと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運動データベースをもとに、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る。その後、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運動データベース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。

このような内部処理を行って訓練を実施する場合、ペトリネットをシミュレーションするための処理が新たに必要となり、計算機への負荷が増えるという問題がある。しかし、負荷が増えるのはイベントが生じた時だけであり、また、6.1項で述べる逆止弁の保守作業を行う訓練を実施する場合では、ペトリネットの処理時間は全体の処理時間の約0.9%に過ぎず、さらに、ペトリネットシミュレータを訓練システムとは別のプロセスで処理する事も可能であるため、大きな問題とはならないと思われる。

5.4 ペトリネットを利用することの利点

仮想空間内の物体の状態遷移のモデル化にペトリネットを利用することによる主な利点として、以下の3つを挙げることができる。

- グラフにより視覚的に表現できる

ペトリネットを用いることにより、物体の状態遷移の仕方を視覚的に表現することができる。これにより、

実行可能な一連の作業を直観的に知ることができる。例えば、図3を見ると、「右手でペンを触る」「右手でペンを握る」「右手のペンを手放す」という名前のトランジションが直列につながれていることから、これらの動作を連続して行うことができることがわかる。また、「右手でペンを触る」トランジションと「ペンから右手を放す」トランジションが1つのループを形成していることから、これらの作業は繰り返し実行することができることが分かる。

- モデル化能力が高い

ペトリネットは並列に生起する事象を扱うことができるため、「右手でスパナを緩める作業をしながら左手でペンをういて印を付ける」といった2つの動作を同時に実行できる仮想空間も容易にモデル化することができる。また、任意に対象を選択させる訓練環境が容易に構築可能であり、訓練生に与えることが可能な選択枝の数には制限はない。実際に6.1項で述べる訓練環境では、訓練開始時に、「ペンを触る」「スパナを触る」「蓋を触る」「ナットを触る」の4つの動作の何れからでも作業を始めることができる。

- 構築したペトリネットを用いて新たな機能を付加することができる

ペトリネットの発火系列を検索することにより、例えば、「機器を目的の状態にするまでの最短の作業系列」を求めることができる[11]。この結果を利用することで、訓練環境に自動実演機能を付加することができる。

6. 支援システムの使用例と評価

6.1 スイング式逆止弁の分解組立作業

筆者らの先行研究[4]において、プログラミングにより構築した「スイング式逆止弁」の分解・組立作業を行うことができる訓練環境を、本研究で開発した構築支援システムを用いて新たに構築し、両方法での構築の作業効率の比較を行った。

6.1.1 構築する訓練対象の機器

スイング式逆止弁の構造を図10に示す。今回構築した訓練環境は、簡単化の為に操作の対象を弁の蓋の部分に限定し、また蓋を固定するナットとボルトの総数も実際の機器に比べ16個から4個に減らしている。しかし、蓋を固定するナットは、訓練開始時には固く締め付けられているため、最初にスパナで緩めなければ、手で回転させることはできない等、できる限り実世界と同様の訓練環境を提供するようにした。また、一度外したナットや蓋は、任意の時点で再び填める作業を行うことができる等、出来る限り自由な手順で作業を行うことができる環境を構築した。このような訓練環境は、ペン、スパナ、ナット等、8種類、計14個のオブジェクトで構成され、

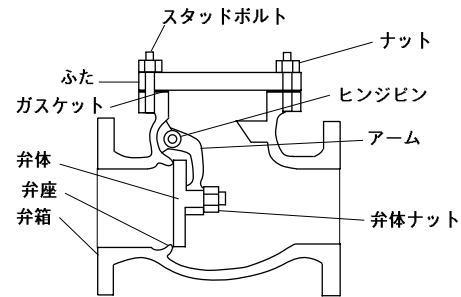


図10 逆止弁の構造

Fig.10 Structure of check valve

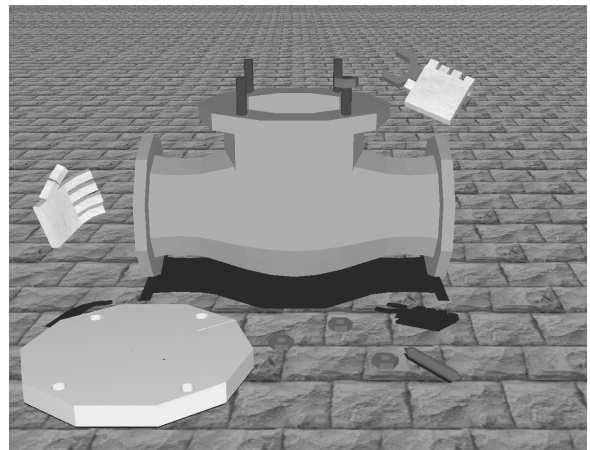


図11 逆止弁の保守訓練環境の画面例

Fig.11 An example scene of maintenance training of check valve

作業開始から、約100回トランジションを発火させる（すなわち約100回イベントを生起させる）ことにより作業が終了する。

6.1.2 構築結果

構築した訓練環境の画面例を図11に示す。構築した訓練環境のデータをファイルに保存した場合のデータ量は、3次元の形状データが394kbyte、テクスチャデータが211kbyte、運動データベースが429kbyte、状態遷移データベースが47kbyteとなった。今回の構築作業では、構築支援システムの操作に習熟したユーザが1人で作業を行い、作業開始から作業終了までに約12時間が必要であった。この作業時間には訓練対象機器の3次元形状データを構築する時間は含まれていないが、先行研究においては、同様の訓練環境をプログラミングにより構築したので、2人のプログラマにより、約4ヶ月の作業時間を要したと比べると、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作業効率が飛躍的に改善されたと言える（表2）。

6.1.3 考察

このように飛躍的に作業効率が改善された理由は次の様に考えることができる。プログラミングにより訓練環境を構築する場合には、仮想空間内で訓練作業を可能に

表 2 作業時間の比較

Table 2 Comparison of time for constructing training environment

	先行研究	支援システムを使用
構築作業人数	2人 (プログラマ)	1人 (支援システムユーザ)
構築作業時間	約4カ月	約12時間

するための各種アルゴリズムの考察や、プログラムのバグ取り等の作業が必要であり、これらの作業はプログラマが熟練していなければかなりの時間を要する。しかし、構築支援システムを用いて作業を行う場合には、各種アルゴリズムを細部まで考察する必要がなく、また、作業手順が簡略化されたため、間違いを起こしにくくなった。

6.2 支援システムの操作性の実験による評価

構築支援システムを初めて利用する初心者にも容易に訓練環境が構築できることを確認するために評価実験を行った。評価は初心者に簡単な訓練環境を構築させ、その構築に必要とした時間と、作業終了後のアンケート調査により行った。

6.2.1 被験者と実験方法

実験の被験者は支援システムを初めて使用する20歳代と40歳代の男性2人とした。両者ともプログラミングにより訓練環境を構築した経験は無いが、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することができる。訓練に利用する機器の3次元形状データとテクスチャファイル、提示する効果音は、予め適当なアプリケーションを用いて作成されており、被験者には各ファイルの格納場所とその内容について予め説明した。また、構築作業を開始する直前に、構築支援システムの操作に十分習熟した者が、構築支援システムの各機能の説明と操作方法等を約30分間説明し、最後に構築作業の例として、「手でペンを握ったり放したりできる訓練環境」の構築作業を見学させた。実験では、作業開始から作業終了までに要した時間を計測し、また、作成した訓練環境データの内容も記録した。アンケートは、構築作業が終了した直後に行った。

6.2.2 構築の対象とする訓練環境

被験者が構築する訓練環境は、「手でスイッチを操作することにより、モータのシャフトの回転を制御する訓練環境」とした。仮想空間に配置する物体は、モータ、シャフト（ファン付）、スイッチ、スイッチの台の計4つとなり、そのそれぞれについて、3次元形状データとテクスチャデータを予め準備した。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3と図12に示す。仮想空間内の物体の状態は、シャフトが「回転中」と「静止」の2つの状態をとり、他の物体は状態が変化しない。従って、この構築作業では、ペトリ

表 3 回転中のシャフトの状態を表す情報

Table 3 State information of rotating shaft

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	shaft.rgb	状態タイプ	normal
描き方	polygon	サイズ	1.0
初期位置	無し	平行移動	不可
回転移動	可能(時間)	回転可能軸	ローカル Z 軸
マウス移動	不可		

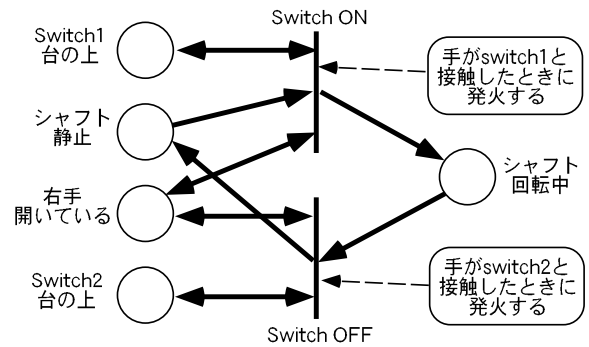


図 12 シャフトの状態遷移のモデル化例

Fig. 12 An example of Petri net (state transition of a shaft)

ネットの作成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、と予想した。

6.2.3 実験結果

最終的には2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。構築した訓練環境の画面例を図13に示す。訓練環境構築に必要な作業時間は、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少し戸惑ったが、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、両者から次のような感想が得られた。

- 支援システムを利用してみて、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、使っているうちに分かるようになった。
- 3次元マウスが使いにくく、オブジェクトの位置合わせが難しかった。
- 使っている内に、もっと凝った仮想空間を作ってみ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。

6.2.4 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

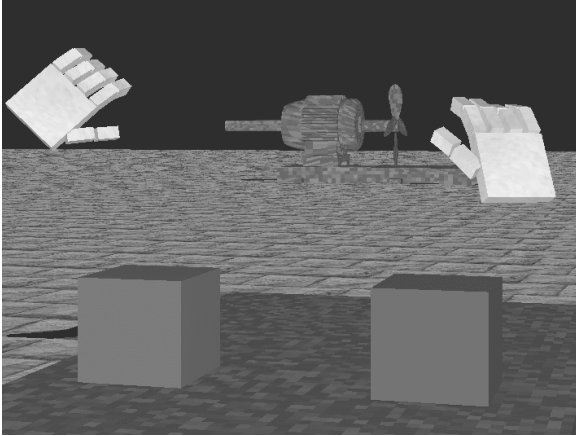


図 13 モータ操作の訓練環境の画面例
Fig.13 An example scene of training motor operation

最終的には正しく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、プログラミングにより仮想空間を構築させた場合、プログラミングの基礎から学ぶ必要があり、非常に時間が必要となると思われるが、本システムを用いることにより、せいぜい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ことが判明した。また、3 次元マウスを用いて、正確にオブジェクトの位置を設定することは、3 次元マウスの操作に慣れていない初心者には困難なことも判明した。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についても検討すべきと思われる。

今回、構築の対象となった訓練環境は単純であったため、両被験者が構築した仮想空間やペトリネットは、物体の配置位置等を除けば、熟練者が構築するものと全く同じであった。しかし、訓練環境が複雑になると、幾通りの仮想空間の記述が可能になるため、構築する人によって構築方法に違いが生じ、作業効率にも違いが生じると予想される。これは、仮想空間構築のテクニックを利用できるかどうかに影響される問題であり、こういったテクニックをユーザが効率的に学ぶことができれば、訓練環境を構築する際の作業効率をさらに改善できると考えられる。

7. ま と め

本研究では、人工現実感技術を用いた訓練環境の構築作業の効率化を図るため、訓練環境の構築支援システムの開発を行った。その際、ペトリネットを用いて仮想空間内の物体の状態遷移の仕方をモデル化する手法を考案し、支援システムのインタフェースに応用した。そして、開発したシステムの評価実験を行い、プログラミングに

より訓練環境を構築する場合に比べて、作業効率が飛躍的に改善されることを確認した。また、支援システムを初めて利用する場合でも、数時間の練習を行うだけで、目的の訓練環境を構築できることを確認した。

本研究で開発した構築支援システムは、インタフェースの善し悪しにより、作業効率がかかなり違ってくることが考えられる。例えば、操作メニューの階層構造は、作業効率に大きく関わってくると予想される。また、アイデア次第では、支援システムに様々な機能を加えることにより、さらに使いやすいシステムに改良できると期待される。今後は、システムの更なるインタフェースの改良と、システムへの自動実演機能の付加、複数人で作業を行う訓練環境を構築できるシステムへの拡張を行っていきたい。

参 考 文 献

- [1] Seidel, R.J. and Chatellier, P.R. : "Virtual Reality, Training's Future?", NATO Defence Research Group, PLENUM PRESS(1997)
- [2] Loftin, R.B. and Kenney, P.J. : "Training the Hubble Space Telescope Flight Team", IEEE, Computer Graphics and Applications, Vol.15, No.5, pp.21-37(1995)
- [3] 新井浩一、阿倍慶子、上地登 : "VR 技術を用いた変電所保守員向け集合教育用体感型シミュレータの開発", 日本バーチャルリアリティ学会論文誌, Vol.2, No.4, pp.7-16(1997)
- [4] 吉川榮和、手塚哲央、柏健一郎、石井裕剛 : "仮想空間における機器保守訓練シミュレーション", 日本原子力学会誌, Vol.39, No.12, pp.1078-1089(1997)
- [5] 野村淳二、澤田一哉編著 : "バーチャルリアリティ", 朝倉書店, pp.22-47(1997)
- [6] 神部勝之 : "Real Master", 第 5 回産業用バーチャルリアリティ展・セミナー要録 (IVR'97), pp.87-93(1997)
- [7] 村上竜一 : "dVISE", 第 5 回産業用バーチャルリアリティ展・セミナー要録 (IVR'97), pp.69-80(1997)
- [8] 中島豊四郎、伊藤宏隆、南里孝行 : "ペトリネットによる伸線プラントシステムの解析", 電気学会論文誌, Vol.114-C, No.9, pp.881-887(1994)
- [9] 玉置久、棕本進、西川喜清、荒木光彦 : "一般化ペトリネット・モデルを用いたシミュレーション・ベースド・スケジューリング", 電気学会論文誌, Vol.114-C, No.9, pp.888-897(1994)
- [10] J. L. Peterson : "ペトリネット入門", 共立出版, pp.190-212(1984)
- [11] 包原考英、稲葉昭夫、川瀬俊夫、鈴木達也、大熊繁 : "ペトリネットを用いた組立作業計画", 計測自動制御学会論文誌, Vol.29, No.4, pp.461-469(1992)

(1998 年 5 月 15 日受付)