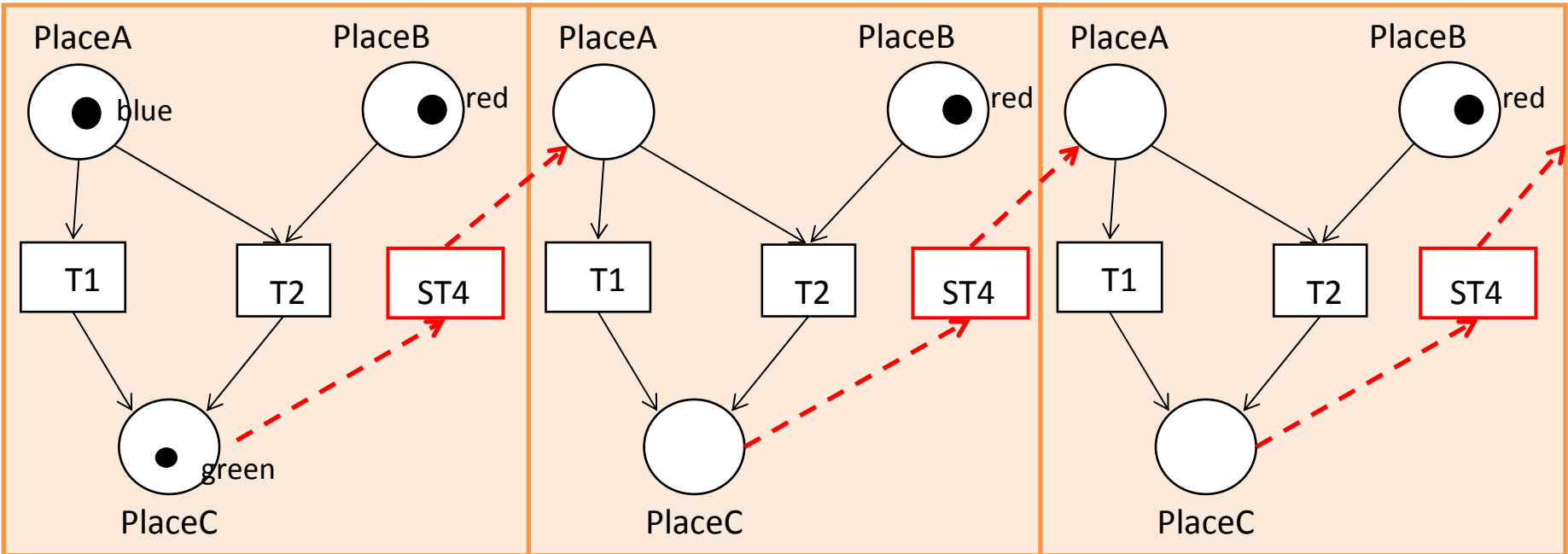


# セルペトリネット開発ツールの構築

情報工学専攻 08TA564G

宮野 八州子

# セルペトリネットとは



セル

カラーペトリネットの定義

セル

+ 近傍の定義

セル

= セル



セルペトリネット

# セルペトリネット実行例

The image displays four overlapping browser windows showing the execution progress of a 2D Cell Petri net simulation. Each window displays configuration parameters, a 5x5 grid visualization, and control buttons. The progression shows the simulation moving from step 1 to step 6, with a final completion message.

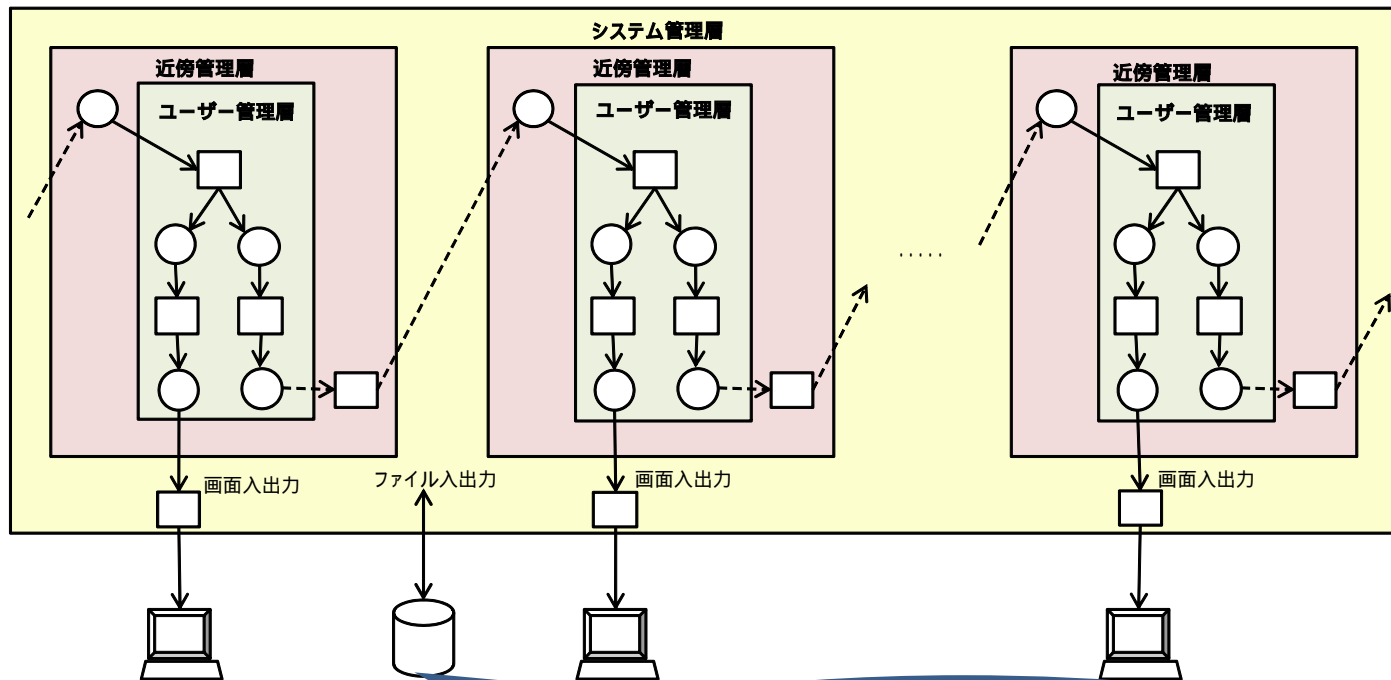
**Window 1 (Leftmost):** Configuration screen for "Cell Petri net 構築 (2次元)". Parameters include: 実行日: 2010/01/24, ペトリネット定義名: User\_FRF\_基本, 初期状態定義名: なし, 実行するセルの個数: (5 \* 5), 開始セル\_1: (0(+), 0(+)) 色=(Red) パラメータ=(.), 開始セル\_2: (., .) 色=(.) パラメータ=(.), 開始セル\_3: (., .) 色=(.) パラメータ=(.), 開始セル\_4: (., .) 色=(.) パラメータ=(.), 最大実行回数: 10, 引数の画面表示: はい. Buttons: プレビュー, STEP, データを初期値にリセット, データクリア.

**Window 2:** Shows the simulation at Step 1. The 5x5 grid has a single red cell at (0,0). Text: \*\*\*現在 1 番目のSTEPを実行中\*\*\*. Buttons: STEP, 戻る, 実行完了.

**Window 3:** Shows the simulation at Step 2. The 5x5 grid has two red cells at (0,0) and (0,1). Text: \*\*\*現在 2 番目のSTEPを実行中\*\*\*. Buttons: STEP, 戻る, 実行完了.

**Window 4 (Rightmost):** Shows the simulation at Step 6. The 5x5 grid has six red cells forming a diagonal from (0,0) to (5,5). Text: \*\*\*現在 6 番目のSTEPを実行中\*\*\*. Below the grid: \*\*\*\*\*今回の実行は 6 回で終了しました。\*\*\*\*\*. Buttons: STEP, 戻る, 実行完了.

# セルペトリネット開発ツールの概念図



FRF : Firing Rule File (セルペトリネット定義ファイル)  
ACF: Active Cell File (実行セルファイル)  
MSF: Mark Status File(マーク状態ファイル)  
INI: Initial Status File(初期状態定義ファイル) ほか

- システム管理層 : 画面入出力、ファイル管理、パラメータ管理 (PHPプログラム)
- 近傍管理層 : 次に実行されるセルの決定 (システムFRF、PHPプログラム)
- ユーザー管理層 : カラーペトリネットの定義 (ユーザーFRF)、近傍の定義 (ユーザーFRF)、初期状態の定義 (INI、開始画面)

# セルペトリネットの定義 (FRF: Firing Rule File)

- 発火条件およびtransitionが発火した時の出力情報を定義するcsvファイル
- ユーザーFRFでは、カラーペトリネットと近傍を定義する
- システムFRFでは、近傍処理実行のtransitionと、画面出力のtransitionを定義する

```
*      rule      transition  input/  output_  place  mark
*      id        id          output  cell_id  id
*  同じセル内の別のPlaceにmarkを書く
      Rule1      T1          i              PlaceA  red
      Rule1      T1          o      =      PlaceC  red
*  入力条件が複数あるパターン
      Rule2      T2          i              PlaceA  blue
      Rule2      T2          i              PlaceB  red
      Rule2      T2          o      =      PlaceC  green
```

システムFRF (抜粋)

ユーザーFRF (例)

```
*      rule id    transition  input/  output_  place  mark
*      id        id          output  cell id  id
*
      SR30      ST3          i              to_next  %=split[%o_towards:%o_color:%parm]
      SR30      ST3          o      %o_towards  receive  %=concat[%o_towards:%o_color:%parm]

*  "SC_DISPLAY"セルに現在のセルIDと色を書く(システムFRFでの定義例)
      SR41      ST4          i              to_display  %color
      SR41      ST4          o      SC_DISPLAY  %cell_id  %color
```

# 近傍の定義 (出力セルの決定)

その後も継続する動きを伝えるタイプ: (ユーザーFRFの place to\_next の mark で指定)

◆ 規則的に動きを指定できるもの (X軸方向に+1、Y軸方向に+2など) . . . + , - , =

◆ 数式を用いて表すことができるもの . . . . . C 1 , C 2

◆ セルの範囲の中でランダムに動くもの . . . . . R

1回だけの単発のタイプ: (ユーザーFRFの output\_cell\_id で指定)

◆ ある特定のセル (例: あるセルに定数の値を書き込む) . . . . . セルID

◆ すべてのセル (例: すべてのセルに終了通知placeを書き込む) . . . . . all

## ユーザーFRFでの定義例

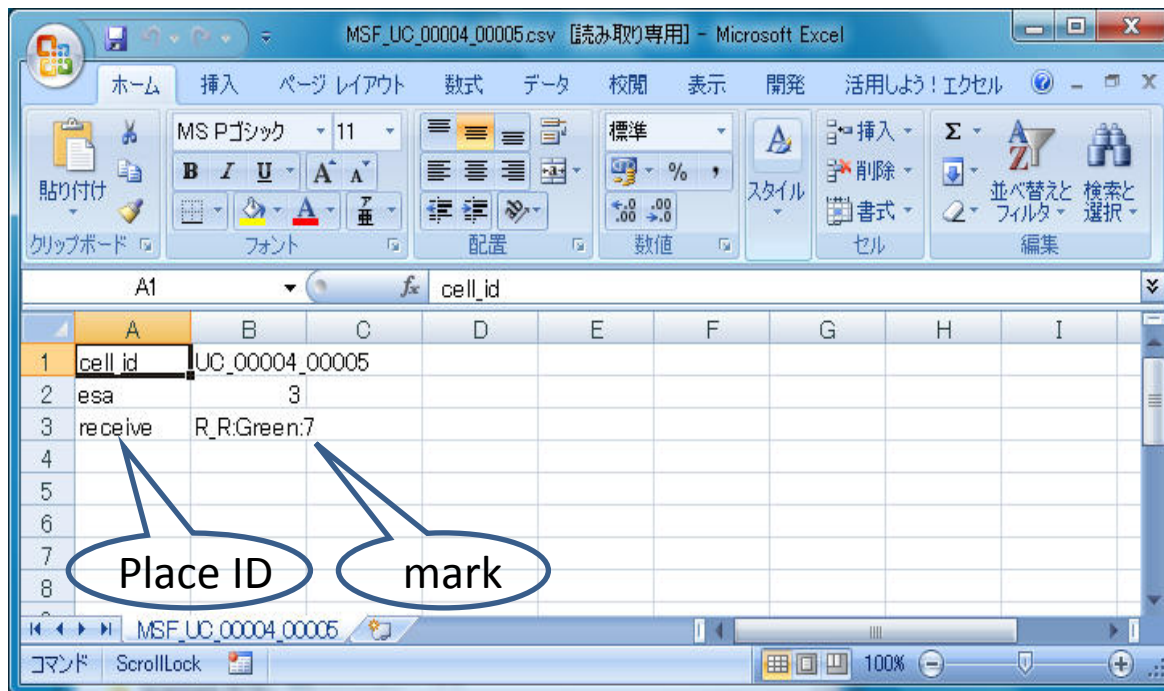
* rule_id	transi- tion id	input/ output	output_ cell_id	place_id	mark
UR01	T1	o	=	to_next	+1_+2:red:0
UR02	T2	o	=	to_next	-10_+3:red:0
UR03	T3	o	=	to_next	C1_(2&3):red:0
UR04	T4	o	=	to_next	C2_(2&3&5):red:0
UR05	T5	o	=	to_next	R_R:red:0
UR06	T6	o	UC_00002_00008	bairitsu	2
UR07	T7	o	all	Stop	yes

動きを指定

C1\_(2&3) : Xは+1, Y=2X+3  
 C2\_(2&3&5): Xは+1, Y=2X<sup>2</sup>+3X+5 で計算

# 各セルのマーク状況の更新・保持 (MSFの役割)

- MSF (Mark Status File) : セルの現在のマーク状況を保持するcsvファイル
- セルごとに作られる
- ペトリネット実行前に読み込まれ、実行後ファイルに新しい状況に書きなおされる
- このファイルはシステムが自動的に作成・更新していく

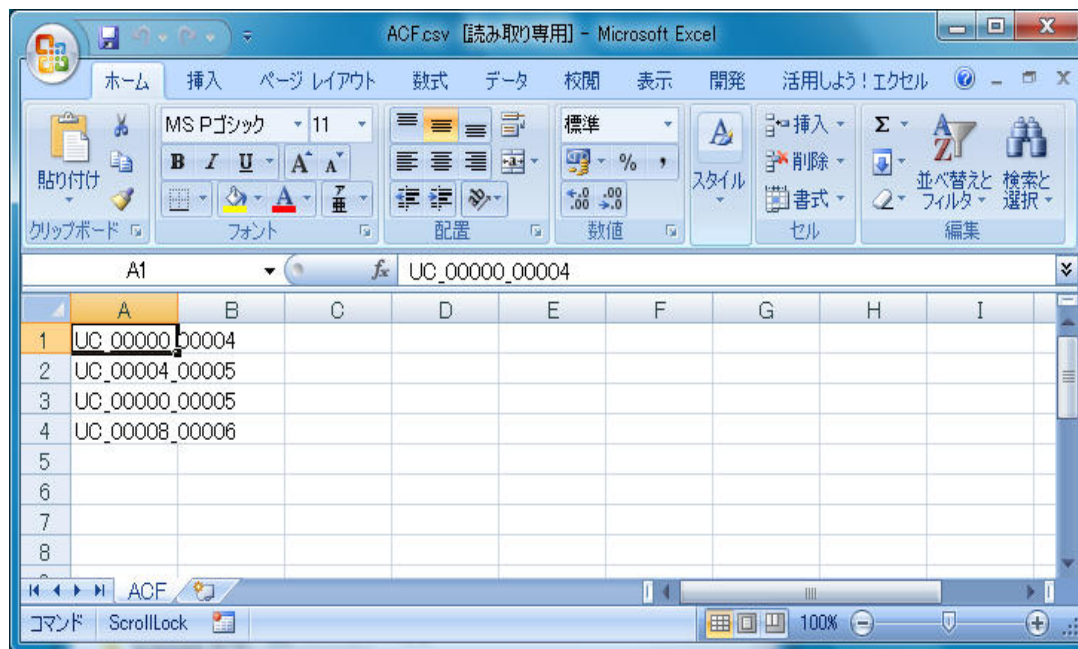


## セルIDの規則

- ◆システムセル : 画面出力用セル  
SC\_\_DISPLAY
- ◆ユーザーセル :  
UC\_\_nnnnn\_mmmmm  
nnnnnはX軸方向の番号(5桁)  
mmmmmはY軸方向の番号

# 実行セルの制御 ( A C F の役割)

- A C F (Active Cell File) : 現在実行可能なセルの番号を定義するcsvファイル
- 最初は開始画面の開始セルで指定したセルの番号が蓄えられる
- ペトリネット実行前に読み込まれ、読み込まれたすべてのセルでペトリネットを実行し、すべてのセルの実行が終わったら、次に実行するセルの番号に書き換えられる
- このファイルはシステムが自動的に作成・更新していく





# セルペトリネットの実行手順

1. ユーザーFRFを作成する
2. 初期状態定義ファイルを作成する
3. セルペトリネット開始画面を呼び出す

The image shows the 'Cell Petri net 構築 Setup Screen (v5)' in a Windows Internet Explorer browser window. The browser address is `http://localhost/2jigen_CPTN_v5/CELL_PTN_setup_5.php?f`. The page title is 'CELL\_PTN\_setup\_5.php - Windows Internet Explorer'. The browser menu includes 'ファイル(E)', '編集(E)', '表示(V)', 'お気に入り(A)', 'ツール(T)', and 'ヘルプ(H)'. The address bar shows 'Google' and search options. The main content area is titled 'Cell Petri net 構築 Setup Screen (v5)'. It contains the following configuration fields:

- \*\*\* 二次元 Cell Petri Nets \*\*\*
- ペトリネット定義名 : User\_FRF\_torisan (必須)
- 初期状態定義名 : INI\_torisan (任意)
- 実行するセルの個数 : ( 10 \* 10 )
- 開始セル定義 : 位置 (x,y) 動く方向(x,y) 色 初期パラメータ
- ( 0, 0 ) ( =,+n,-n,R : n=1~最大値 )
- 開始セル\_1 : ( 5 , 5 ) ( R , R ) 青
- 開始セル\_2 : ( 2 , 8 ) ( R , R ) 緑
- 開始セル\_3 : ( , ) ( , ) 赤
- 開始セル\_4 : ( , ) ( , ) 赤
- 最大実行回数 : 30
- 引数の画面表示 : はい

At the bottom, there are buttons for 'プレビュー', 'STEP', and 'データクリア', along with the text '\*\*\* データを初期値に戻す \*\*\*'.

Two CSV files are shown on the left side of the image, with red circles highlighting their names and red arrows pointing to the corresponding fields in the setup screen:

- User\_FRF\_torisan.csv** (circled in red): This file defines the transition rules. It contains columns for `rule_id`, `transiti`, `id`, and `動作`. The first section lists transitions with `UR02` and `UT02`. The second section lists transitions with `UR31` and `UT3`. The third section lists transitions with `UR41` and `UT4`.
- INI\_torisan.csv** (circled in red): This file defines the initial state. It contains columns for `cell_id_x`, `cel`, and `初期値`. The first section lists cells with `セルC_00000_0000` and `初期値` 0. The second section lists cells with `初期値` 1.



# まとめ

本研究では、Web上で動くセルペトリネットを実現するツールを開発した。

- ◆開始画面で実行するセルの個数、ペトリネット定義ファイルの名前等を指定する
- ◆ACFにより、実行するセルを管理し、
- ◆MSFにより、実行するセルのマーク状態を知り、
- ◆FRFにより、カラーペトリネットを実行し、近傍定義により次の実行セルを決定する
- ◆いずれかのセルで画面表示が指示されたら、画面に全セルの色を表示する

利点:

- ユーザーはプログラムを書くことなく、Excelファイルを作成することでカラーペトリネットおよび近傍を定義できる
- 開始画面で実行するセルの個数や開始セルを変更するだけで、異なる大きさのセルペトリネットを簡単に動かすことができる



# 動的変数 (変数・関数の提供)

➤ F R F で定義する Place や mark の値を実行時に動的に変更する

➤ F R F を記述するときの規則

◆ “%” で始めた名前は変数

%cell\_id はその時のセルIDを表す

それ以外の名前はユーザーが任意に指定可能 (同一発火条件の間有効)

◆ “%=” で始めたら関数

現在 9 個の関数を提供

関数名	形式	用法
concat	%=concat[%a:%b: ... :%x]	“:” をセパレーターとして元の mark 値を分割し、それぞれ変数名を %a,%b...%x とする
concat_any	%=concat_any[(セパレーター):%a:%b: ... %x]	ユーザーが指定したセパレーターで mark 値の分割を行う
split	%=split[%a:%b: ... :%x]	“:” をセパレーターとして %a,%b...%x を結合する
split_any	%=split_any[(セパレーター):%a:%b: ... %x]	ユーザーが指定したセパレーターで %a,%b...%x を結合する
加算	%=+[%a:%b]	%a + %b を計算する
減算	%=-[%a:%b]	%a - %b を計算する
掛け算	%=*[%a:%b]	%a * %b を計算する
割り算	%=/[%a:%b]	%a / %b の商を計算する (小数点以下第 2 位で四捨五入)
Mod	%=%[%a:%b]	%a / %b の余りを計算する

# 変数・関数の使用例

## ユーザーFRFの定義

* rule_id	transition	input/	output_	place_id	mark
*	id	output	cell_id		
* 渡された引数を動く方向(o_towards)と色(o_color)と食べた餌の合計(gokei)に分離する					
UR02	UT02	i		receive	%=split[%o_towards:%o_color:%gokei]
UR02	UT02	o	=	to_display	%o_color
UR02	UT02	o	=	o_towards	%o_towards
UR02	UT02	o	=	o_color	%o_color
UR02	UT02	o	=	gokei	%gokei
* ここで食べた餌の数を合計に加える					
UR31	UT03	i		esa	%esa
UR31	UT03	i		gokei	%gokei
UR31	UT03	o	=	gokei	%=+[%gokei:%esa]
* 次のセルに移動					
UR07	UT7	i		o_towards	%o_towards
UR07	UT7	i		o_color	%o_color
UR07	UT7	i		gokei	%gokei
UR07	UT7	o	=	to_next	%=concat[%o_towards:%o_color:%gokei]

## MST(Mark Status Table)の変化

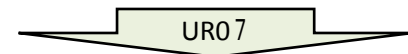
Cell Id	Place ID	Mark
UC_00004_00005	cell id	UC_00004_00005
UC_00004_00005	receive	+1_+1:red:5
UC_00004_00005	esa	3



UC_00004_00005	cell id	UC_00004_00005
UC_00004_00005	to_display	red
UC_00004_00005	o_towards	+1_+1
UC_00004_00005	o_color	red
UC_00004_00005	gokei	5
UC_00004_00005	esa	3



UC_00004_00005	cell id	UC_00004_00005
UC_00004_00005	to_display	red
UC_00004_00005	o_towards	+1_+1
UC_00004_00005	o_color	red
UC_00004_00005	gokei	8

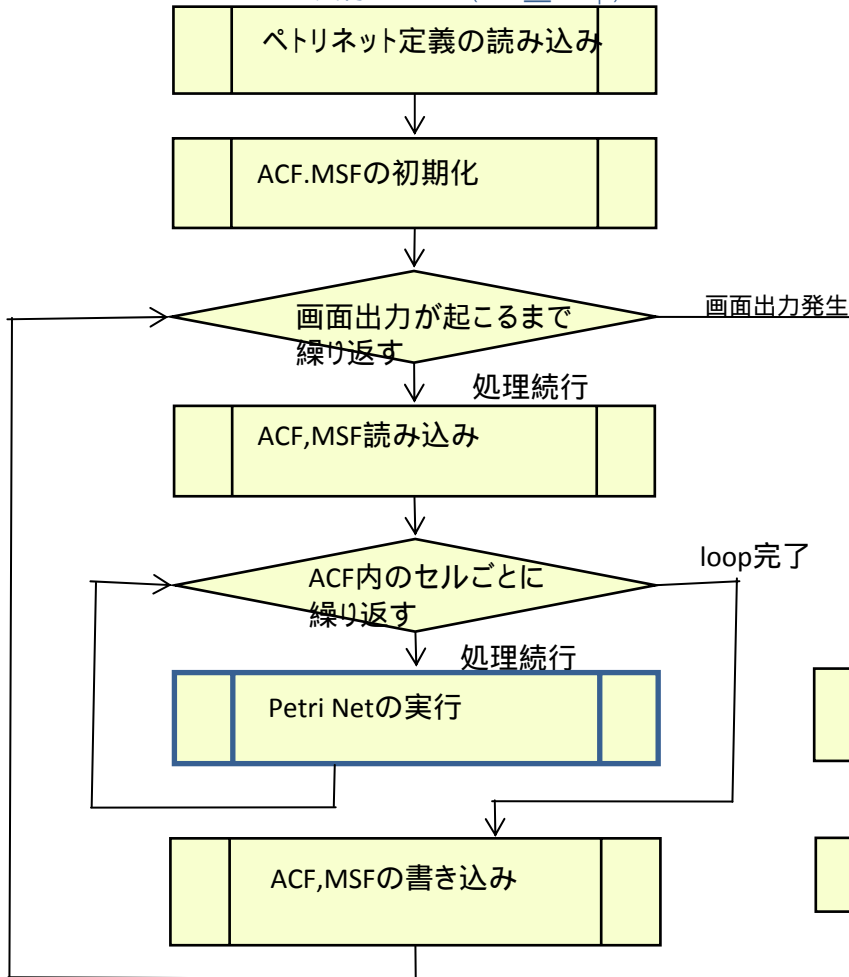


UC_00004_00005	cell id	UC_00004_00005
UC_00004_00005	to_display	red
UC_00004_00005	to_next	+1_+1:red:8



# セルペトリネット実行プログラムの流れ

セルペトリネットの実行ルーチン(run\_Step)



Petri Netの実行ルーチン(run\_PTIN)

