

〔非公開〕

TR-C-0064

ネットワーク仕様  
記述処理システムについて

全 明 康

Akiyasu Zen

1 9 9 1 . 3 . 2 5

A T R 通信システム研究所

1	まえがき	1
2	システムの概要	2
3	取り扱い説明	3
3.1	動作環境	3
3.2	起動	3
3.3	ウィンドウ構成	4
3.4	仮想ネットワークの構築と表示	7
3.4.1	Command Syntax	7
3.4.2	初期化の流れ	9
3.4.3	状態遷移表	10
3.4.4	タスク管理	11
3.4.5	Configuration File Format	13
3.5	動作状況表示	15
3.6	システムへの入力	18
3.7	動作環境設定コマンド	19
3.8	デバッグモードにおける処理系の実行状態制御用コマンド	24
3.9	その他のコマンド	27
4	システムの構成	30
4.1	処理系サブシステム	30
4.1.1	処理系の動作環境設定	30
4.1.2	ステップ処理モードにおける外部とのインタラクション	33
4.1.2.1	制約評価部の移行に際しての処理の制御	33
4.1.2.2	発火すべき制約式の発火処理の制御	34
4.1.3	処理系の動作環境／実行状況問い合わせ	36
4.1.4	処理系の実行状況操作	39
4.2	表示系サブシステム	43
4.2.1	仮想ネットワーク管理	43
4.2.2	画面管理	43
4.2.2.1	仮想ネットワークの構成を表示するための図の作成管理	44
4.2.2.2	仮想ネットワークの動作状況を 表示するための図要素の作成管理	44
4.2.2.3	仮想ネットワークの動作状況を 表示するための図要素の運用管理	44
4.2.3	状態遷移管理	46
5	あとがき	48
	参考文献	48
	付録	49

近年、通信サービスへ寄せられる多種多様なより一層高度化・複雑化した要求に答えるために、既存サービスの再利用が重要な課題となっている。

本研究所では、通信サービスをネットワーク上に分散配置した機能群（プロセス群）の協調処理として捉え（図1参照）、制約指向型の言語により個々のプロセスの動作仕様をプロセス固有の機能を記述する機能部と他のプロセスとの協調処理を記述する相互作用部に分けて記述することで既存サービスの再利用、複合サービスの設計支援が可能になると考え、動作仕様記述用の言語CSLの制定を行った。

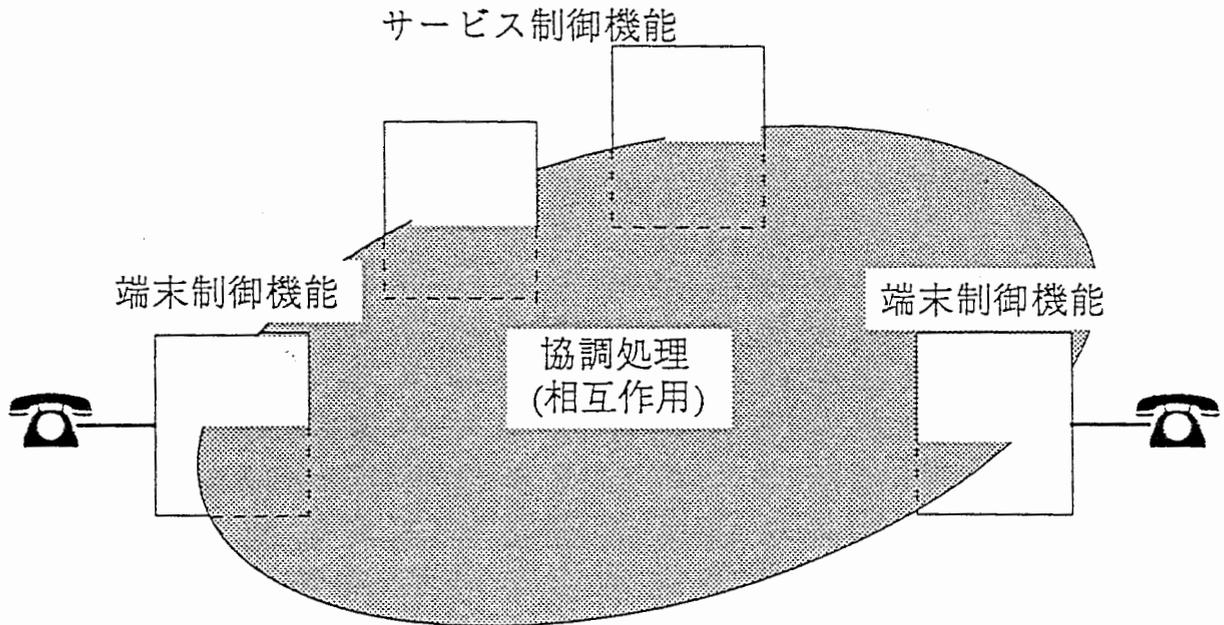


図1 プロセス群の協調処理として捉えた通信サービス

本文は、上記の言語CSLにより各プロセスの動作仕様を記述したものを分散的に配置して仮想ネットワークを構築し、ネットワークとしての動作状況をシミュレートすることを目的としたネットワーク仕様処理システムについてその取り扱い、システムの構成および各サブシステムの内容を記述したものである。

## § 2 システムの概要

ネットワーク化様処理システムは、CSLにより記述されたプロセスの動作仕様を解釈実行する処理系サブシステム（以下処理系と略す）と、ユーザとのI/Oインタフェースや仮想ネットワークの構成等を司る表示系サブシステム（以下表示系と略す）からなる。

処理系は仮想ネットワークを構成するプロセス毎に起動され、表示系をコントローラとしたスター型接続で接続されている。表示系はユーザが画面に図的に表示した仮想ネットワークを操作することによって与えた入力を処理系に配送し、処理系間でやり取りされる情報の転送をすると共にその情報を用いてシステムの動作状況を図的に表示する。

システムはユーザの設定に応じた仮想ネットワークを作成し、ユーザの端末機操作によりイベントを発生させてネットワークの動作状況を表示するものである。システムの構成イメージを図2に示す。

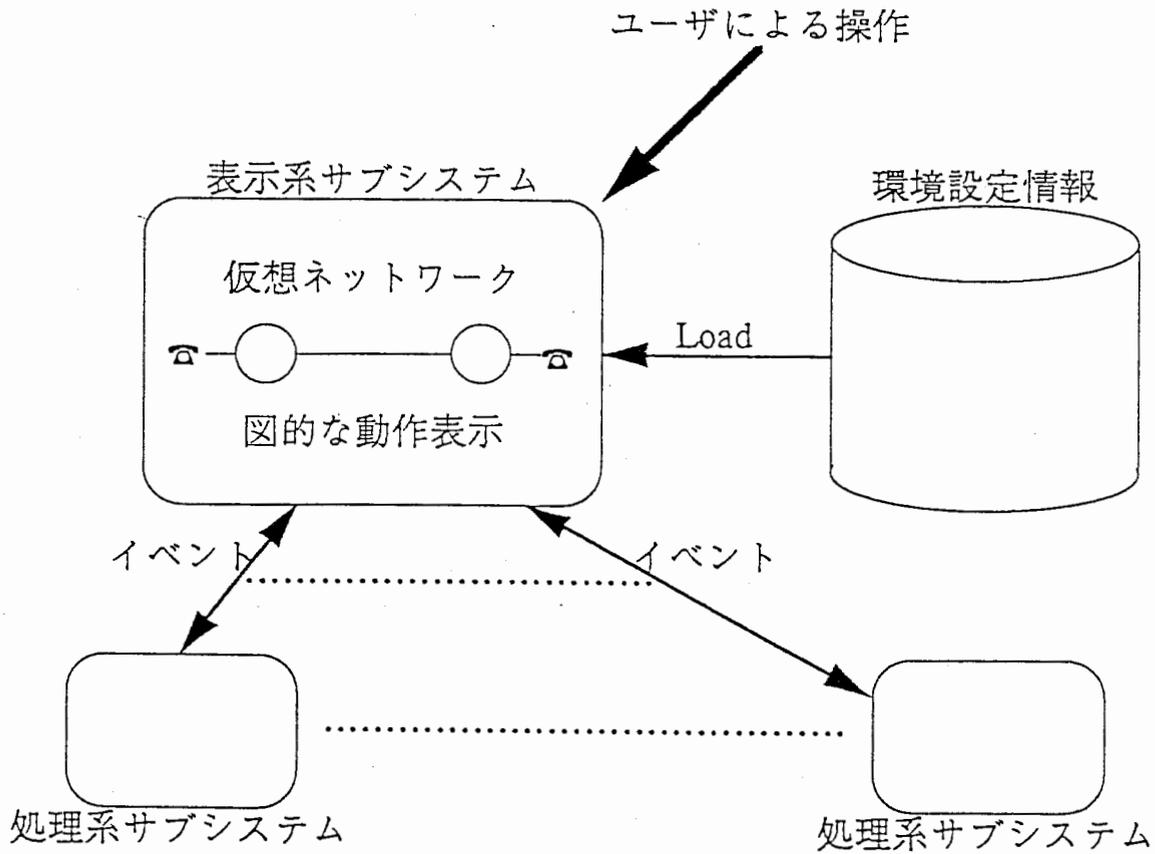


図2 システムの構成イメージ

## § 3 取 扱 い、説 明

### 3.1 動作環境

ネットワーク仕様処理システムはSymbolics Genera 7.2を対象として作成されており、処理系と表示系は同一のホストで稼働できないという制約がある。従って、ネットワーク仕様処理システムを稼働させるためには最低で二台のホストがSymbolics Genera 7.2で稼働している必要がある。

処理系として稼働させるためにはCSLというSystemを、表示系として稼働させるためにはDCSLというSystemをロードする必要がある。以下ロードの方法を示す。

Load System CSL (処理系の場合)

Load System DCSL (表示系の場合)

また、ネットワーク仕様処理システムを稼働させるホストにはCSL-PROTOCOLというプロトコルでCSLというサービスが提供されていなければならない。

### 3.2 起動

ネットワーク仕様処理システムの起動は、表示系をロードしたホストで表示系のウィンドウを開くことによって行う。また、処理系は仮想ネットワークを構築する際に表示系からの指示により自動的に起動される。表示系のウィンドウは次のようにして開く。

<Select> + <Network>

これは<Select>キーに引続き<Network>キーをタイプインすることを示す。

### 3.3 ウィンドウ構成

表示系のウィンドウはネットワーク仕様の動作状況を表示するためのノーマルモードのものと、あるプロセスに着目し、その着目されたプロセス内部の動作状況を含めた詳細な情報を表示するためのデバッグモードのもの二種類がある。ネットワーク仕様処理システムが初期化された直後、ウィンドウはノーマルモードのものに設定されている（Default がノーマルモード）。以下ノーマルモードのウィンドウを図3に、デバッグモードのウィンドウを図4に示しその構成について規定する。

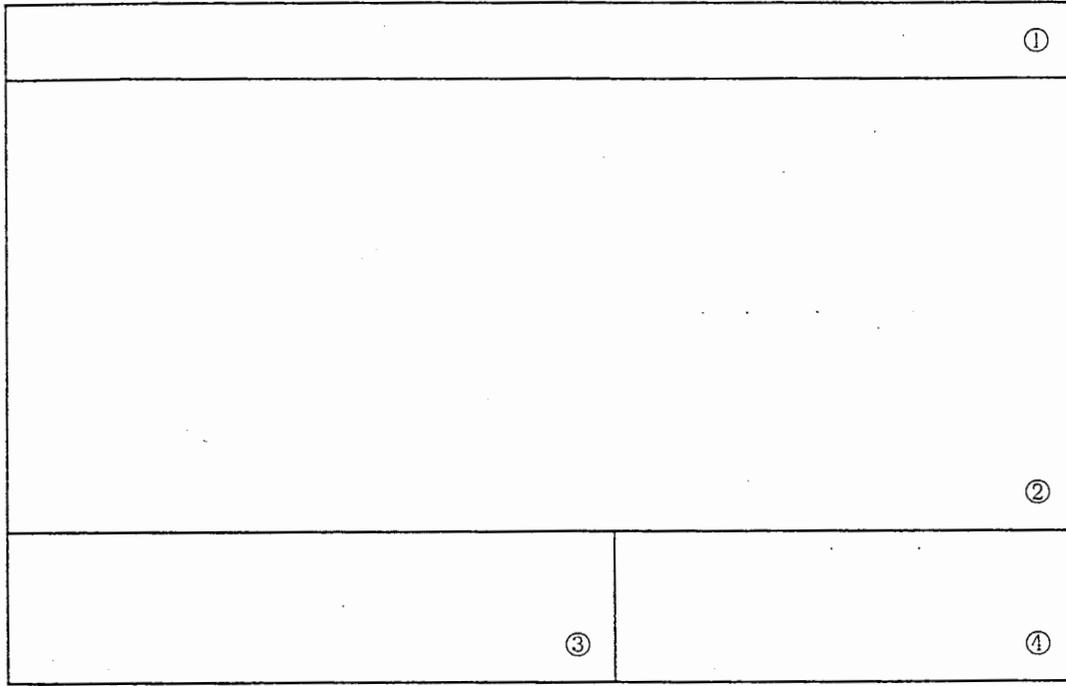


図3 ノーマルモードのウィンドウ構成

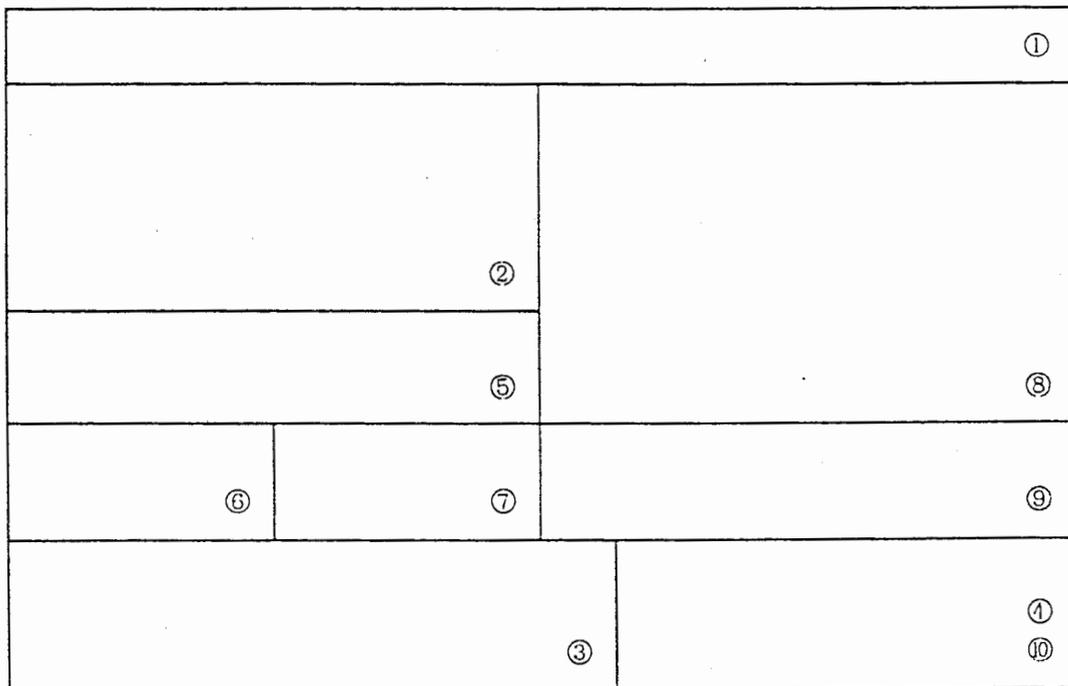


図4 デバッグモードのウィンドウ構成

ウィンドウは一つ以上のペインと呼ばれる表示領域で構成されるものであり、表示系のウィンドウはノーマルモードが①から④の4つのペイン、デバッグモードが①から⑩の10個ペインで構成されている。  
以下、個々のペインの役割について規定する。

① タイトル表示ペイン

システムのタイトルを表示するためのペインで、ノーマルモードの場合は単純にタイトルが、デバッグモードの場合は着目しているプロセスに関する情報が表示される。

② システム動作表示ペイン

仮想ネットワークの構成を表示し、図示した端末機を操作させることによってシステムへの入力手段を提供、システムの動作状況を図的に表示するためのペインで、デバッグモードではその表示に用いる図形がノーマルモードでのその縮尺1/2で表示される。

③ イベント表示ペイン

仮想ネットワークを構成する各プロセスが処理対象とする仕様記述にしたがってプロセスの内部およびプロセス間でやり取りされるイベントを表示するためのペインである。

④ コマンド入力ペイン

表示系のトップレベルのコマンドインタプリタが入出力に用いるペインである。

⑤ プロセス内部動作表示ペイン

デバッグモードにおいて、着目するプロセスの内部動作を図的に表示するためのペインである。

⑥ 状態変数表示ペイン

デバッグモードにおいて、着目するプロセスが持つ状態変数およびアドレス変数の値を表示するためのペインである。該当するプロセスが管理下に置く全ての状態変数およびアドレス変数について表示する。

⑦ バウンドイベント表示ペイン

デバッグモードにおいて、着目するプロセスが持つイベント変数の値を表示するためのペインである。ただし、値を持たないイベント変数についてはその表示を行わないものとする。

⑧ 制約表示ペイン

デバッグモードにおいて、着目するプロセスが処理の対象とする仕様記述の内容を表示するためのペインである。

⑨ 発火制約表示ペイン

デバッグモードにおいて、着目するプロセスが処理の対象とする仕様記述の内発火が予定される制約式を表示し、ユーザにその制約を発火させるかどうかを問い合わせた結果を併せて表示するためのペインである。

⑩ 問い合わせメッセージ表示ペイン

デバッグモードにおいて、着目するプロセスが処理の対象とする仕様記述の内発火が予定される制約式が見つかったときに、その制約式を発火させるかどうかをユーザに問い合わせる（選択操作を促す）メッセージを表示するためのペインである。

システムはユーザからの次の図5に示すようなコマンド操作により上記の二つのモードを切り替え、該当するモードでの表示を行う。

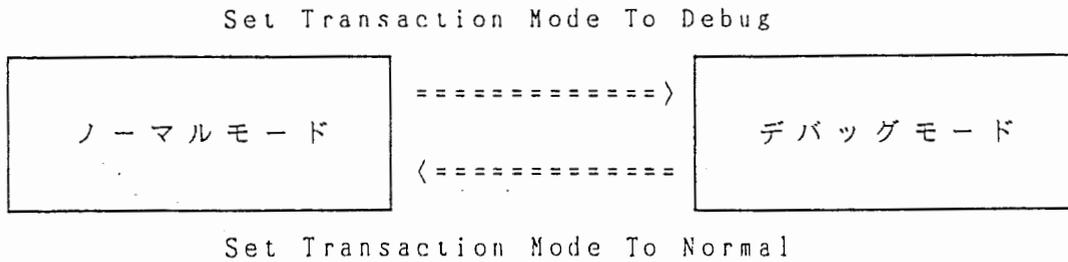


図5 コマンド操作によるモードの切り替え

### 3.4 仮想ネットワークの構築と表示

仮想ネットワークの構築は表示系において Init System Environment command を実行することによって行われる。 Init System Environment command は各種 configuration file の内容にしたがって仮想ネットワークおよび仮想ネットワークを表示するのに用いる図形、画面の作成を行いネットワーク仕様処理システムの初期化を行うものである。以下その command syntax とシステムの初期化の流れと各種関連テーブルの内容について記述する。

#### 3.4.1 Command Syntax

```
Init System Environment :Csl Configuration File file-name  
                        :Chart Layout File file-name  
                        :Chart Element File file-name  
                        :Event Conversion File file-name  
                        :State Transition File file-name  
                        :Redraw Device Icon ( Yes | No )
```

ただし、

- :Csl Configuration File は仮想ネットワークの構成に関する記述がなされた Configuration File を指定するためのキーワードである。
- :Chart Layout File は仮想ネットワークを表示するために用いる画面レイアウトに関する記述がなされた Configuration File を指定するためのキーワードである。
- :Chart Element File はネットワーク仕様処理システムが表示に用いる各種図形に関する記述がなされた Configuration File を指定するためのキーワードである。
- :Event Conversion File は表示系が解釈し画面の表示を変更させるためのタスクと CSL による仕様記述の際に Device から / Device へのイベントとして記述したものを変換するための変換テーブルを指定するためのキーワードである。
- :State Transition File は表示系が画面の表示を変更させるために用いるタスクによって遷移する各種端末機や回線の状態の遷移系列について記述された状態遷移表を指定するためのキーワードである。
- :Redraw Device Icon は表示に用いる図形に関する記述を元に表示系が内部操作で用いる各種インスタンスの作成やアイコンのビットマップの規定等の実施を指定するためのキーワードである。 Default は Yes。

また各キーワードで指定される File の Default を次のように規定する。

```
:Csl Configuration File Dcsl:Configuration;Csl-System  
:Chart Layout File      Dcsl:Configuration;Chart-Layout  
:Chart Element File     Dcsl:Configuration;Chart-Element  
:Event Conversion File  Dcsl:Configuration;Event-Conversion  
:State Transition File  Dcsl:Configuration;State-Transition
```

全て Default の File を用いたとき、コマンド実行後のウィンドウは次の図 6 のようになる。

Network Specification Evaluator Display

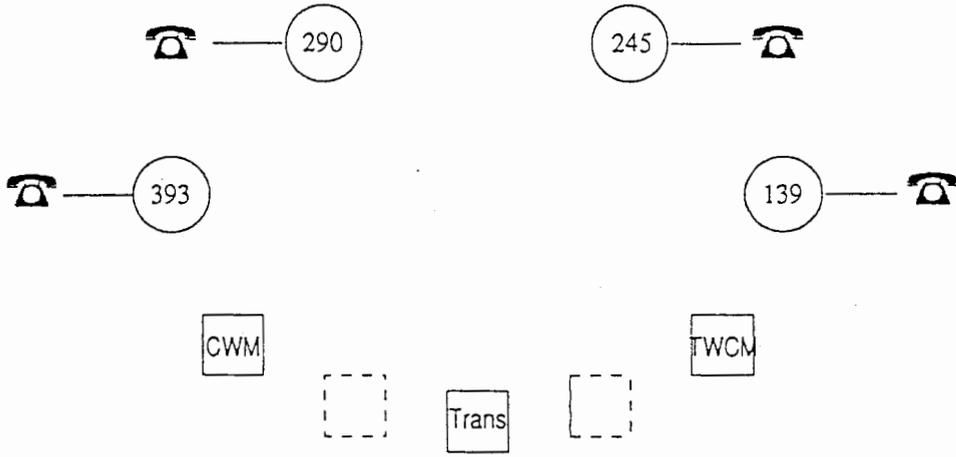


図6 Default Fileによる環境設定後のウィンドウ

### 3.4.2 初期化の流れ

ネットワーク仕様処理システムの初期化の流れを図7に示す。

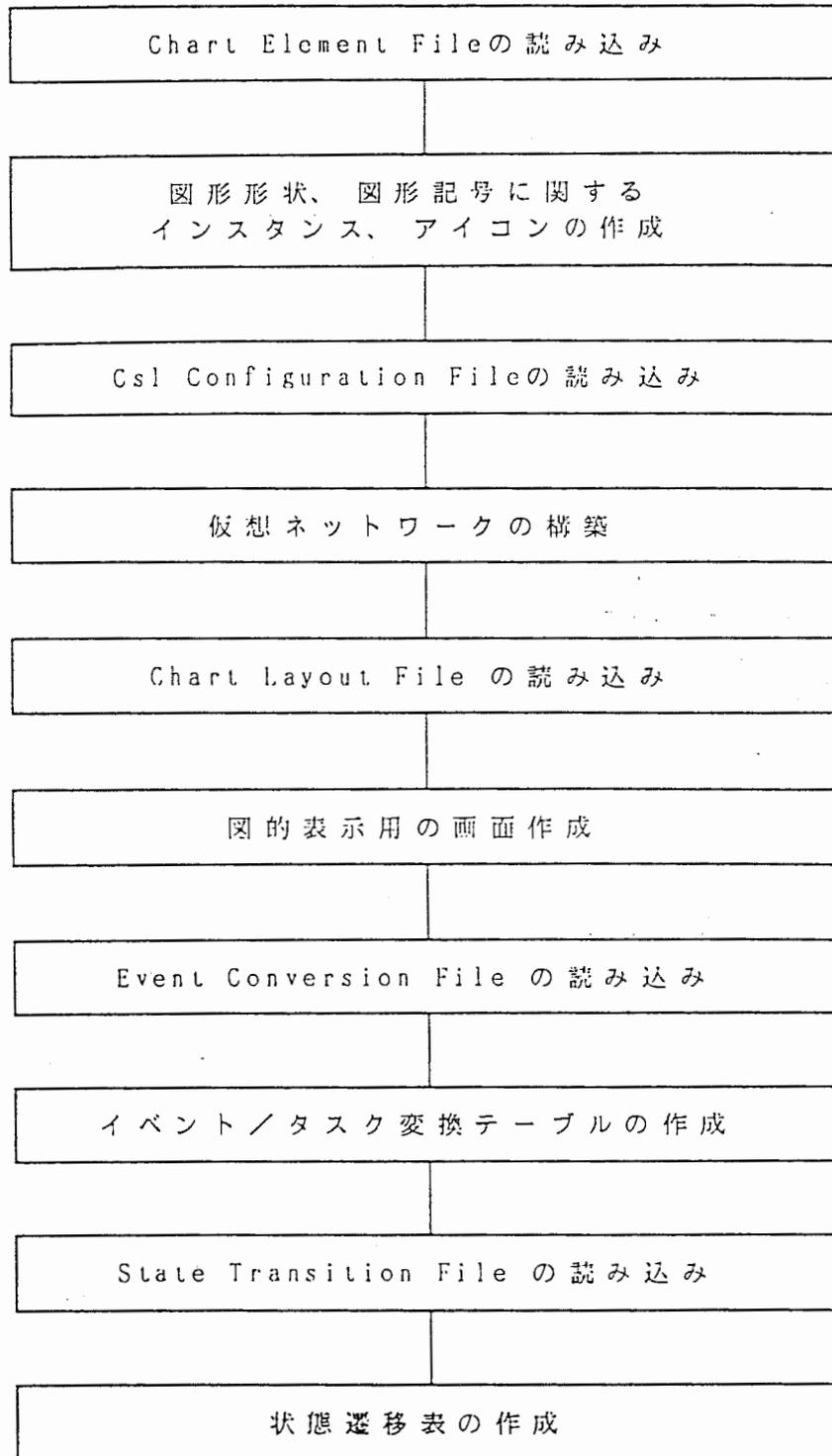


図7 ネットワーク仕様処理システム初期化の流れ

### 3.4.3 状態遷移管理

表示系は端末機の状態と回線の状態を管理し、タスクと呼ばれる状態遷移を促す信号を受けることにより該当する端末機もしくは回線の状態を遷移させて、該当する端末機もしくは回線を表す図形記号を遷移後の状態を表す図形記号に差し替えることによって動作状況を表示する。

状態遷移表は表示系が管理するこれらの状態と状態遷移を促すタスクとの関係を示した表で、ユーザはこの状態遷移表の内容を書き換えることによってシステムの動作を変更することが可能である。以下状態遷移表について述べる。

表示系は端末機の状態をハンドセットの状態、音の状態、データリンクの状態の三つの状態要素の組合せによって扱っている。また、回線の状態はリソースの状態とデータリンクの状態を結合して扱っている。ここの状態要素の取る値と状態遷移を促すタスクは次のようになる。

ハンドセットの状態

{ :Onhook | :Offhook }

関連タスク

{ ONHOOK, OFFHOOK }

音の状態

{ :Silent | :Dialtone | :Busytone | :Ringing |  
:Ringback | :Interrupt | :Howler | :WorngNo }

関連タスク

{ START }

データリンクの状態

{ :Idle | :Active | :Suspend }

関連タスク

{ CONNECT, DISCONNECT, HOLD, SUSPEND }

リソースの状態

{ :Free | :Hunted }

関連タスク

{ HUNT, RELEASE }

回線の状態としては上記のデータリンクの状態をリソースの状態が:Huntedの時のサブ状態として定義している。また、音の状態については、無条件にタスクSTARTのParameterで与えられた新しい状態に上書きされるように規定されているので状態遷移表には記述しないが、その他の状態の遷移については現状態、タスク、次状態のトリプルによって記述される。現状では次のような状態遷移表が定義されている。

(:Free	HUNT	:Idle)
(:Free	CONNECT	:Active)
(:Idle	CONNECT	:Active)
(:Idle	RELEASE	:Free)
(:Suspend	CONNECT	:Active)
(:Suspend	RELEASE	:Free)
(:Suspend	DISCONNECT	:Idle)
(:Active	DISCONNECT	:Idle)
(:Active	HOLD	:Suspend)
(:Active	SUSPEND	:Suspend)
(:Active	RELEASE	:Free)
(:Onhook	OFFHOOK	:Offhook)
(:Offhook	ONHOOK	:Onhook)

### 3.4.4 タスク管理とイベント／タスク変換

表示系はユーザがCSLによってネットワーク仕様を記述するとき、端末機からもしくは端末機へのイベントとして記述したものに對して実際に起動すべきタスクを決定し処理する。イベント／タスク変換テーブルは表示系が解釈実行するこれらのタスクと仕様中に用いられたイベントとの関係を示したテーブルで、ユーザはこのテーブルの内容を書き換えることにより自由に端末機に對するイベントを定義することができる。以下イベント／タスク変換テーブルについて述べる。

表示系はユーザの端末機操作の結果システムへ的人力として、あるいは処理系用からのイベントを受けその動作状況を表示するために予め規定されたタスクを用いる。これらのタスクはいろいろな意味で用いられるものではなく、それぞれある特定の動作に固有のものとして定義されている。いま、次のようなタスクが規定されている。

#### a. 端末機からの入力

##### 1.(ONHOOK)

受話器が下ろされたことを入力として与えるタスク

##### 2.(OFFHOOK)

受話器が上げられたことを入力として与えるタスク

##### 3.(FLASHHOOK)

端末機がフッキングされたことを入力として与えるタスク

##### 4.(DIGITS number)

端末機からnumberが番号入力されたことを入力として与えるタスク

#### b. 端末機の音の状態制御

##### 1.(START tone-state)

端末機の音に状態をtone-stateの状態に遷移させるためのタスク

#### c. データリンクの状態制御

##### 1.(CONNECT address)

address で与えられたプロセスとの間のデータリンクをアクティブ状態に遷移させるためのタスク

##### 2.(DISCONNECT address)

address で与えられたプロセスとの間のデータリンクをアイドル状態に遷移させるためのタスク

##### 3.(HOLD address)

address で与えられたプロセスとの間のデータリンクを保留状態に遷移させるためのタスク

##### 4.(SUSPEND address)

address で与えられたプロセスとの間のデータリンクを保留状態に遷移させるためのタスク

## d. リソースの状態制御

### 1. (HUNT address)

address で与えられたプロセスとの間のリソースをハントするためのタスク

### 2. (RELEASE address)

address で与えられたプロセスとの間のリソースをリリースするためのタスク

ユーザはこれらのタスクを実行するために用いるイベントを定義し、イベント／タスク変換テーブルに登録することによりネットワーク仕様を任意のイベントで自由に記述することができる。このイベント／タスク変換テーブルは

イベント→タスク (出力動作の表示用) と  
タスク→イベント (システムへの入力用) に

分けることができるが、テーブルに記述する内容としてはある名称のイベント／タスクがあるパラメータを持つときにどういったタスク／イベント群に変換すれば良いのかを記述すれば良いという立場では同等なので、実際には同一のテーブル内に双方の内容を記述する。以下テーブル作成のためのソースフォーマットと変換方式について記述する。

## Source Format

```
(search-key parameter operation-list)
```

ただし、

search-key: テーブルをサーチするためのキーワードでイベントもしくはタスクの名称

parameter: operation-listに記述されたタスクもしくはイベントの集合を展開するとき用いるパラメータのリスト

operation-list: search-keyで与えられたイベントもしくはタスクを変換すべきタスクもしくはイベントの集合について記述したリスト

## 変換方式

表示系は与えられたイベント／タスクの名称を元にテーブルをサーチし、該当するテーブルエントリを探す。次に与えられたパラメータの数と検出したテーブルエントリに記述されたパラメータの数が一致することを調べて変換すべきタスク／イベントの集合に記述された該当する変数の記述を与えられたパラメータの内容に置き換える。

例 イベント／タスク変換テーブルとして次の内容が規定されているとする。

```
(dialtone nil ((start :dialtone)))  
(connect&hold (addr1 addr2) ((connect addr1) (suspend addr2)))
```

(dialtone nil)は(start :dialtone)に変換され

(connect&hold abc xyz)は(connect abc),(suspend xyz)に変換される。

### 3.4.5 Configuration File Format

#### a. 仮想ネットワークの構成について記述する Configuration File

仮想ネットワークの構成を規定するには、どんな名前のプロセスが、どのホストで、どんな処理対象仕様を処理するのかを規定してやれば良い。従って、Configuration file format を次のように規定する。

```
(Csl-System-Configuration)
( Information about Process-a )
( Information about Process-b )
```

```
:
:
```

```
( Information about Process-? )
```

ただし、先頭行はFileの内容を表す識別子で、二行目以降は各プロセスについての記述で具体的には次のようになる。

```
(:Process-id ID :Load-host HOST名 :Target-spec FILES)
```

ただし、

IDはプロセスに付与される識別子

HOST名は処理系を稼働されるホスト名

FILES は処理対象仕様を記述したファイル名をメンバーとしたリストで個々のファイル名はストリング

#### b. 表示画面レイアウトについて記述する Configuration File Format

表示画面の情報を記述するに際しては、どこに端末機やプロセスを配置するのか、どの端末はどのプロセスの支配下にあるのか、各プロセスはどんな図形で表現するのか、仮想ネットワークを構成するどのプロセスを表すのかを規定してやれば良い。また、各端末機にはネットワーク上で固有の電話番号が付与されるので、それも併せて記述するものとする。従って、Configuration File Format を次のように規定する。

```
(Csl-System-Layout-Configuration)
(:Device
( Information about device-1 )
( Information about device-2 )
:
:
( Information about device-# ) )
```

```
(:Process
( Information about process-1 )
( Information about process-2 )
:
:
( Information about process-# ) )
```

ただし、先頭行はFileの内容を表す識別子で、二行目以降は各プロセスまたは端末機についての記述で、第一要素が:Deviceの行は端末機についての記述で、第一要素が:Processの行はプロセスについての記述であり具体的には次のようになる。

## 端末機についての記述

(:Id ID :X X-POS :Y Y-POS :Process-id PID :Phoneno NUM)

ただし、

IDは端末機を表す図の識別子

X-POS は端末機を配置する画面上の X 座標値

Y-POS は端末機を配置する画面上の Y 座標値

PID は端末機を支配下に置く処理系プロセスの識別子

NUM は端末機に付与される電話番号 (String)

## プロセスについての記述

(:Id ID :Symbol SID :X X-POS :Y Y-POS :Process-id PID)

ただし、

IDはプロセスを表す図の識別子

SID はプロセスを表す図形記号の識別子

X-POS はプロセスを配置する画面上の X 座標値

Y-POS はプロセスを配置する画面上の Y 座標値

PID は図が表す処理系プロセスの識別子

### c. 表示に用いる図形について記述する Configuration File Format

表示に用いる図形について記述する Configuration File Format については、表示系が図的にシステムの動作を表示する方法とともに表示に用いるデータの構造、File Format を動作状況の表示のところで記述する。

### 3.5 動作状況の表示

表示系はシステムの動作状況を表示するのに図8に示すような階層的なデータ構造を持つ図表現方式を用いて、図形記号階層のデータを操作することによってシステムの動作を表示する。

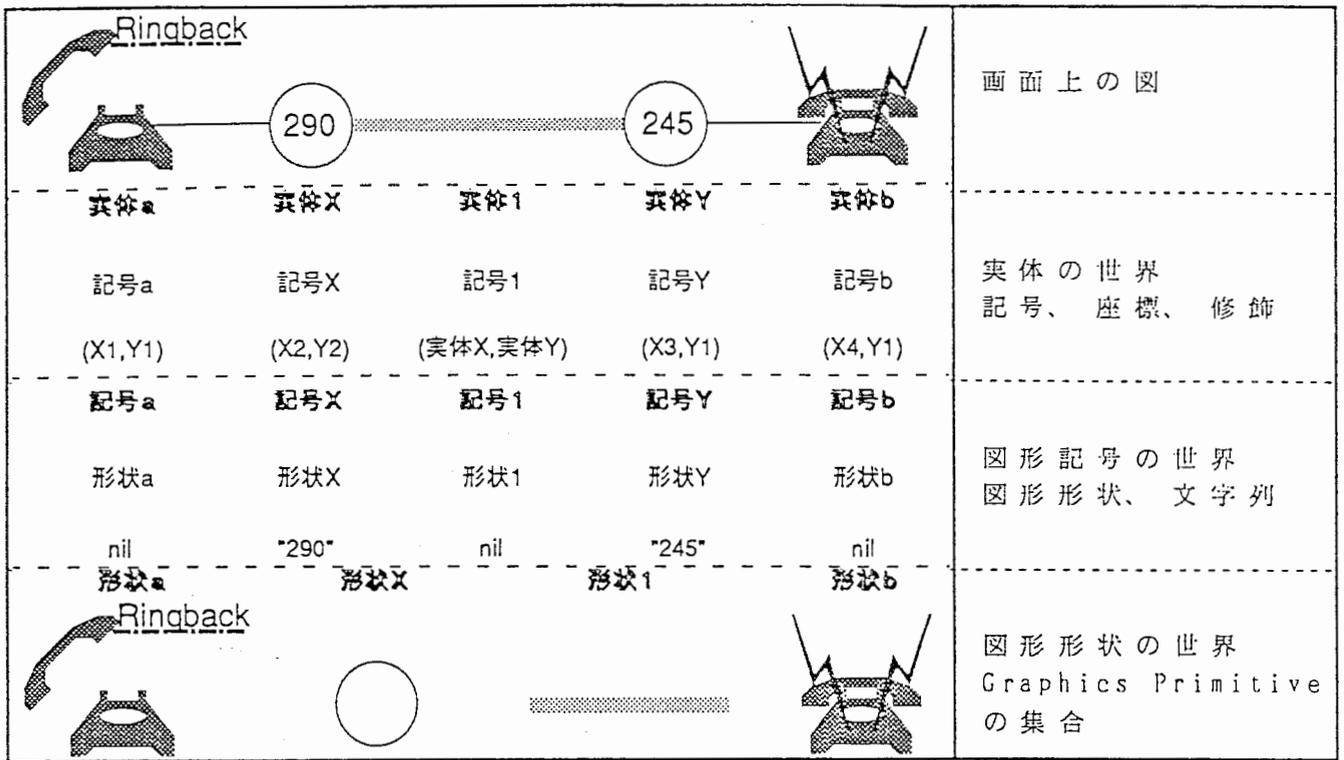


図8 階層的なデータ構造を持つ図表現方式

画面上の図は実体の集合として扱われ、実体は表示に必要な座標値や修飾情報の他その座標位置に配置される図形記号についての記述により構成されている。図形記号は図に記述される単位としての図形形状に関する記述とその記号が表す内容を代表するような文字列より構成されている。図形形状は図に記述される単位を構成するGraphics Primitiveの集合体である。ネットワーク仕様処理システムは仮想ネットワークを構成する要素（端末機やプロセス、回線、データリンク）を実体として対応付け、各種状態の要素を表現するための図形記号を規定し、システムの動作状況に合わせて該当する実体を構成する図形記号を差し替えることによってシステムの動作状況を表示する。

上記図8において示した図を構成している実体aについて、実体を構成する図形記号を記号aから記号cに差し替えることによって表示される図の世界は図9のように表される。

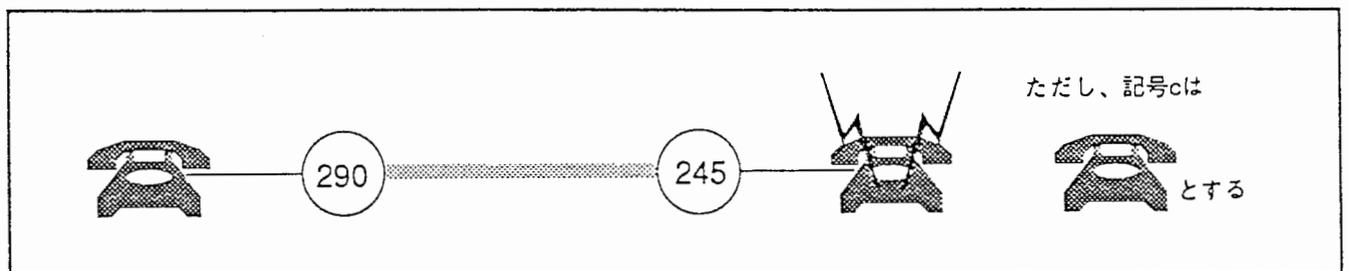


図9 差し替え後の図の世界

具体的には、先に述べた表示画面レイアウトについての記述を元に端末機とプロセスについての実体の世界を構築し、引続き端末機とプロセスの間、各プロセス間に必要な回線やデータリンク、イベントを表示するための実体の世界を構築する。更に、次に述べる図形について記述する Configuration File を元に図形記号や図形形状の世界の構築を行い、図を管理下に置く。  
以下図形について記述する Configuration File Format について規定する。

### Configuration File Format

```
(Csl-System-Symbol-Configuration)
( Information about Node-Symbol )
:
( Information about Node-Shape )
:
( Information about Variable Definition )
:
```

ただし、先頭行は File の内容を表す識別子で、二行目以降は各図形記号または図形形状についての記述もしくはこれらの記述に際して用いる変数の設定に関する記述で、第一要素が Node-Shape の行は図形形状についての記述で、第一要素が Node-Symbol の行は図形記号についての記述であり、第一要素が Defparameter の行は変数の設定に関する記述である。具体的には次のようになる。

#### 図形形状についての記述

```
(Node-Shape :Id ID :Graphics GRAPHICS)
```

ただし、

ID は図形形状を表す識別子

GRAPHICS は図形形状を規定する Graphics Primitive のリスト

#### 図形記号についての記述

```
(Node-Symbol :Id ID :Shape SID :Label-string LABEL
:Handset HANDSET :Datalink DATALINK :Tone TONE)
```

ただし、

ID は図形記号を表す識別子

SID は表示に用いる図形形状を表す識別子

LABEL は表示に用いる文字列 (String) Default は nil

HANDSET は受話器の状態を示すキーワードで

```
{ :ONHOOK | :OFFHOOK | nil } Default は nil
```

DATALINK は電話機の出線状態を示すキーワードで

```
{ :ACTIVE | :SUSPEND | :IDLE | nil } Default は nil
```

TONE は電話機の発する音の状態を表すキーワードで

```
{ :SILENT | :DIALTONE | :RINGING | :RINGBACK |
:BUSYTONE | :INTERRUPT | :HOWLER | :WRONGNO | nil }
```

Default は nil

:Handset :Datalink :Tone の各キーワードによって設定される値は電話機の状態を示すために用いられるものである。従って、図形記号が電話機を表すものでないときにはその設定を必要とせず省略することができる。

## 変数設定に関する記述

(Defparameter VARIABLE VALUE)

ただし、

VARIABLEは規定する変数名

VALUEはその変数にバインドされる値

ユーザはDefparameterによって設定した変数をこのConfiguration file内の記述に自由に用いることができる。その用例としては、図形形状を構成する同一形状のGraphics Primitiveの集合についての記述を変数として登録し、各種図形形状の規定をする記述にはその変数を用いるなどである。また、ユーザは次の変数の値を変更することによって表示に用いられる回線状態やイベントの図形形状を変更することができる。

```
( *transmitted-event-list*,  
  *timeout-event-list*,  
  *waiting-event-list*,  
  *active-connection-list*,  
  *suspended-connection-list*,  
  *reserved-resource-list* )
```

ただし、

\*transmitted-event-list\* は受け渡されるイベントについて  
\*timeout-event-list\* はタイムアウトイベントについて  
\*waiting-event-list\* は応答待ち状態について  
\*active-connection-list\* は通話中の回線について  
\*suspended-connection-list\* は保留中の回線について  
\*reserved-resource-list\* は接続関係について表す図形形状を記述するための変数で、各変数は次の書式で設定しなくてはならない。

(Defparameter 変数 (:line-shape '(line) :arrow 矢先 :line-thickness太さ  
:line-dash-pattern線種 :line-gray-level濃さ))

各記述要素の詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

### 3.6 システムへの入力（端末機操作）

ユーザは図に示された端末機の図をマウスで操作することによりシステムへの入力を行う。いま、システムへの入力としては次のものが提供されており、各々次のような操作によって入力が可能となっている。

#### a. OFFHOOK

オンフック状態の端末機の図（先述の図形に関する Configuration で :Handset のキーワードを :ONHOOK として記述した図形記号により示された図）をマウスの左ボタンでクリックすることによりその端末機の受話器が上げられたことを OFFHOOK 信号でシステムに通知する。

#### b. ONHOOK

オフフック状態の端末機の図（先述の図形に関する Configuration で :Handset のキーワードを :OFFHOOK として記述した図形記号により示された図）をマウスの左ボタンでクリックすることによりその端末機の受話器が下げられたことを ONHOOK 信号でシステムに通知する。

#### c. FLASHHOOK

オフフック状態の端末機の図（先述の図形に関する Configuration で :Handset のキーワードを :OFFHOOK として記述した図形記号により示された図）をシフトキーを押しながらマウスの中ボタンでクリックすることによりその端末機がフッキングされたことを FLASHHOOK 信号でシステムに通知する。

#### d. DIGITS

端末機の図をマウスの中ボタンでクリックした後キーボードより番号 (String) をタイプインすることによりその端末機からダイヤルされたことを DIGITS 信号で与えられた番号をパラメータとしてシステムに通知する。

### 3.7 動作環境設定コマンド

#### a. Init Process Environment

処理系の動作環境を初期化する（再構築する）ためのコマンド

#### Command Syntax

```
Init Process Environment PID :New Process Id new-id
                             :Files file
                             :Host host
```

ただし、

PID は初期化する処理系に付与された識別子

:New Process Id は既に与えられている識別子を変更するために用いるキーワードで、Default は既存の識別子

:Files は既に与えられている処理対象仕様を記述したファイル名をメンバーとしたリストを変更するために用いるキーワードで、Default は既存の処理対象仕様記述ファイルリスト

:Host は処理系を稼働させるホストを変更するために用いるキーワードで、Default は現在稼働中のホスト

#### Function

先に Init System Environment command で構築された仮想ネットワークにおいて、Parameter PID で指定された識別子を持つ処理系の動作環境を再構築する。この時、新たに構築する処理系の動作環境について、各種キーワードを用いて任意の環境に設定することができる。また、先に稼働していた処理系サブシステムは殺され、初期化を指定されたプロセスとなんらかの関係にある各処理系プロセスについてはその実行状態が初期化される。

#### b. Init All Process Environment

処理系の動作環境を初期化する（再構築する）ためのコマンド

#### Command Syntax

```
Init All Process Environment
```

#### Function

先に Init System Environment command で構築された仮想ネットワークを構築する全ての処理系の動作環境を再構築する。先に稼働していた処理系サブシステムは全て殺される。

### c. Init Process

処理系の実行状態を初期化する（リセットする）ためのコマンド

#### Command Syntax

```
Init Process PID
```

ただしPID は初期化する処理系に付与された識別子である。

#### Function

先に Init System Environment command で構築された仮想ネットワークにおいて、Parameter PID で指定された識別子を持つ処理系の実行状態をリセットする。また、初期化を指定されたプロセスとなんらかの関係にある各処理系プロセスについてもその実行状態が初期化される。

### d. Init All Process

処理系の実行状態を初期化する（リセットする）ためのコマンド

#### Command Syntax

```
Init All Process
```

#### Function

先に Init System Environment command で構築された仮想ネットワークを構築する全ての処理系の実行状態をリセットする。

### e. Reload Specification

処理系の処理対象仕様の変更をするためのコマンド

#### Command Syntax

```
Reload Specification PID FILES
```

ただし

PID は処理対象仕様を変更しようとする処理系に付与された識別子

FILES は処理対象仕様を記述したファイル名をメンバーとしたリストで個々のファイル名はストリングで記述するものとする

このParameter は省略可能で省略時には先に Init System Environment command で構築された仮想ネットワークにおいてParameter PID で指定された処理系が処理対象としていたものと同じファイルとする。

#### Function

仮想ネットワークにおいて、Parameter PID で指定した識別子を持つ処理系の処理対象仕様をParameter FILES で指定した内容に変更し、処理系の実行状態を初期化する。また、処理対象仕様の変更を受けたプロセスとなんらかの関係にある各プロセスについてはその実行状態が初期化される。

#### f. Append Specification

処理系の処理対象仕様を追加するためのコマンド

#### Command Syntax

Append Specification PID FILE

ただし、

PID は処理対象仕様を追加しようとする処理系に付与された識別子  
FILEは追加する仕様が記述されたファイル名（ストリング）

#### Function

仮想ネットワークにおいて、Parameter PID で指定した識別子を持つ処理系の処理対象仕様にParameter FILEで指定した内容を追加し、処理系の実行状態を初期化する。また、処理対象仕様の追加を受けたプロセスとなんらかの関係にある各プロセスについてはその実行状態が初期化される。

#### g. Set Transaction Mode To Debug

システムの処理モードをデバッグモードにするためのコマンド

#### Command Syntax

Set Transaction Mode To Debug PID

ただしPID は着目したい処理系に付与された識別子

#### Function

ネットワーク仕様処理システムの処理モードをParameter PID で指定されたプロセスに着目したデバッグモードに切り替える。処理モードはデバッグモードに切り替えられた後、着目している処理系が上記の動作環境もしくは実行状態の初期化コマンドにより初期化された場合には自動的にノーマルモードに戻る。

#### h. Set Transaction Mode To Normal

システムの処理モードをノーマルモードにするためのコマンド

#### Command Syntax

Set Transaction Mode To Normal

#### Function

ネットワーク仕様処理システムの処理モードをノーマルモードにする。

### i. Set Breakpoint

ブレークポイントを設定するためのコマンド

#### Command Syntax

```
Set Breakpoint PID
```

ただし、

PID はブレークポイントを設定しようとする処理系に付与された識別子

#### Function

ノーマルモードで稼働中のネットワーク仕様処理システムの処理モードをユーザとのインタラクションにより設定した状態に達したときにParameter PIDで指定したプロセスに着目したデバッグモードに切り替えるようにブレークポイントを設定する。処理モードがデバッグモードに切り替えられた後、ブレークポイントは自動的に解除される。

ブレークポイントとして設定できる状態とは、Parameter PIDで指定した処理系プロセスが管理する状態/アドレス変数がある値を取ったときを意味し、コマンド実行時にシステムからVARIABLE, VIEW, VALUEの順に問い合わせるのに答えたものが用いられる。

### j. Release Breakpoint

ブレークポイントを解除するためのコマンド

#### Command Syntax

```
Release Breakpoint
```

#### Function

Set Breakpoint commandにより設定したブレークポイントを解除する。

### k. Create State Transition Table

状態遷移表作成のためのコマンド

#### Command Syntax

```
Create State Transition Table FILE
```

ただし、

FILEは状態遷移表を作成するためのソースファイル名

#### Function

Parameter FILEにより指定されたソースファイルの内容にしたがって状態遷移表を作り直す。

## 1. Create Event Conversion Table

イベント／タスク変換テーブル作成のためのコマンド

### Command Syntax

Create Event Conversion Table FILE

ただし、

FILEはイベント／タスク変換テーブルを作成するためのソースファイル名

### Function

Parameter FILEにより指定されたソースファイルの内容にしたがってイベント／タスク変換テーブルを作り直す。

### 3.8 デバッグモードにおける処理系の実行状態制御用コマンド

ネットワーク仕様処理システムにはノーマルモードとデバッグモードの二つの処理モードがあることは既に述べた。ノーマルモードで稼働するシステムはユーザからの入力を解釈し、仮想ネットワークの動作状況を表示するものであるが、デバッグモードで稼働するシステムはユーザからの入力を解釈するだけでなくその解釈結果をユーザとのインタラクションによって制御することができる。以下それらの制御用コマンドについて規定する。

#### a. Edit State Variables

処理系が管理する状態変数 / アドレス変数の持つ値を変更するためのコマンド

##### Command Syntax

```
Edit State Variables VARIABLE VIEW VALUE
```

ただし、

VARIABLEは値を変更したい状態変数 / アドレス変数名

VIEWはVARIABLEが参照される処理パート

VALUEは変更したい値

##### Function

デバッグモードで稼働中に着目している処理系の管理するParameter VIEWで指定した処理パートで参照しているParameter VARIABLEで指定した状態変数 / アドレス変数の値をParameter VALUEで指定した値に変更する。

#### b. Send Event

仮想ネットワーク上に疑似的にイベントを流すためのコマンド

##### Command Syntax

```
Send Event TO EVENT FROM PARAMETERS
```

ただし、

TOは疑似イベントを受け取る処理系の識別子

EVENTは疑似的に流したいイベント名

FROMは疑似イベントを送出したようにみせる処理系の識別子

PARAMETERSは疑似イベントにつけてやるべきパラメータ

##### Function

仮想ネットワーク上に存在するParameter TOで指定された識別子を持つ処理系の同期部にParameter FROMで指定した識別子を持つ処理系からParameter EVENTで指定したイベントがParameter PARAMETERSで指定したパラメータを持って返されたように操作するため、該当するイベント式を作成し仮想ネットワーク上に流す。

### c. Send Internal Event

処理系内部で発生すべきイベントを疑似的に送ってやるためのコマンド

#### Command Syntax

Send Internal Event PID EVENT FROM-PART TO-PART PARAMETERS

ただし、

PID は疑似的に発生させた内部イベントを受け取る処理系の識別子

EVENT は疑似的に発生させる内部イベント名

FROM-PART は内部イベントが発生したようにみせる処理パート

TO-PART は内部イベントを受け取る処理パート

PARAMETERSは内部イベントにつけてやるべきパラメータ

#### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系に内部でParameter FROM-PART で指定した処理パートからParameter TO-PART で指定された処理パートへParameter EVENT で指定されたイベントがParameter PARAMETERSで指定されたパラメータを持って送られたように操作するため、該当するイベント式を作成し仮想ネットワーク上に流す。

### d. Invoke Transaction

デバッグモードで稼働中に処理対象が移行した直後に中断している処理の継続を指示するためのコマンド

#### Command Syntax

Invoke Transaction

#### Function

デバッグモードで着目しているステップ処理モードで稼働中の処理系が処理対象パートを移行した直後に処理を中断している時、その処理の継続を指示する。

#### e. Unbound Event

処理系が制約評価処理の対象としているイベントをリセットするためのコマンド

#### Command Syntax

```
Unbound Event PID EVENT PART FROM
```

ただし、

PID はリセットしたいイベントを管理する処理系オブジェクトに付与された識別子

EVENT はリセットしたイベント名

PART はリセットしたいイベントを処理の対象としている処理パート

FROM はリセットしたいイベントを送出した処理系の識別子もしくは処理パート

#### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系のParameter PARTで指定した処理パートで処理の対象となっているParameter FROMで指定された処理系／処理パートから送られたParameter EVENT で指定したイベントをリセットする。

### 3.9 その他のコマンド

#### a.Show Process Environment

処理系の動作環境をみるためのコマンド

##### Command Syntax

```
Show Process Environment PID
```

ただし、

PID は動作環境をみたい処理系オブジェクトに付与された識別子

##### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系の動作環境を表示する。

#### b.Show State Transition Table

状態遷移表の内容をみるためのコマンド

##### Command Syntax

```
Show State Transition Table
```

##### Function

状態遷移表の内容を表示する。

#### c.Show Event Conversion Table

イベント/タスク変換テーブルの内容をみるためのコマンド

##### Command Syntax

```
Show Event Conversion Table
```

##### Function

イベント/タスク変換テーブルの内容を表示する。

#### d.Show Breakpoint

ブレークポイントの設定内容をみるためのコマンド

##### Command Syntax

```
Show Breakpoint
```

##### Function

設定されているブレークポイントに関する情報を表示する。

#### e. Insert Comment Line

イベント表示ペインもしくは発火制約表示ペインにコメント行を挿入するためのコマンド

#### Command Syntax

```
Insert Comment Line COMMENT :Into History PANE
```

ただし、

COMMENT は挿入したいコメント行

PANEはイベント表示ペインもしくは発火制約表示ペイン

{ Event-History | Fired-Rule }

Event-History はイベント表示ペイン

Fired-Ruleは発火制約表示ペイン

Default はEvent-History

#### Function

Parameter PANEで指定された表示ペインの最終行にParameter COMMENT で与えられたコメント行を挿入する。

#### f. Hardcopy Event History

イベント表示ペインに表示された内容をハードコピーするためのコマンド

#### Command Syntax

```
Hardcopy Event History
```

#### Function

イベント表示ペインに表示された内容をハードコピーする。

#### g. Hardcopy Fired Rule History

発火制約表示ペインに表示された内容をハードコピーするためのコマンド

#### Command Syntax

```
Hardcopy Fired Rule History
```

#### Function

発火制約表示ペインに表示された内容をハードコピーする。

#### h. Clear Event History

イベント表示ペインに表示された内容を消去するためのコマンド

#### Command Syntax

```
Clear Event History
```

#### Function

イベント表示ペインの内容を消去しイベント表示ペインをリフレッシュする。

#### i. Clear Fired Rule History

発火制約表示ペインに表示された内容を消去するためのコマンド

#### Command Syntax

Clear Fired Rule History

#### Function

発火制約表示ペインの内容を消去し発火制約表示ペインをリフレッシュする。

#### j. Refresh Display

システム動作表示ペインの内容を書き直すためのコマンド

#### Command Syntax

Refresh Display

#### Function

システム動作表示ペインに表示された内容を書き直しシステム動作表示ペインをリフレッシュする。

#### k. Move Chart

システム動作表示ペインに表示されたプロセスや端末機の図を移動させるためのコマンド

#### Command Syntax

Move Chart CID

ただし、

CID は移動させたい図に与えられた識別子

#### Function

Parameter CID で指定された識別子を持つ図をユーザとのインタラクションで与えられた新しい座標位置に移動させて表示する。

## § 4 システムの構成

システムの概要のところでも述べたように、ネットワーク仕様処理システムは処理系と表示系の二つのサブシステムから構成されていて、複数の処理系が一つの表示系をネットワークコントローラとしたスター型接続で接続されている。以下各サブシステムの役割と構成について記述する。

### 4.1 処理系サブシステム

処理系サブシステムは仮想ネットワークを構成するプロセス毎に起動される。従って、処理系の役割は仮想ネットワークを構成する一つのプロセスの動作について記述された処理対象仕様を元に外部より与えられるイベントを解釈し、出力動作を決定実行することである。

処理系はこの役割を果たすために連続処理モード（ノーマルモード）とステップ処理モード（デバッグモード）の二つの処理モードを提供する。連続処理モードが与えられたイベントを外部とのインタラクションなしで連続的に解釈実行するのに対して、ステップ処理モードでは与えられたイベントを外部とのインタラクションにより解釈実行するという大きな違いがあるが、基本となるイベントの処理方式（解釈ロジック）については何等変わるものではない。

CSLの言語仕様並びに処理系の解釈ロジックの詳細については『ネットワーク仕様記述言語処理系プログラム』（外部仕様書第10節言語仕様）を参照のこと。以下両モードで共通の処理系の動作環境設定方式とステップ処理モードにおける外部とのインタラクションをはじめ処理系を操作する内部コマンドについて記述する。

#### 4.1.1 処理系の動作環境設定

処理系の動作環境はCSLというシステム（3.1節参照）がロードされたホストと表示系の間で内部コマンドを用いて一つのオブジェクトとして作成され、そのオブジェクトが保持する各種情報を操作することで設定される。以下処理系の動作環境設定のために表示系から送受する内部コマンドについて記述する。

##### a. Server-Host

処理系の動作環境を設定するために新しいオブジェクトを作成するためのコマンド

##### Command Syntax

(Server-Host Server PID)

ただし、

Serverはオブジェクトを管理する表示系の稼働するホスト名

PIDはオブジェクトが対応させられる仮想ネットワーク内のプロセスに付与された識別子

##### Function

Parameter Serverで指定したホストから受けるもので、そのホストで稼働する表示系をネットワークコントローラとした仮想ネットワーク中でParameter PIDで指定された識別子を持つプロセスとして動作するための処理系オブジェクトを新規に作成する。また、処理系はオブジェクトが新規に作成されたことを内部コマンドClient-Readyにより表示系に報告する。

#### b. Load-Spec

処理系に処理対象仕様を指示し動作環境を初期化するためのコマンド

#### Command Syntax

(Load-Spec PID FILES)

ただし、

PID は先に Server-Host command でオブジェクトに付与された識別子 FILES は処理対象仕様が記述されたファイル名をメンバーとするリスト

#### Function

仮想ネットワーク上に存在する Parameter PID で指定された識別子を持つ処理系オブジェクトに対して処理対象仕様を Parameter FILES で指定し、処理系オブジェクトの動作環境を初期化する。また、処理系はオブジェクトの動作環境が初期化されたことを内部コマンド Service-Enableにより表示系に報告する。

#### c. Append-Load-Spec

処理系の処理対象仕様を追加し動作環境を初期化するためのコマンド

#### Command Syntax

(Append-Load-Spec PID FILES)

ただし、

PID は先に Server-Host command でオブジェクトに付与された識別子 FILES は処理対象仕様が記述されたファイル名をメンバーとするリスト

#### Function

仮想ネットワーク上に存在する Parameter PID で指定された識別子を持つ処理系オブジェクトに対して Parameter FILES で指定された仕様記述を処理対象仕様として追加し、処理系オブジェクトの動作環境を初期化する。また、処理系はオブジェクトの動作環境が初期化されたことを内部コマンド Service-Enableにより表示系に報告する。

#### d. Init-System

処理系の実行状態を初期化するためのコマンド

#### Command Syntax

(Init-System PID)

ただし、

PID は先に Server-Host command でオブジェクトに付与された識別子

#### Function

仮想ネットワーク上に存在する Parameter PID で指定された識別子を持つ処理系オブジェクトの実行状態を初期化する。また、処理系はオブジェクトの実行状態が初期化されたことを内部コマンド Service-Enableにより報告する。

### e. Kill

処理系の終了のためのコマンド

#### Command Syntax

(Kill PID)

ただし、

PID は先に Server-Host command でオブジェクトに付与された識別子

#### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系オブジェクトを仮想ネットワーク上から消去する（表示系の管理対象オブジェクトから外す）。

### f. Set-Mode

処理系の処理モードの切り替えを行うためのコマンド

#### Command Syntax

(Set-Mode PID MODE)

ただし、

PID は先に Server-Host command でオブジェクトに付与された識別子  
MODEは切り替えるモードの指定 ( Normal | Debug )

Normalは連続処理モード

Debug はステップ処理モードを表す

#### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系オブジェクトの処理モードをParameter MODEで指定したモードに切り替える。

一連の処理系の動作環境設定の流れを図10に示す。

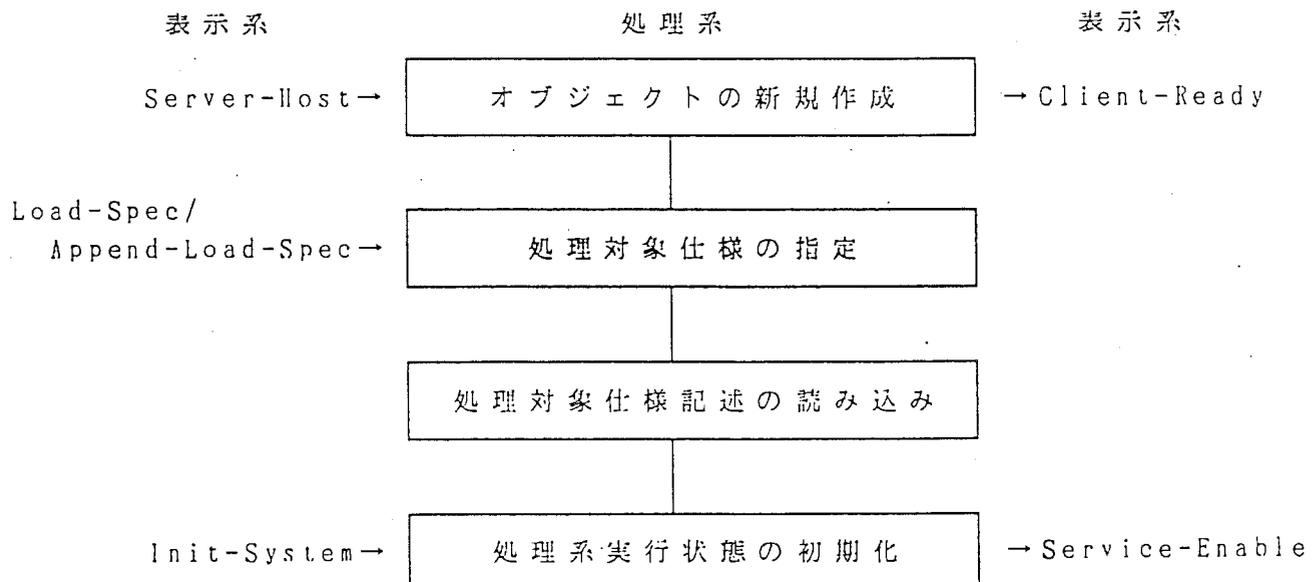


図10 処理系動作環境設定の流れ

処理系の動作環境設定に際しての正常な流れについて記述したが、上記の流れの中でエラーが発生した場合の詳細については『ネットワーク仕様記述言語処理プログラム』（外部仕様書第9節 ネットワークコントロール部）を参照のこと。

#### 4.1.2 ステップ処理モードにおける外部とのインタラクション

C S Lで記述される処理対象仕様は分配部、機能部、同期部の三つのパートに分けて記述されており、処理系はその制約評価の対象を分配部、機能部、同期部の順に順次回すことによって仕様記述を解釈実行する。

連続処理モードでは制約評価の対象をインタラクションなしに順次移して、発火条件の評価の結果発火すべきと判定される制約式の記述にしたがって実行動作を無条件に行う。

ステップ処理モードでは上記の一連の制約評価の手順の制約評価の対象を移して発火条件の評価を開始する前にその処理を中断し、外部（表示系）よりの継続指示により処理を継続する。また、発火条件の評価の結果発火すべきと判定される制約式についてもその発火処理の実行を外部（表示系）に問い合わせ、表示系からの指示にしたがって処理する。

以下、その制御方法について記述する。

##### 4.1.2.1 制約評価部の移行に際しての処理の制御

ステップ処理モードで稼働する処理系はその制約評価の対象を分配部から機能部、機能部から同期部、同期部から分配部へ移行した直後に表示系に評価の対象が移行したことをExecution-Part commandにより通知し、制約評価処理を中断する。その後表示系からのInvoke-Next-Step commandによって制約評価処理の継続指示がある迄は一切の制約評価処理を行わない。ただし、その間表示系から送られて来る各種コマンドについての処理は行うものとする。

以下、制約評価部の移行の制御用の内部コマンドについて記述する。

##### a. Execution-Part

ステップ処理モードで稼働中の処理系がその制約評価の対象を移行した直後に制約評価を中断していることを表示系に通知するためのコマンド

##### Command Syntax

(Execution-Part PID PART)

ただし、

PIDは処理系オブジェクトに付与された識別子

PARTは制約評価が移されたパートを表すキーワードで次の値を取る

( idling | decomposing | functional superposing )

idlingは制約評価の対象が入力待ち状態であることを表す

decomposing は制約評価の対象が分配部に移行したことを表す

functionalは制約評価の対象が機能部に移行したことを表す

superposing は制約評価の対象が同期部に移行したことを表す

##### Function

ステップ処理モードで稼働中の処理系から表示系に送られるコマンドで、コマンドを送出後表示系より処理継続指示があるまで制約評価処理を中断する。詳細な情報としてはParameter PIDで指定された処理系がParameter PARTで指定したパートに制約評価の対象を移行し、処理を中断していることを表す。

## b. Invoke-Next-Step

制約評価の対象を移行した直後に処理を中断している処理系に処理の継続を指示するためのコマンド

### Command Syntax

(Invoke-Next-Step PID)

ただし、

PID は処理系オブジェクトに付与された識別子

### Function

制約評価の対象を移行した直後に処理を中断しているParameter PIDで指定したステップ処理モードで稼働中の処理系に制約評価処理の継続を指示する。Parameter PIDで指定した処理系が連続処理モードで稼働中か制約評価の対象を移行した直後に処理を中断している時以外はこのコマンドは無視される。

#### 4.1.2.2 発火すべき制約式の発火処理の制御

ステップ処理モードで稼働する処理系はイベントを処理する過程で発火条件の評価の結果発火すべきと判断した制約式をFire-Rule commandにより表示系に通知し、制約評価処理を中断する。その後表示系からのCollect-Rule commandもしくはCancel-Rule command, Skip-Rule commandによって該当する制約式に関する発火処理を決定し、制約評価処理を継続する。中断中に処理モードの変更要求がなされた場合は該当する制約式に関する発火処理が採択されたものとして処理した後に処理モードの変更を行う。また、この間に各種処理系の処理状態を操作するためのコマンドによって処理系の状態を操作しても該当する制約式にはその影響が及ばないものとする。

以下、制約式の発火処理制御用の内部コマンドについて記述する。

## a. Fire-Rule

ステップ処理モードで稼働中の処理系がイベントを処理する過程で発火条件の評価の結果発火すべきと判断した制約式を表示系に通知するためのコマンド

### Command Syntax

(Fire-Rule PID CONSTRAINT TEMP-VALUE)

ただし、

PID は処理系オブジェクトに付与された識別子

CONSTRAINTは発火条件の評価の結果発火すべきと判断された制約式

TEMP-VALUEは上記の制約式の中で用いられている一時変数の取る値について記述されたリスト

### Function

ステップ処理モードで稼働中の処理系から表示系に送られるコマンドで、処理系がイベントを処理する過程で発火条件の評価の結果発火すべきと判断される制約式を表示系に通知し、表示系からの継続指示があるまで処理を中断する。詳細な情報としてはParameter PIDで指定された処理系がイベントの処理の過程でParameter CONSTRAINTで指定した制約式が発火条件の評価の結果発火すべきと判断しその時該当する制約式で用いられる一時変数の値をParameter TEMP-VALUEで指定する。

#### b. Collect-Rule

発火処理の問い合わせに対して制約式の採択を指示するためのコマンド

#### Command Syntax

(Collect-Rule PID)

ただし、

PID は処理系オブジェクトに付与された識別子

#### Function

Parameter PID で指定した処理系が Fire-Rule command で問い合わせてきた制約式の発火処理について、該当する制約式を採択することを処理系に通知し、評価処理の継続指示を行う。

#### c. Cancel-Rule

発火処理の問い合わせに対して制約式の廃棄を指示するためのコマンド

#### Command Syntax

(Cancel-Rule PID)

ただし、

PID は処理系オブジェクトに付与された識別子

#### Function

Parameter PID で指定した処理系が Fire-Rule command で問い合わせてきた制約式の発火処理について、該当する制約式を廃棄することを処理系に通知し、評価処理の継続指示を行う。処理系は廃棄の指示を受けると発火が予定されていた制約式の評価を取りやめるが、制約評価後の後処理については同様に行う。即ち、処理の廃棄指示は処理の流れの上では制約その物は発火したのと同等の扱いをし、制約の評価の結果実行される動作記述の内容を無にし処理を継続する。

#### d. Skip-Rule

発火処理の問い合わせに対して制約式のスキップを指示するためのコマンド

#### Command Syntax

(Skip-Rule PID)

ただし、

PID は処理系オブジェクトに付与された識別子

#### Function

Parameter PID で指定した処理系が Fire-Rule command で問い合わせてきた制約式の発火処理について、該当する制約式をなかったものとすることを処理系に通知し、評価処理の継続指示を行う。処理系はスキップの指示を受けると発火が予定されている制約式についてあたかも発火が予定されていなかったものとして扱う。即ち、一切の後処理を含めて該当する制約式の評価を行わずに処理を継続する。

#### 4.1.3 処理系の動作環境／実行状況問い合わせ

ネットワークコントローラとしての表示系は必要に応じて処理系の実行状況、動作環境をユーザに示してやらなければならない。  
そこで、次のような動作環境／実行状況の問い合わせおよび報告用のコマンドを規定する。

##### a. Show-Constraints

処理系が処理の対象としている仕様記述の問い合わせのためのコマンド

##### Command Syntax

(Show-Constraints PID PART)

ただし、

PID は処理系オブジェクトに付与された識別子

PARTは仕様記述を問い合わせるパート

{ decomposing | functional | superposing }

decomposing は分配部

functional は機能部

superposing は同期部である

##### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系が処理の対象としているParameter PARTで指定したパートの仕様記述の問い合わせを行う。問い合わせに対して処理系はConstraints command で応答する。

##### b. Show-Exec-Part

処理系が処理中のパートを問い合わせるためのコマンド

##### Command Syntax

(Show-Exec-Part PID)

ただし、

PID は処理系オブジェクトに付与された識別子

##### Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系の処理中のパートの問い合わせを行う。

問い合わせに対して処理系はCurrent-Exec-Part command で応答する。

##### c. Show-Variables-Value

処理系が管理している状態変数／アドレス変数の持つ値を問い合わせるためのコマンド

##### Command Syntax

(Show-Variables-Value PID)

ただし、

PID は処理系オブジェクトに付与された識別子

## Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系が管理する状態変数／アドレス変数の持つ値の問い合わせを行う。問い合わせに対して処理系はVariables-Value command で応答する。

## d. Show-Bound-Events

処理系が制約処理の条件評価の対象としているイベントを問い合わせるためのコマンド

## Command Syntax

(Show-Bound-Events PID PART)

ただし、

PID は処理系オブジェクトに付与された識別子

PARTは評価対象イベントを問い合わせるパート

( decomposing | functional | superposing )

decomposing は分配部

functional は機能部

superposing は同期部である

## Function

仮想ネットワーク上に存在するParameter PID で指定された識別子を持つ処理系のParameter PARTで指定された処理パートで評価の対象となっているイベントの問い合わせを行う。

問い合わせに対して処理系はCurrent-Bound-Events commandで応答する。

## e. Constraints

表示系からShow-Constraints commandによって問い合わせられた処理対象仕様を表示系に送り返すためのコマンド

## Command Syntax

(Constraints PID PART CONSTRAINTS)

ただし、

PID は処理系オブジェクトに付与された識別子

PARTは返答する仕様記述を持つパート

( decomposing | functional | superposing )

decomposing は分配部

functional は機能部

superposing は同期部である

CONSTRAINTS は0個以上の任意の数の制約式

## Function

表示系からの問い合わせに対してParameter PID で指定した識別子を持つ処理系のParameter PARTで指定した処理パートはParameter CONSTRAINTS で指定された内容の制約を評価の対象としていることを返答する。

#### f.Current-Exec-Part

表示系から Show-Exec-Part command によって問い合わせされた処理系の処理パートを表示系に送り返すためのコマンド

#### Command Syntax

(Current-Exec-Part PID PART)

ただし、

PID は処理系オブジェクトに付与された識別子

PART は処理系が処理中のパート

{ idling | decomposing | functional | superposing }

idling は入力待ち

decomposing は分配部

functional は機能部

superposing は同期部である

#### Function

表示系からの問い合わせに対して Parameter PID で指定した識別子を持つ処理系が Parameter PART で指定された処理パートを処理中であることを表示系に返答する。

#### g.Variables-Value

表示系から Show-Variables-Value command によって問い合わせされた処理系の管理する状態変数 / アドレス変数の値を表示系に送り返すためのコマンド

#### Command Syntax

(Variables-Value PID VARIABLE-LIST)

ただし、

PID は処理系オブジェクトに付与された識別子

VARIABLE-LIST は 0 個以上任意の数の変数について記述されたリストで各リストは次の形をしている

(Variable View Value)

ただし、

Variable は変数名

View は変数の持つ VIEW で

{ global | decomposing |  
functional | superposing | interactive }

Value は変数の取る値

#### Function

表示系からの問い合わせに対して Parameter PID で指定された識別子を持つ処理系が Parameter VARIABLE-LIST で指定した変数を管理し、その時点での値を表示系に返答する。

#### h.Current-Bound-Events

表示系から Show-Bound-Events command によって問い合わせされた処理系の制約評価処理対象イベントを表示系に送り返すためのコマンド

(Current-Bound-Events PID PART EVENT-LIST)

ただし、

PID は処理系オブジェクトに付与された識別子

PART は制約評価処理対象イベントが記述されたパート

{ decomposing | functional | superposing }

decomposing は分配部

functional は機能部

superposing は同期部である

EVENT-LIST は 0 個以上の任意の数の処理対象となっているイベントについて記述されたリストで各リストは次の形をしている

(Event From Value)

ただし、

Event はイベント変数名

From はイベントの送出元

Value は 0 個以上の任意の数のパラメータの値

#### Function

表示系からの問い合わせに対して Parameter PID で指定された識別子を持つ処理系の Parameter PART で指定した処理パートでは Parameter EVENT-LIST で指定されたイベントをイベント処理対象としていることを表示系に返答する。

#### 4.1.4 処理系の実行状況操作

処理系は管理下に置く各種状態変数 / アドレス変数の持つ値と内外から送られて来るイベントによってその動作を決定している。以下通常の処理によるもの以外に外部から強制的に処理系の管理状況を操作変更するための内部コマンドについて記述する。

##### a. Unbound-Event

処理系にイベントのリセットを命ずるためのコマンド

#### Command Syntax

(Unbound-Event PID EVENT-NAME PART FROM)

ただし、

PID は処理系オブジェクトに付与された識別子

EVENT-NAME はリセットしたいイベントの名称

PART はリセットしたいイベントを処理する処理パート

FROM はリセットしたいイベントを送り主

#### Function

Parameter PID で指定された識別子を持つ処理系の Parameter PART で指定された処理パートで処理される Parameter FROM で指定されたところから送ってこられた Parameter EVENT-NAME で指定したイベントをリセットする。

## b. Set-Variables-Value

処理系の管理する状態変数 / アドレス変数の持つ値を変更するためのコマンド

### Command Syntax

```
(Set-Variables-Value PID VARIABLE-LIST)
```

ただし、

PID は処理系オブジェクトに付与された識別子

VARIABLE-LIST は 0 個以上任意の数の変数について記述されたリストで各リストは次の形をしている

```
(Variable View Value)
```

ただし、

Variable は変数名

View は変数の持つ VIEW で

```
( global | decomposing |  
functional | superposing | interactive )
```

Value は変数の取る値

### Function

Parameter PID で指定された識別子を持つ処理系の管理する状態変数 / アドレス変数を Parameter VARIABLE-LIST で指定した内容に変更する。

## c. Event

処理系に与えられるイベントを送るためのコマンド

### Command Syntax

```
(Event EVENT-NAME TO-PID TO-PART FROM-PID FROM-PART  
OWNER-PID MESSAGE# TIMING PARAMETERS)
```

ただし、

EVENT-NAME は与えたいイベントの名称

TO-PID はイベントを受け取る処理系の持つ識別子

TO-PART はイベントを受け取る処理系の処理パートで

```
( dcmp | func | supr | unknown )
```

dcmp は分配部

func は機能部

supr は同期部

unknown は不明を表す

FROM-PID はイベントを送った処理系の持つ識別子

FROM-PART はイベントを送った処理系の処理パートで

```
( dcmp | func | supr | unknown )
```

dcmp は分配部

func は機能部

supr は同期部

unknown は不明を表す

OWNER-PID は一連のイベントに付与される識別子

MESSAGE# はイベントに付与されるシーケンスナンバー

TIMING はイベントを送った後の応答待ち時間

PARAMETERS はその他必要に応じた 0 個以上の任意の数のパラメータ

## Function

強制的に処理系の管理状況を変更するためだけではなく、処理系間の通常のイベントの受渡しにも用いられるコマンドでParameter FROM-PIDで指定された識別子を持つ処理系のParameter FROM-PARTで指定された処理パートからParameter TO-PIDで指定した識別子を持つ処理系のParameter TO-PARTで指定した処理パートにParameter EVENT-NAMEで指定した名称のイベントをParameter PARAMETERSで指定したパラメータとともに送る。

## d. Event-Unbound

処理系の処理対象としていたイベントがリセットされたことを表示系に通知するためのコマンド

## Command Syntax

```
(Event-Unbound EVENT-NAME TO-PID TO-PART FROM-PID FROM-PART  
OWNER-PID MESSAGE# TIMING PARAMETERS)
```

ただし、

EVENT-NAMEはリセットしたイベントの名称

TO-PIDはイベントを受け取った処理系の持つ識別子

TO-PARTはイベントを受け取った処理系の処理パートで

{ dcmp | func | supr | unknown }

dcmp は分配部

func は機能部

supr は同期部

unknown は不明を表す

FROM-PIDはイベントを送った処理系の持つ識別子

FROM-PARTはイベントを送った処理系の処理パートで

{ dcmp | func | supr | unknown }

dcmp は分配部

func は機能部

supr は同期部

unknown は不明を表す

OWNER-PIDは一連のイベントに付与される識別子

MESSAGE#はイベントに付与されるシーケンスナンバー

TIMINGはイベントを送った後の応答待ち時間

PARAMETERSはその他必要に応じた0個以上の任意の数のパラメータ

## Function

Parameter FROM-PIDで指定された識別子を持つ処理系のParameter FROM-PARTで指定された処理パートからParameter TO-PIDで指定した識別子を持つ処理系のParameter TO-PARTで指定した処理パートにParameter PARAMETERSで指定したパラメータを持ったParameter EVENT-NAMEで指定した名称のイベントをリセットしたことを表示系に通知する。

#### e.Result-Set-Variables-Value

Set-Variables-Value command によって変更指示を受けた変数の設定変更完了を表示系に通知するためのコマンド

#### Command Syntax

(Result-Set-Variables-Value PID VARIABLE-LIST)

ただし、

PID は処理系オブジェクトに付与された識別子

VARIABLE-LIST は 0 個以上任意の数の変数について記述されたリストで各リストは次の形をしている

(Variable View Value)

ただし、

Variable は変数名

View は変数の持つ VIEW で

( global | decomposing |

functional | superposing | interactive )

Value は変数の取る値

#### Function

Parameter PID で指定された識別子を持つ処理系が Parameter VARIABLE-LIST で指定したような内容に管理する状態変数 / アドレス変数の値を変更したことを表示系に通知する。

その他内部コマンドの詳細については『ネットワーク仕様記述言語処理プログラム』（外部仕様書第 9 節 ネットワークコントロール部）を参照のこと。

## 4.2 表示系サブシステム

表示系サブシステムはネットワーク仕様処理システムにおいて仮想ネットワークの管理、ユーザからの入力の受け付け、画面管理の役割を果たす。ユーザからの入力受け付けについては前章取り扱い説明のところで既に述べてあるので、ここでは仮想ネットワークの管理と画面管理についてその管理方法を記述する。

### 4.2.1 仮想ネットワーク管理

仮想ネットワークは固有の識別子を持った処理系オブジェクトが表示系をネットワークコントローラとしたスター型の接続によって構成されている。ネットワークコントローラとして表示系は各処理系に対して個々に該当する処理系オブジェクトに対するアクセスストリーム、付与された識別子、稼働しているホスト名、処理対象となっているネットワーク仕様等の情報を管理し、処理系間で処理系オブジェクトに付与された識別子をキーワードに送出されるイベントを正しく該当処理系オブジェクトに配送する。

表示系は仮想ネットワークを管理するに際して次のようなフレーバーを定義している。

```
Flavor NW-MANAGER
  ((Process-id)
   (Load-host)
   (Target-spec)
   (Access-stream)
   (Service-status))
```

ただし、

NW-MANAGERはフレーバー名

Process-idは処理系オブジェクトに付与された識別子

Load-hostは処理系オブジェクトが稼働しているホスト名

Target-specは処理対象ネットワーク仕様が記述されたファイル名

Access-streamはネットワークを経由してI/Oを行うためのStream

Service-statusは処理系オブジェクトの状態

表示系は仮想ネットワークの構成に関するConfiguration Fileに記述された内容に従い、各処理系に対してインスタンスを作成し必要な情報を書き込み、管理下に置くべきインスタンスを\*CSL-SYSTEM-CONFIGURATION-TABLE\*という名の変数に格納している。インスタンスの作成手順等の詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

### 4.2.2 画面管理

表示系は仮想ネットワークの構成の表示、仮想ネットワークの動作状況の表示を目的とした画面管理を行っている。画面管理は仮想ネットワークの構成を表示するための図の作成管理、仮想ネットワークの動作状況を表示するための図要素の作成管理および図要素の運用管理をすることによって行われる。以下、図の作成管理、図要素の作成管理、図要素の運用管理について記述する。

#### 4.2.2.1 仮想ネットワークの構成を表示するための図の作成管理

表示系は表示画面のレイアウトに関する Configuration File に記述された内容に従い、端末機やプロセスを表示するための実体を作成し、各プロセス間には二本のイベントを表示するための実体と一本の回線を表示するための実体を作成する。また、端末機とその端末機を制御するプロセスの間にも二つのイベントを表示するための実体を作成される。作成した端末機やプロセスを表す実体は \*NODE-STRUCT-MANAGER\* という名の変数に格納され、イベントや回線を表す実体は \*ARC-STRUCT-MANAGER\* という名の変数に格納されている。また、端末機を表す実体とそれを制御するプロセスを処理する処理系に付与された識別子との対応表を作成し、\*DEVICE-PROCESS-MANAGER\* という名の変数に格納し、端末機に付与された電話番号と端末機を制御するプロセスを処理する処理系に付与された識別子の対応表を作成し、\*NUMBER-TRANSCRIPTION-TABLE\* という名の変数に格納する。更にプロセスを表す実体とそれを処理する処理系に付与された識別子との対応表を作成し、\*PROCESS-ID-MANAGER\* という名の変数に格納する。その他図の作成手順、各変数のフォーマット等の詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

#### 4.2.2.2 仮想ネットワークの動作状況を表示するための図要素の作成管理

表示系は表示に用いる図形に関する Configuration File の内容に従い、各種状態を表す端末機やプロセスを表示するための図形記号や図形形状を作成し、作成した図形記号を \*NODE-SYMBOL-MANAGER\* という名の変数に格納し、同じく図形形状を \*NODE-SHAPE-MANAGER\* という名の変数に格納する。また、図形記号とその記号が表す状態の対応表を作成し、\*SYMBOL-STATE-TABLE\* という名の変数に格納する。更に予め規定された内容に従いイベントや回線を表示するための図形記号や図形形状を作成し、作成した図形記号を \*ARC-SYMBOL-MANAGER\* という名の変数に格納し、同じく図形形状を \*ARC-SHAPE-MANAGER\* という名の変数に格納する。その他図要素の作成手順、各変数のフォーマット等の詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

#### 4.2.2.3 仮想ネットワークの動作状況を表示するための図要素の運用管理

表示系は実体で表された仮想ネットワークの動作状況の変化を実体を構成する図形記号を差し替えることによって表現する。以下表示系が図要素の運用管理をする際に用いる該当する実体/図形記号を特定する方法、該当する実体を構成する図形記号を差し替える方法について記述する。

##### a. 実体を特定する方法

端末機やプロセスを表す実体は先に述べた変数 \*NODE-STRUCT-MANAGER\* に格納された実体に関する記述の中から該当する識別子を持つものを探すことで特定することができる。イベントや回線を表す実体は先に述べた変数 \*ARC-STRUCT-MANAGER\* の中から両端に位置する端末機やプロセスを表す実体が持つ識別子をキーワードにして検索することで特定する。実体を特定する方法の詳細については各種変数のフォーマット等と共に『ネットワーク仕様記述表示系プログラム』を参照のこと。

## b. 図形記号を特定する方法

端末機やプロセスを表す図形記号は先に述べた変数 \*NODE-SYMBOL-MANAGER\* に格納された図形記号に関する記述の中から該当する識別子を持つものを探すことで特定することができる。イベントや回線を表す図形記号は先に述べた変数 \*ARC-SYMBOL-MANAGER\* に格納された図形記号に関する記述の中から該当する識別子を持つものを探すことで特定することができる。図形記号を特定する方法の詳細については各種変数のフォーマット等と共に『ネットワーク仕様記述表示プログラム』を参照のこと。

## c. 端末機を表す実体を構成する図形記号を差し替える方法

端末機を表す実体を構成する図形記号の差替えは次の Function で行う。

### Syntax

(Replace-draw-telephone STATE PID)

ただし、

STATE は差し替えたい端末機を表す図形記号が持つ端末機の状態で  
(Handset Datalink Tone-state) で記述される

PID は差し替えたい端末機を制御する処理系に付与された識別子

### Function

Parameter PID で指定された処理系が制御する端末機を表す実体を構成する図形記号を Parameter STATE で指定された状態を表す図形記号に差し替えて表示する。

## d. 回線を表す実体を構成する図形記号を差し替える方法

回線を表す実体を構成する図形記号の差替えは次の Function で行う。

### Syntax

(Draw-connection-internal SYMBOL PID1 PID2)

ただし、

SYMBOL は差し替えたい回線状態を表す図形記号の持つ識別子

PID1 は差し替えたい回線の端点に位置する処理系プロセスの持つ識別子

PID2 は差し替えたい回線の他方の端点に位置する処理系プロセスの持つ識別子

### Function

Parameter PID1 と PID2 で指定された処理系プロセスを両端点に持つ回線を表す実体を構成する図形記号を Parameter SYMBOL で指定された図形記号に差し替えて表示する。

#### e. イベントを表す実体を構成する図形記号を差し替える方法

イベントを表す実体を構成する図形記号の差し替えは次のFunctionで行う。

#### Syntax

(Replace-draw-arc SYMBOL FROM TO TIMING)

ただし、

SYMBOLは差し替えたいイベントを表す図形記号の持つ識別子

FROMはイベントを送りだした処理系プロセスに付与された識別子もしくは端末機を表す識別子(Device)

TOはイベントを受けた処理系プロセスに付与された識別子もしくは端末機を表す識別子(Device)

TIMINGは送りだしたイベントに対してタイマーの設定を行うかどうかを示すための値で、正の整数ならばタイマーをその値で設定し、ゼロならばタイマーの設定は行わないことを表す。

#### Function

Parameter FROMで指定された処理系プロセスもしくは端末機からParameter TOで指定された処理系プロセスもしくは端末機へParameter SYMBOLで指定された識別子を持つ図形記号で表されるイベントがParameter TIMINGで指定されたタイミングで送られたことを表示する。

その他各Functionの内部構造、各変数のフォーマット等詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

#### 4.2.3 状態遷移管理

表示系はユーザが定義したイベント/タスク変換テーブルの内容、状態遷移表の内容にしたがって端末機や回線の状態遷移を管理している。イベントを受けてから遷移後の状態を求めるまでの流れを図11に示す。

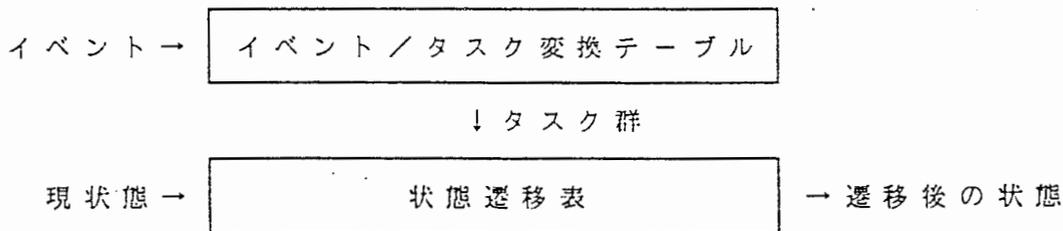


図11 イベントによる遷移後の状態を求める流れ

表示系ではこの流れを一つの関数で規定している。以下その関数について記述する。

## 状態遷移管理関数

現状態とイベントを入力とし、イベント／タスク変換テーブルを検索することで与えられたイベントをタスク群に変換し、与えられた現状態と変換されたタスク群を用いて状態遷移表の記述に従い次状態を求めるための関数。

### Syntax

(Get-Next-State HANDSET DATALINK TONE RESOURCE EVENT)

ただし、

HANDSET は遷移前の受話器の状態

( :ONHOOK | :OFFHOOK | nil )

DATALINKは遷移前のデータリンクの状態

( :IDLE | :SUSPEND | :ACTIVE | nil )

TONEは遷移前の音の状態

( :SILENT | :DIALTONE | :BUSYTONE | :RINGING |  
:RINGBACK | :INTERRUPT | :HOWLER | :WRONGNO | nil )

上記三つのパラメータがNilの場合は該当する処理系が端末機を支配下に置いていない(端末制御プロセスではない)ことを意味する

RESOURCEは遷移前の回線の状態で該当するプロセスの持つすべての回線を表す実体を構成する図形記号を次のように記述する

((SYMBOL ADDRESS).....)

ただし、

SYMBOLは該当する回線を表す実体を構成する図形記号の識別子

ADDRESS は回線の他方の終端のプロセスの識別子

EVENT はネットワーク上に流されるイベント式

### Functional Return

関数のリターン値として次のフォーマットのリストが返される。

(DEVICE-STATE EVENT RESOURCE-STATE)

ただし、

DEVICE-STATEは端末機の次状態を表すもので

(Handset Datalink Tone)のリストもしくはNil,この値がNilの時は端末機は状態遷移を行わない(遷移後の状態が遷移前と同じ)ことを意味する

EVENT は表示すべきイベント式についての記述で

ネットワーク上を流れるイベント式もしくはNil,この値がNilの時はイベントの表示を必要としないことを意味する

RESOURCE-STATEは回線の次状態を表すもので

(SYMBOL ADDRESS)のリストをメンバーに持つリスト(Nilを含む)で、

ADDRESSで表される端点との間の回線をSYMBOLで表される図形記号で表されるものに遷移させたことを意味する

### Function

処理系がイベントを受ける前の支配下にある端末機の状態、他のプロセスとの間の回線状態、受けたイベント式を入力として、ユーザの規定したイベント／タスク変換テーブル、状態遷移表の内容にしたがって遷移後の状態を求め、遷移後の状態を表示するのに必要な情報を返す。

その他各Functionの内部構造等詳細については『ネットワーク仕様記述表示系プログラム』を参照のこと。

## § 5 あとがき

ネットワーク仕様記述用の言語CSLにより記述されたネットワークの動作状況をシュミレートすることを目的としたネットワーク仕様処理システムについて記述してきたが、このシステムは動作環境や使用する図形に関連する各種設定情報を外部から与えることでローカライズできるように構成されている。また、図形操作に関するファンクションを用途別にモジュール化しているので、システムの動作表示方式の拡張についても容易に行うことができると考えている。

以後、研究の発展とともに多様化するシステムへの要求を処理していく上で有効に活用されることを切に望むものである。

最後になりましたが本システムを作成するに際して協力して下さいました株式会社東洋情報システムの皆様にこの場を借りて御礼申し上げます。

## 参考文献

- |                      |          |              |
|----------------------|----------|--------------|
| ネットワーク仕様記述言語処理系プログラム | (1990/2) | (株) 東洋情報システム |
| ネットワーク仕様記述表示系プログラム   | (1991/2) | (株) 東洋情報システム |

## 付 録 目 次

動作環境設定に用いられるDefault File

DCSL:Configuration:Csl-System

DCSL:Configuration:Chart-Layout

DCSL:Configuration:Chart-Element

DCSL:Configuration:Event-Conversion

DCSL:Configuration:State-Transition

起動直後のウィンドウと環境設定後のウィンドウ

端末機操作の例

```
;;; -*- Base: 10; Package: CSL; Mode: LISP; Syntax: Common-Lisp -*-  
(csl-system-configuration)  
(:process-id bc139 :load-host clm14 :target-spec ("clm12:>constraint>bc"))  
(:process-id bc393 :load-host clm14 :target-spec ("clm12:>constraint>cw"))  
(:process-id bc290 :load-host clm14 :target-spec ("clm12:>constraint>twc"))  
(:process-id cwm :load-host clm14 :target-spec ("clm12:>constraint>cwm"))  
(:process-id bc245 :load-host clm14 :target-spec ("clm12:>constraint>cw"))  
(:process-id twcm :load-host clm14 :target-spec ("clm12:>constraint>twcm"))
```

```
;;; -*- Mode: LISP; Base: 10; Syntax: Common-Lisp; Package: CSL -*-  
(csl-system-layout-configuration)
```

```
(:device
```

```
(:id :device1 :x 390 :y 0 :process-id bc139 :phoneno "139")  
(:id :device2 :x 285 :y -182 :process-id bc245 :phoneno "245")  
(:id :device3 :x -285 :y -182 :process-id bc290 :phoneno "290")  
(:id :device4 :x -390 :y 0 :process-id bc393 :phoneno "393"))
```

```
(:process
```

```
(:id :bcp1 :symbol :bc139 :x 210 :y 0 :process-id bc139)  
(:id :bcp2 :symbol :bc245 :x 105 :y -182 :process-id bc245)  
(:id :bcp3 :symbol :bc290 :x -105 :y -182 :process-id bc290)  
(:id :bcp4 :symbol :bc393 :x -210 :y 0 :process-id bc393)  
(:id :srp1 :symbol :cwm :x -180 :y 150 :process-id cwm)  
(:id :srp2 :symbol :unassigned-service-process :x -90 :y 180 :process-id nil)  
(:id :srp3 :symbol :trans :x 0 :y 195 :process-id trans)  
(:id :srp4 :symbol :unassigned-service-process :x 90 :y 180 :process-id nil)  
(:id :srp5 :symbol :twcm :x 180 :y 150 :process-id twcm))
```

```
;; -- Mode: LISP; Base: 10; Syntax: Common-Lisp; Package: CSL; Patch-File: Yes --
```

```
(csl-system-symbol-configuration)
```

```
(node-shape :id :onhook-idle-silent :graphics *graphics-onhook-idle-list*)
(node-shape :id :onhook-idle-ringing
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-ringing-list*))
(node-shape :id :onhook-idle-dialtone
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-dialtone-list*))
(node-shape :id :onhook-idle-ringbacktone
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :onhook-idle-busytone
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-busytone-list*))
(node-shape :id :onhook-idle-WrongNo
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-wrongno-list*))
(node-shape :id :onhook-idle-interrupt
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-interrupt-list*))
(node-shape :id :onhook-idle-howler
:graphics '(,@*graphics-onhook-idle-list* ,@*graphics-howler-list*))

(node-shape :id :onhook-active-silent :graphics *graphics-onhook-active-list*)
(node-shape :id :onhook-active-ringing
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-ringing-list*))
(node-shape :id :onhook-active-dialtone
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-dialtone-list*))
(node-shape :id :onhook-active-ringbacktone
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :onhook-active-busytone
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-busytone-list*))
(node-shape :id :onhook-active-WrongNo
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-wrongno-list*))
(node-shape :id :onhook-active-interrupt
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-interrupt-list*))
(node-shape :id :onhook-active-howler
:graphics '(,@*graphics-onhook-active-list* ,@*graphics-howler-list*))

(node-shape :id :onhook-suspend-silent :graphics *graphics-onhook-suspend-list*)
(node-shape :id :onhook-suspend-ringing
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-ringing-list*))
(node-shape :id :onhook-suspend-dialtone
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-dialtone-list*))
(node-shape :id :onhook-suspend-ringbacktone
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :onhook-suspend-busytone
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-busytone-list*))
(node-shape :id :onhook-suspend-WrongNo
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-wrongno-list*))
(node-shape :id :onhook-suspend-interrupt
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-interrupt-list*))
(node-shape :id :onhook-suspend-howler
:graphics '(,@*graphics-onhook-suspend-list* ,@*graphics-howler-list*))

(node-shape :id :offhook-idle-silent :graphics *graphics-offhook-idle-list*)
(node-shape :id :offhook-idle-ringing
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-ringing-list*))
(node-shape :id :offhook-idle-dialtone
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-dialtone-list*))
(node-shape :id :offhook-idle-ringbacktone
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :offhook-idle-busytone
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-busytone-list*))
(node-shape :id :offhook-idle-WrongNo
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-wrongno-list*))
(node-shape :id :offhook-idle-interrupt
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-interrupt-list*))
(node-shape :id :offhook-idle-howler
:graphics '(,@*graphics-offhook-idle-list* ,@*graphics-howler-list*))

(node-shape :id :offhook-active-silent
:graphics *graphics-offhook-active-list*)
(node-shape :id :offhook-active-ringing
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-ringing-list*))
(node-shape :id :offhook-active-dialtone
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-dialtone-list*))
(node-shape :id :offhook-active-ringbacktone
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :offhook-active-busytone
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-busytone-list*))
```

```

(node-shape :id :offhook-active-WrongNo
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-wrongno-list*))
(node-shape :id :offhook-active-interrupt
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-interrupt-list*))
(node-shape :id :offhook-active-howler
:graphics '(,@*graphics-offhook-active-list* ,@*graphics-howler-list*))

(node-shape :id :offhook-suspend-silent :graphics *graphics-offhook-suspend-list*)
(node-shape :id :offhook-suspend-ringing
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-ringing-list*))
(node-shape :id :offhook-suspend-dialtone
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-dialtone-list*))
(node-shape :id :offhook-suspend-ringbacktone
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-ringbacktone-list*))
(node-shape :id :offhook-suspend-busytone
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-busytone-list*))
(node-shape :id :offhook-suspend-WrongNo
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-wrongno-list*))
(node-shape :id :offhook-suspend-interrupt
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-interrupt-list*))
(node-shape :id :offhook-suspend-howler
:graphics '(,@*graphics-offhook-suspend-list* ,@*graphics-howler-list*))

(node-shape :id :illegal
:graphics '((polygon (-35 -35 35 -35 35 35 -35 35) (:filled t :gray-level 1.0)
(:thickness 1 :dash-pattern nil))))
(node-shape :id :basiccall-process
:graphics '((ellipse (0 0 20 20) (:filled t :gray-level 0.0)
(:thickness 1 :dash-pattern nil))))
(node-shape :id :active-service-process
:graphics '((polygon (-17 -17 17 -17 17 17 -17 17) (:filled t :gray-level 0.0)
(:thickness 1 :dash-pattern nil))))
(node-shape :id :idle-service-process
:graphics '((polygon (-17 -17 17 -17 17 17 -17 17) (:filled t :gray-level 0.0)
(:thickness 1 :dash-pattern '(3 2))))))

```

```
(node-symbol :id :onhook-idle-silent-device :shape :onhook-idle-silent
:handset :onhook :datalink :idle :tone :silent)
(node-symbol :id :onhook-idle-ringing-device :shape :onhook-idle-ringing
:handset :onhook :datalink :idle :tone :ringing)
(node-symbol :id :onhook-idle-dialtone-device :shape :onhook-idle-dialtone
:handset :onhook :datalink :idle :tone :dialtone)
(node-symbol :id :onhook-idle-ringbacktone-device :shape :onhook-idle-ringbacktone
:handset :onhook :datalink :idle :tone :ringback)
(node-symbol :id :onhook-idle-busytone-device :shape :onhook-idle-busytone
:handset :onhook :datalink :idle :tone :busytone)
(node-symbol :id :onhook-idle-WrongNo-device :shape :onhook-idle-WrongNo
:handset :onhook :datalink :idle :tone :wrongno)
(node-symbol :id :onhook-idle-interrupt-device :shape :onhook-idle-interrupt
:handset :onhook :datalink :idle :tone :interrupt)
(node-symbol :id :onhook-idle-howler-device :shape :onhook-idle-howler
:handset :onhook :datalink :idle :tone :howler)

(node-symbol :id :onhook-active-silent-device :shape :onhook-active-silent
:handset :onhook :datalink :active :tone :silent)
(node-symbol :id :onhook-active-ringing-device :shape :onhook-active-ringing
:handset :onhook :datalink :active :tone :ringing)
(node-symbol :id :onhook-active-dialtone-device :shape :onhook-active-dialtone
:handset :onhook :datalink :active :tone :dialtone)
(node-symbol :id :onhook-active-ringbacktone-device :shape :onhook-active-ringbacktone
:handset :onhook :datalink :active :tone :ringback)
(node-symbol :id :onhook-active-busytone-device :shape :onhook-active-busytone
:handset :onhook :datalink :active :tone :busytone)
(node-symbol :id :onhook-active-WrongNo-device :shape :onhook-active-WrongNo
:handset :onhook :datalink :active :tone :wrongno)
(node-symbol :id :onhook-active-interrupt-device :shape :onhook-active-interrupt
:handset :onhook :datalink :active :tone :interrupt)
(node-symbol :id :onhook-active-howler-device :shape :onhook-active-howler
:handset :onhook :datalink :active :tone :howler)

(node-symbol :id :onhook-suspend-silent-device :shape :onhook-suspend-silent
:handset :onhook :datalink :suspend :tone :silent)
(node-symbol :id :onhook-suspend-ringing-device :shape :onhook-suspend-ringing
:handset :onhook :datalink :suspend :tone :ringing)
(node-symbol :id :onhook-suspend-dialtone-device :shape :onhook-suspend-dialtone
:handset :onhook :datalink :suspend :tone :dialtone)
(node-symbol :id :onhook-suspend-ringbacktone-device :shape :onhook-suspend-ringbacktone
:handset :onhook :datalink :suspend :tone :ringback)
(node-symbol :id :onhook-suspend-busytone-device :shape :onhook-suspend-busytone
:handset :onhook :datalink :suspend :tone :busytone)
(node-symbol :id :onhook-suspend-WrongNo-device :shape :onhook-suspend-WrongNo
:handset :onhook :datalink :suspend :tone :wrongno)
(node-symbol :id :onhook-suspend-interrupt-device :shape :onhook-suspend-interrupt
:handset :onhook :datalink :suspend :tone :interrupt)
(node-symbol :id :onhook-suspend-howler-device :shape :onhook-suspend-howler
:handset :onhook :datalink :suspend :tone :howler)

(node-symbol :id :offhook-idle-silent-device :shape :offhook-idle-silent
:handset :offhook :datalink :idle :tone :silent)
(node-symbol :id :offhook-idle-ringing-device :shape :offhook-idle-ringing
:handset :offhook :datalink :idle :tone :ringing)
(node-symbol :id :offhook-idle-dialtone-device :shape :offhook-idle-dialtone
:handset :offhook :datalink :idle :tone :dialtone)
(node-symbol :id :offhook-idle-ringbacktone-device :shape :offhook-idle-ringbacktone
:handset :offhook :datalink :idle :tone :ringback)
(node-symbol :id :offhook-idle-busytone-device :shape :offhook-idle-busytone
:handset :offhook :datalink :idle :tone :busytone)
(node-symbol :id :offhook-idle-WrongNo-device :shape :offhook-idle-WrongNo
:handset :offhook :datalink :idle :tone :wrongno)
(node-symbol :id :offhook-idle-interrupt-device :shape :offhook-idle-interrupt
:handset :offhook :datalink :idle :tone :interrupt)
(node-symbol :id :offhook-idle-howler-device :shape :offhook-idle-howler
:handset :offhook :datalink :idle :tone :howler)

(node-symbol :id :offhook-active-silent-device :shape :offhook-active-silent
:handset :offhook :datalink :active :tone :silent)
(node-symbol :id :offhook-active-ringing-device :shape :offhook-active-ringing
:handset :offhook :datalink :active :tone :ringing)
(node-symbol :id :offhook-active-dialtone-device :shape :offhook-active-dialtone
:handset :offhook :datalink :active :tone :dialtone)
(node-symbol :id :offhook-active-ringbacktone-device :shape :offhook-active-ringbacktone
:handset :offhook :datalink :active :tone :ringback)
(node-symbol :id :offhook-active-busytone-device :shape :offhook-active-busytone
```

```
:handset :offhook :datalink :active :tone :busytone)
(node-symbol :id :offhook-active-WrongNo-device :shape :offhook-active-WrongNo
:handset :offhook :datalink :active :tone :wrongno)
(node-symbol :id :offhook-active-interrupt-device :shape :offhook-active-interrupt
:handset :offhook :datalink :active :tone :interrupt)
(node-symbol :id :offhook-active-howler-device :shape :offhook-active-howler
:handset :offhook :datalink :active :tone :howler)

(node-symbol :id :offhook-suspend-silent-device :shape :offhook-suspend-silent
:handset :offhook :datalink :suspend :tone :silent)
(node-symbol :id :offhook-suspend-ringing-device :shape :offhook-suspend-ringing
:handset :offhook :datalink :suspend :tone :ringing)
(node-symbol :id :offhook-suspend-dialtone-device :shape :offhook-suspend-dialtone
:handset :offhook :datalink :suspend :tone :dialtone)
(node-symbol :id :offhook-suspend-ringbacktone-device :shape :offhook-suspend-ringbacktone
:handset :offhook :datalink :suspend :tone :ringback)
(node-symbol :id :offhook-suspend-busytone-device :shape :offhook-suspend-busytone
:handset :offhook :datalink :suspend :tone :busytone)
(node-symbol :id :offhook-suspend-WrongNo-device :shape :offhook-suspend-WrongNo
:handset :offhook :datalink :suspend :tone :wrongno)
(node-symbol :id :offhook-suspend-interrupt-device :shape :offhook-suspend-interrupt
:handset :offhook :datalink :suspend :tone :interrupt)
(node-symbol :id :offhook-suspend-howler-device :shape :offhook-suspend-howler
:handset :offhook :datalink :suspend :tone :howler)

(node-symbol :id :illegal-device :shape :illegal)
(node-symbol :id :bc245 :shape :basiccall-process :label-string "245")
(node-symbol :id :bc393 :shape :basiccall-process :label-string "393")
(node-symbol :id :bc290 :shape :basiccall-process :label-string "290")
(node-symbol :id :bc139 :shape :basiccall-process :label-string "139")
(node-symbol :id :cwm :shape :active-service-process :label-string "CWM")
(node-symbol :id :trans :shape :active-service-process :label-string "Trans")
(node-symbol :id :twcm :shape :active-service-process :label-string "TWCM")
(node-symbol :id :unassigned-service-process :shape :idle-service-process)
```

```

(defparameter *graphics-onhook-idle-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17 27 17
             27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled t :gray-level 0.6) (:thickness 0 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (polygon (-13 -23 -27 -18 -32 -13 -32 -8 -17 -8 -17 -13 -12 -18 12 -18 17 -13
             17 -8 32 -8 32 -13 27 -18 13 -23)
              (:filled t :gray-level 0.6) (:thickness 0 :dash-pattern nil))))

(defparameter *graphics-onhook-active-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17
             27 17 27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled nil) (:thickness 1 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled nil)
              (:thickness 1 :dash-pattern nil))
    (polygon (-13 -23 -27 -18 -32 -13 -32 -8 -17 -8 -17 -13 -12 -18 12 -18
             17 -13 17 -8 32 -8 32 -13 27 -18 13 -23)
              (:filled nil) (:thickness 1 :dash-pattern nil))))

(defparameter *graphics-onhook-suspend-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17
             27 17 27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled t :gray-level 1.0) (:thickness 0 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (polygon (-13 -23 -27 -18 -32 -13 -32 -8 -17 -8 -17 -13 -12 -18 12 -18
             17 -13 17 -8 32 -8 32 -13 27 -18 13 -23)
              (:filled t :gray-level 1.0) (:thickness 0 :dash-pattern nil))))

(defparameter *graphics-offhook-idle-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17 27 17
             27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled t :gray-level 0.6) (:thickness 0 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (polygon (-45 -31 -51 -18 -51 -10 -48 -7 -37 -18 -40 -21 -40 -29 -24 -45 -16
             -45 -13 -42 -2 -53 -5 -56 -13 -56 -26 -50)
              (:filled t :gray-level 0.6) (:thickness 0 :dash-pattern nil))))

(defparameter *graphics-offhook-active-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17 27 17
             27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled nil) (:thickness 1 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled nil) (:thickness 1 :dash-pattern nil))
    (polygon (-45 -31 -51 -18 -51 -10 -48 -7 -37 -18 -40 -21 -40 -29 -24 -45
             -16 -45 -13 -42 -2 -53 -5 -56 -13 -56 -26 -50)
              (:filled nil) (:thickness 1 :dash-pattern nil))))

(defparameter *graphics-offhook-suspend-list*
  '((polygon (-8 -17 -12 -17 -12 -12 -27 7 -27 17 -20 17 -15 12 15 12 20 17 27 17
             27 7 12 -12 12 -17 8 -17 8 -12 -8 -12)
    (:filled t :gray-level 1.0) (:thickness 0 :dash-pattern nil))
    (ellipse (0 -4 12 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (polygon (-45 -31 -51 -18 -51 -10 -48 -7 -37 -18 -40 -21 -40 -29 -24 -45
             -16 -45 -13 -42 -2 -53 -5 -56 -13 -56 -26 -50)
              (:filled t :gray-level 1.0) (:thickness 0 :dash-pattern nil))))

(defparameter *graphics-dialtone-list*
  '((ellipse (0 17 25 25) (:filled t :gray-level 0.8)
              (:thickness 0 :dash-pattern nil))
    (ellipse (0 17 15 15) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (17 6 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (11 0 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (3 -3 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (-6 -2 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (-14 3 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (-19 11 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))
    (ellipse (-20 20 4 4) (:filled t :gray-level 0.0)
              (:thickness 0 :dash-pattern nil))))

```

```

                (:thickness 0 :dash-pattern nil))
(ellipse (-17 28 4 4) (:filled t :gray-level 0.0)
          (:thickness 0 :dash-pattern nil))
(ellipse (-11 34 4 4) (:filled t :gray-level 0.0)
          (:thickness 0 :dash-pattern nil))
(ellipse (-3 37 4 4) (:filled t :gray-level 0.0)
          (:thickness 0 :dash-pattern nil)))

(defparameter *graphics-ringing-list*
  '((polygon (6 5 7 5 19 -29 26 -19 37 -56 36 -56 25 -24 17 -35)
            (:filled t :gray-level 0.8) (:thickness 0 :dash-pattern nil))
    (polygon (-6 5 -7 5 -19 -29 -26 -19 -37 -56 -36 -56 -25 -24 -17 -35)
            (:filled t :gray-level 0.8) (:thickness 0 :dash-pattern nil))))

(defparameter *graphics-busytone-list*
  '((lines (-5 -47 65 -47)
          (:filled t :gray-level 1.0) (:thickness 3 :dash-pattern '(5 5)))
    (string ("BusyTone" -2 52) (:character-style '(:fix :roman :normal)
      :attachment-x :left :attachment-y :baseline))))

(defparameter *graphics-ringbacktone-list*
  '((lines (-5 -47 65 -47)
          (:filled t :gray-level 1.0) (:thickness 3 :dash-pattern '(9 2 2 2)))
    (string ("RingBack" -2 52) (:character-style '(:fix :roman :normal)
      :attachment-x :left :attachment-y :baseline))))

(defparameter *graphics-interrupt-list*
  '((lines (-5 -47 65 -47)
          (:filled t :gray-level 1.0) (:thickness 3 :dash-pattern '(2 2)))
    (string ("Interrupt" -2 52) (:character-style '(:fix :roman :normal)
      :attachment-x :left :attachment-y :baseline))))

(defparameter *graphics-howler-list*
  '((lines (-54 2 -64 -13 -54 -13 -64 -23 -54 -23 -64 -33 -43 -33 -64 -43)
          (:filled t :gray-level 1.0) (:thickness 2 :dash-pattern nil))
    (lines (-50 2 -35 -13 -45 -13 -25 -23 -35 -23 -15 -33 -25 -33 -2 -48)
          (:filled t :gray-level 1.0) (:thickness 2 :dash-pattern nil))))

(defparameter *graphics-wrongno-list*
  '((lines (-5 -47 65 -47)
          (:filled t :gray-level 1.0) (:thickness 3 :dash-pattern nil))
    (string ("WrongNo" -2 52) (:character-style '(:fix :roman :normal)
      :attachment-x :left :attachment-y :baseline))))

(defparameter *transmitted-event-list*
  '(:line-shape '(line) :arrow '(single 10 8) :line-dash-pattern nil
    :line-thickness 1 :line-gray-level 1.0))

(defparameter *timeout-event-list*
  '(:line-shape '(line) :arrow nil :line-dash-pattern '(5 2 2 2)
    :line-thickness 2 :line-gray-level 1.0))

(defparameter *waiting-event-list*
  '(:line-shape '(line) :arrow '(single 10 8) :line-dash-pattern '(10 10)
    :line-thickness 1 :line-gray-level 1.0))

(defparameter *active-connection-list*
  '(:line-shape '(line) :arrow nil :line-dash-pattern nil
    :line-thickness 10 :line-gray-level 0.5))

(defparameter *suspended-connection-list*
  '(:line-shape '(line) :arrow nil :line-dash-pattern '(10 10)
    :line-thickness 10 :line-gray-level 0.5))

(defparameter *reserved-resource-list*
  '(:line-shape '(line) :arrow nil :line-dash-pattern nil
    :line-thickness 5 :line-gray-level 0.2))

(defparameter *hardwired-arc-list*
  '(:line-shape '(line) :arrow nil :line-dash-pattern nil
    :line-thickness 1 :line-gray-level 1.0))

(defparameter *unused-arc-list*
  '(:line-shape '(none) :arrow nil :line-dash-pattern nil
    :line-thickness 0 :line-gray-level 0.0))

```

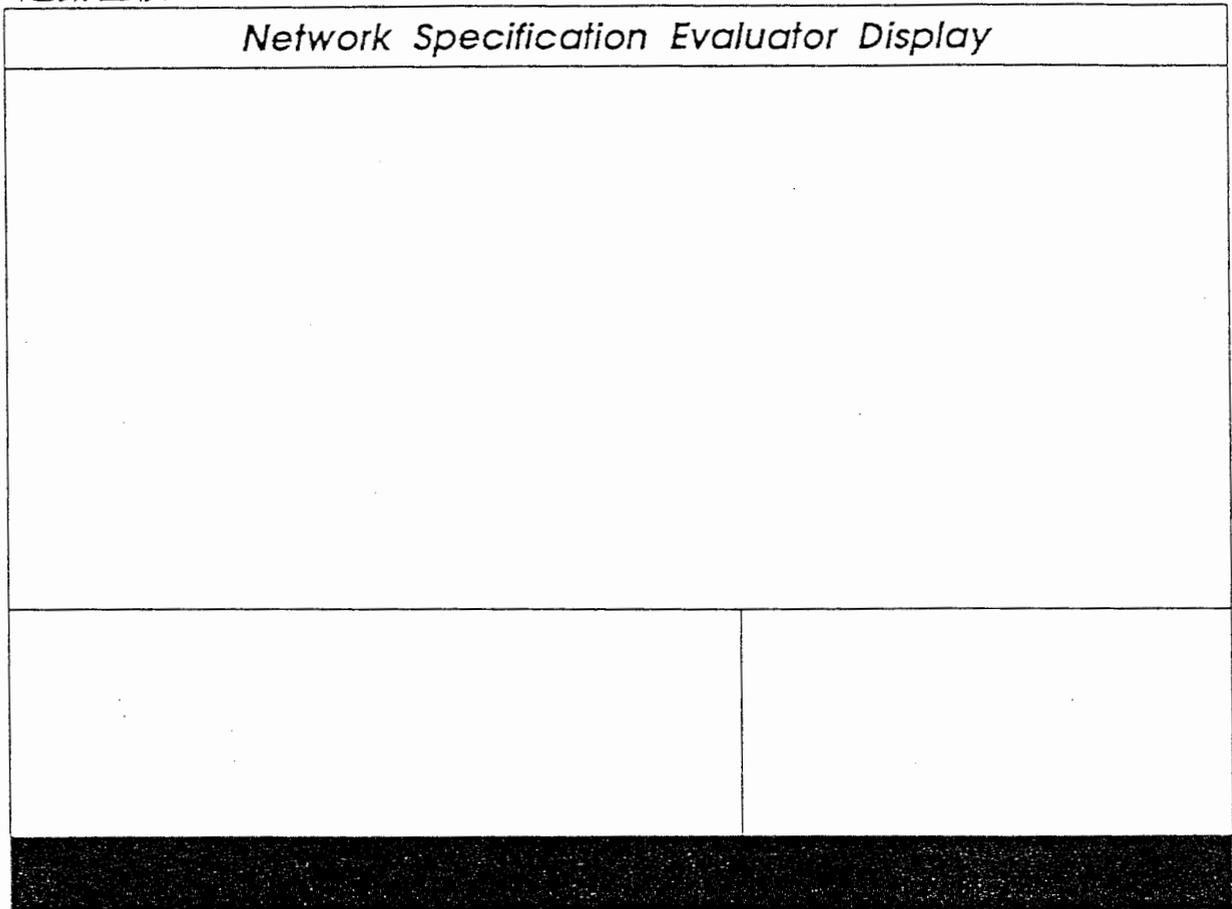
```
;;; -*- Mode: LISP; Base: 10; Syntax: Common-Lisp; Package: CSL -*-
```

```
(silent nil ((start :silent)))  
(dialtone nil ((start :dialtone)))  
(ringback nil ((start :ringback)))  
(ringing nil ((start :ringing)))  
(alert (addr) ((start :ringing) (hunt addr)))  
(busytone nil ((start :busytone)))  
(cwtone nil ((start :interrupt)))  
(wrongno nil ((start :wrongno)))  
(howling nil ((start :howler)))  
(digits (num) ((setup num)))  
(setup (num) ((digits num)))  
(onhook nil ((onhook)))  
(offhook nil ((offhook)))  
(flashhook nil ((flashhook)))  
(connect (addr) ((connect addr)))  
(disconnect (addr) ((disconnect addr)))  
(hold (addr) ((hold addr)))  
(suspend (addr) ((suspend addr)))  
(hunt (addr) ((hunt addr)))  
(release (addr) ((release addr)))  
(silentRel (addr) ((start :silent) (release addr)))  
(discRel (addr) ((disconnect addr) (release addr)))
```

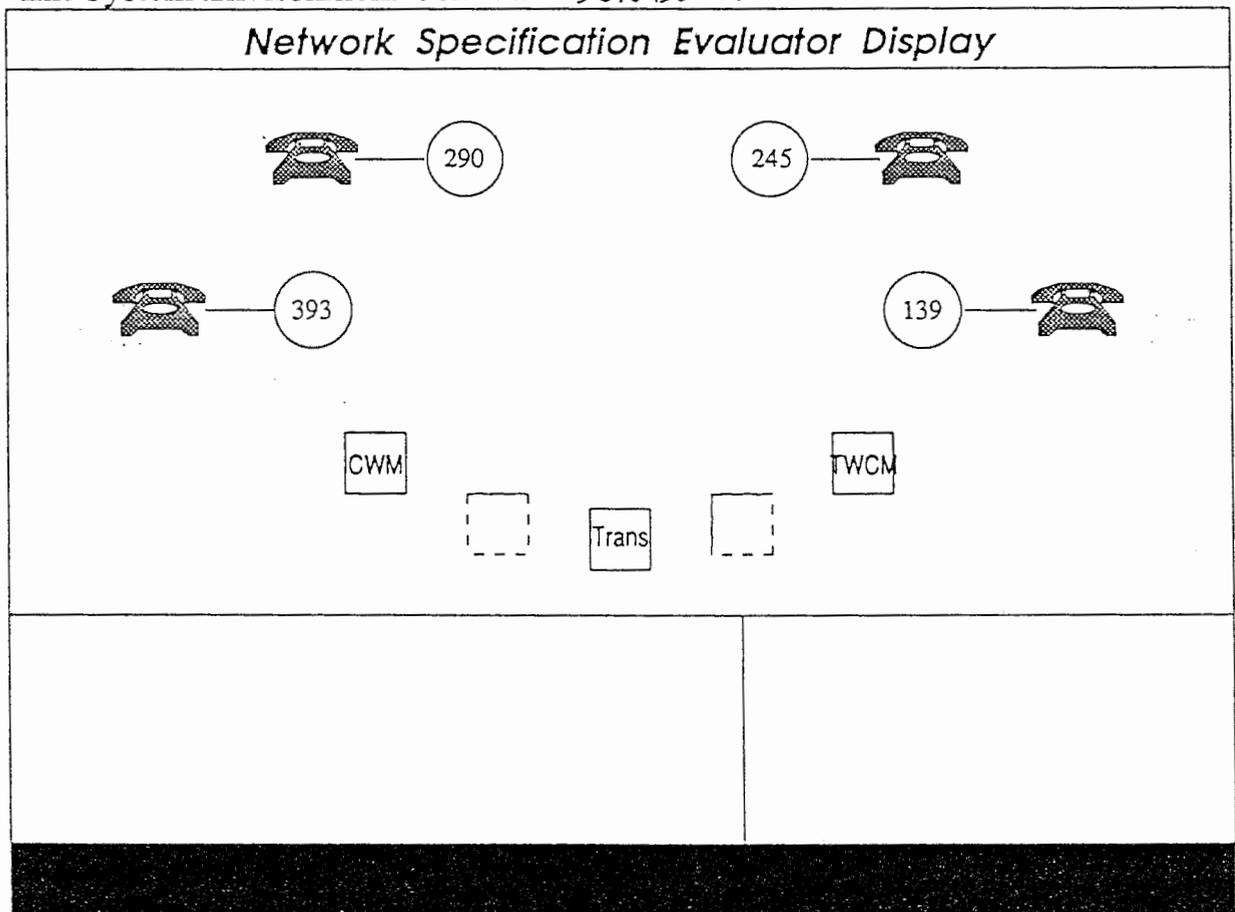
```
;;; -*- Mode: LISP; Syntax: Common-Lisp; Base: 10; Package: CSL -*-
```

```
(:free hunt :idle)  
(:free connect :active)  
(:idle connect :active)  
(:idle release :free)  
(:suspend connect :active)  
(:suspend disconnect :idle)  
(:suspend release :free)  
(:active disconnect :idle)  
(:active hold :suspend)  
(:active suspend :suspend)  
(:active release :idle)  
(:onhook offhook :offhook)  
(:offhook onhook :onhook)
```

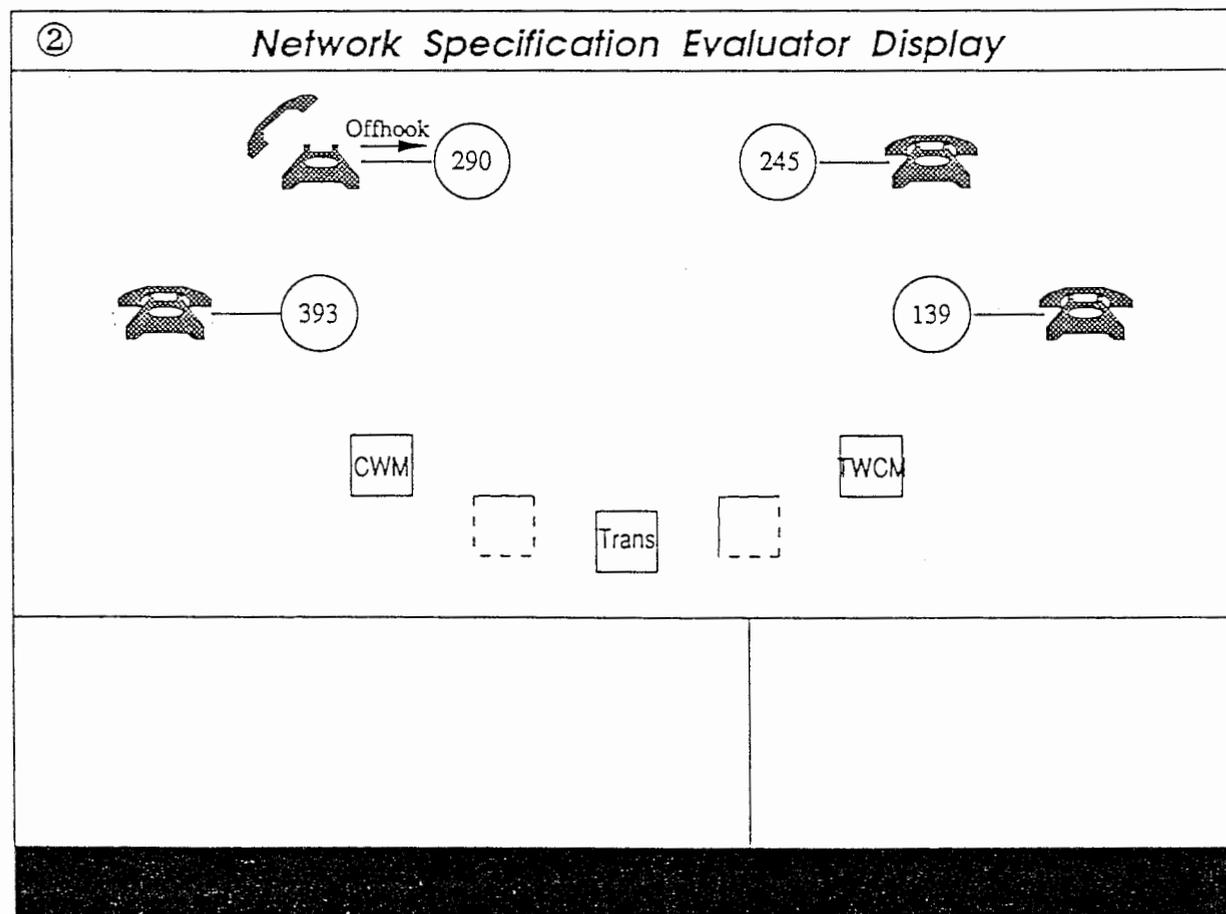
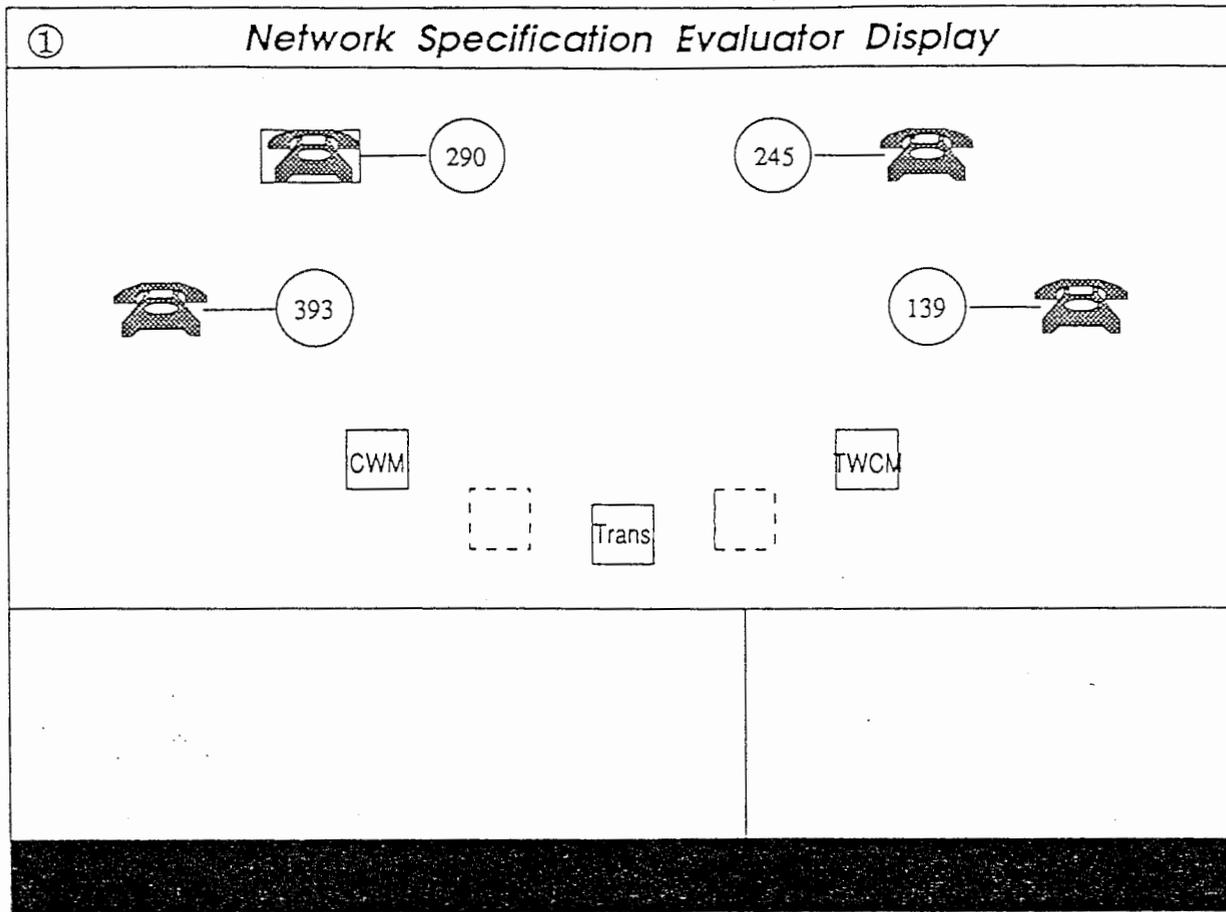
起動直後のウィンドウ



Init System Environment Command 実行後のウィンドウ

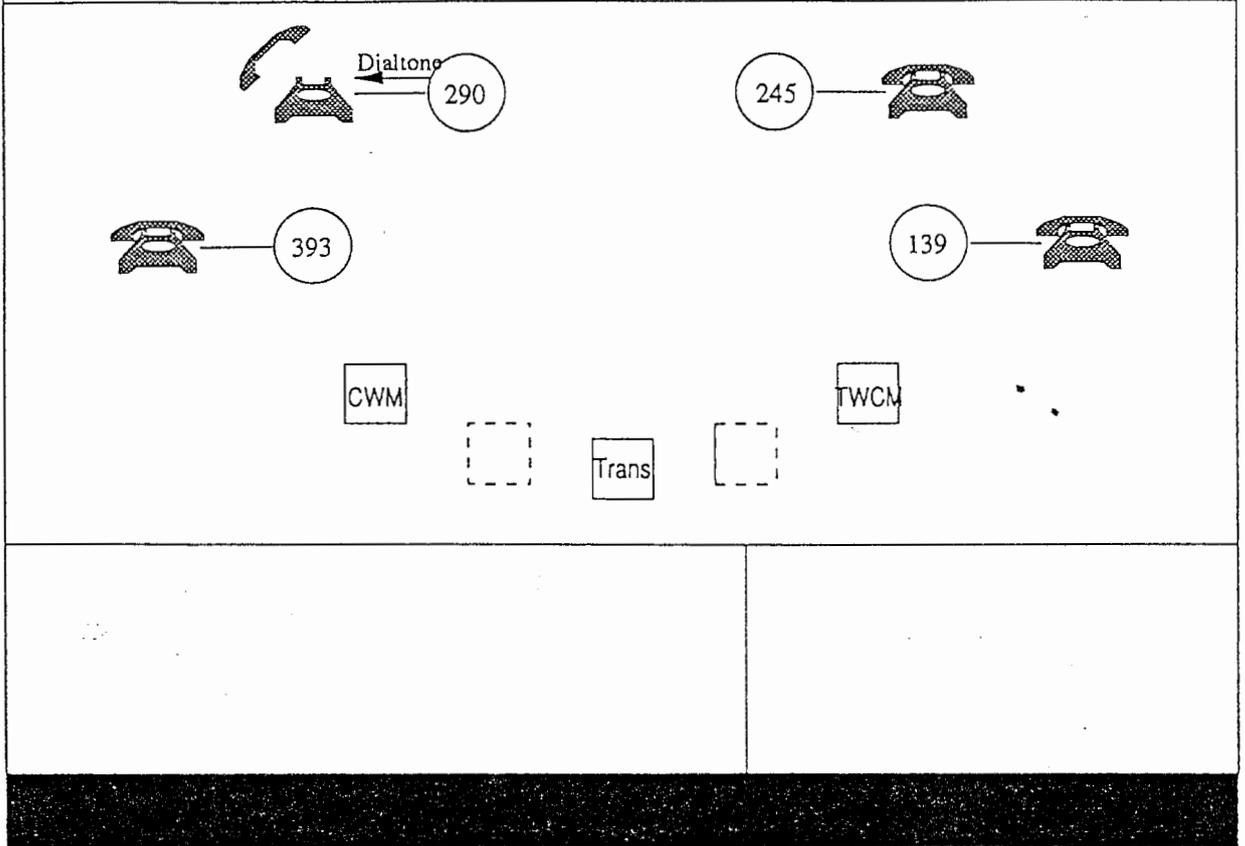


マウスの左ボタンによるクリック(オフフック)の例



③

### Network Specification Evaluator Display



④

### Network Specification Evaluator Display

