

45分で動かせる  
buri/escapeFlow入門

# 自己紹介



- いまい たかふみ
  - imai78(旧itengineer)
  - <http://d.hatena.ne.jp/imai78/>
  - buri/escafeFlowコミッタ

# 所属

- buri-ja
  - <http://groups.google.co.jp/group/buri-ja?hl=ja&pli=1>
- java-ja
- システムズデザイン株式会社
  - <http://www.sdcj.co.jp/>

おさらい

# buri/escaflow

- **B**usiness  
**U**nit  
**R**outing  
**I**ntegration

# huri/escafeFlowという 名称について

- 前のバージョンがHAMACHIだったとか
- 歴史
  - Buriの名称で開発
  - Seasarに寄贈され、S2Buriに
  - escafeProject(<http://www.escafe.org/>)に移動
    - escafeFlowに
    - でもみんな「ぶり」って呼ぶ

- 作者
  - **id:makotan**
- 何をするもの？
  - **Work state engine**
  - XPDLでフローを書き、それをプログラムの一部として動作させる

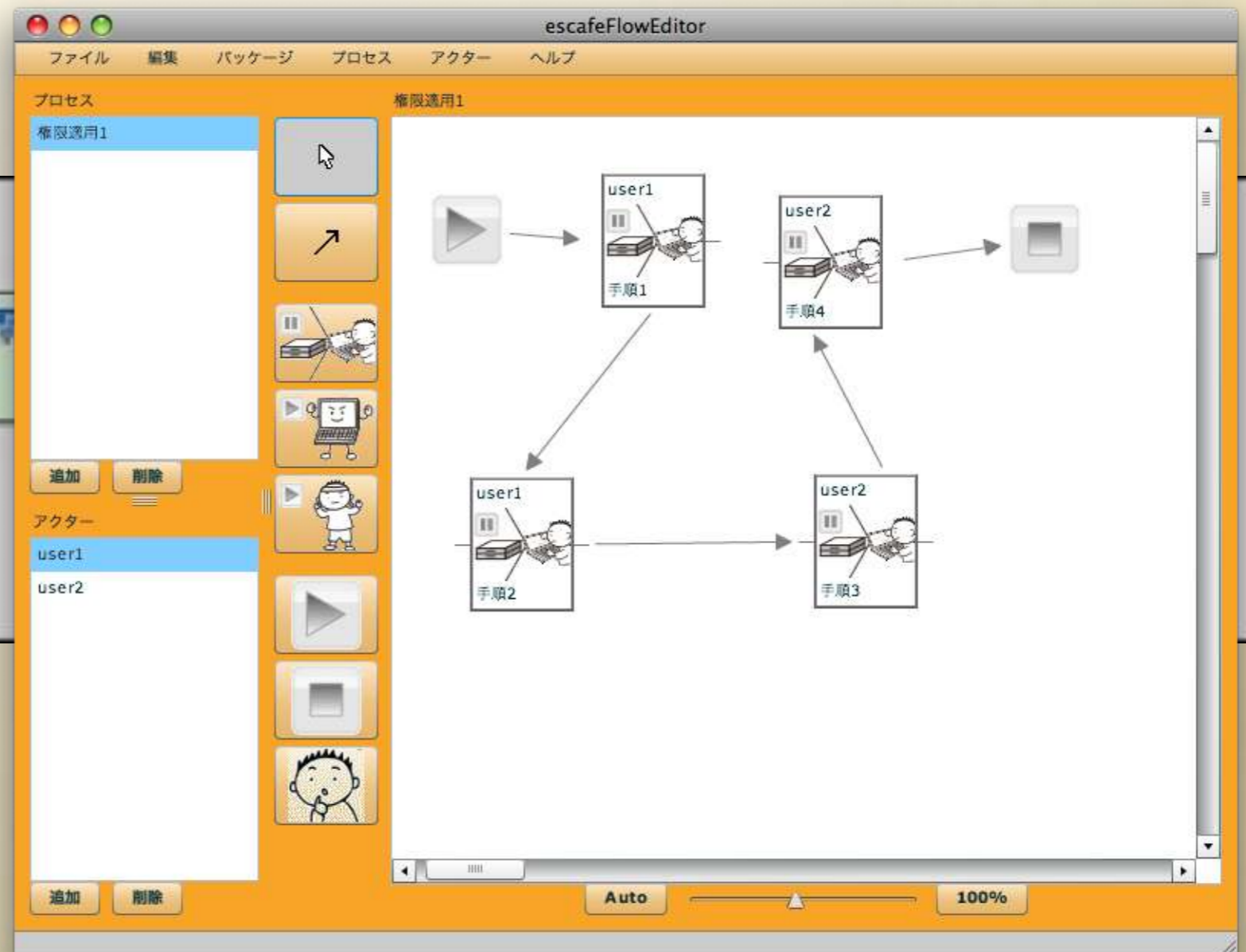
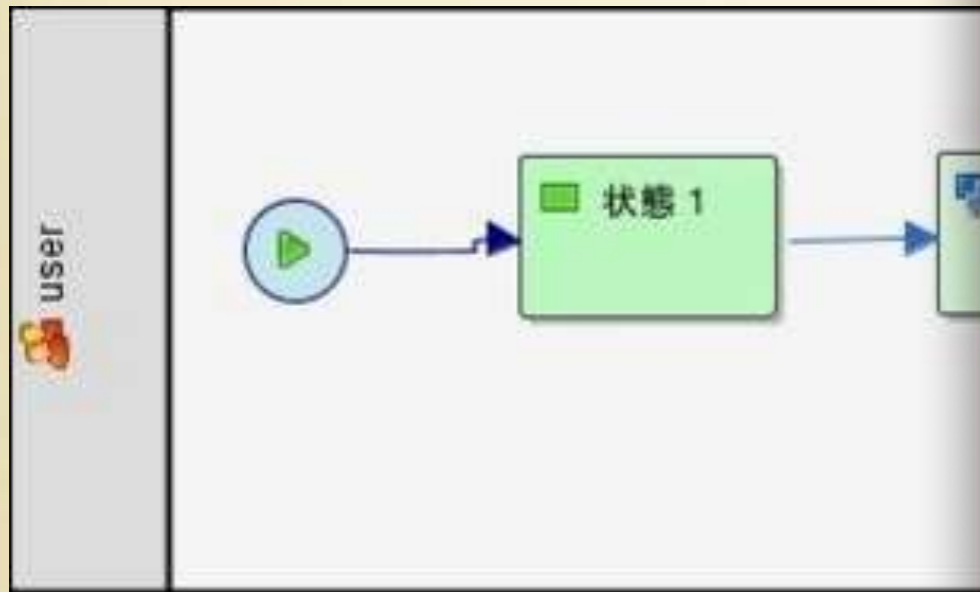
# 用語



# XPDL

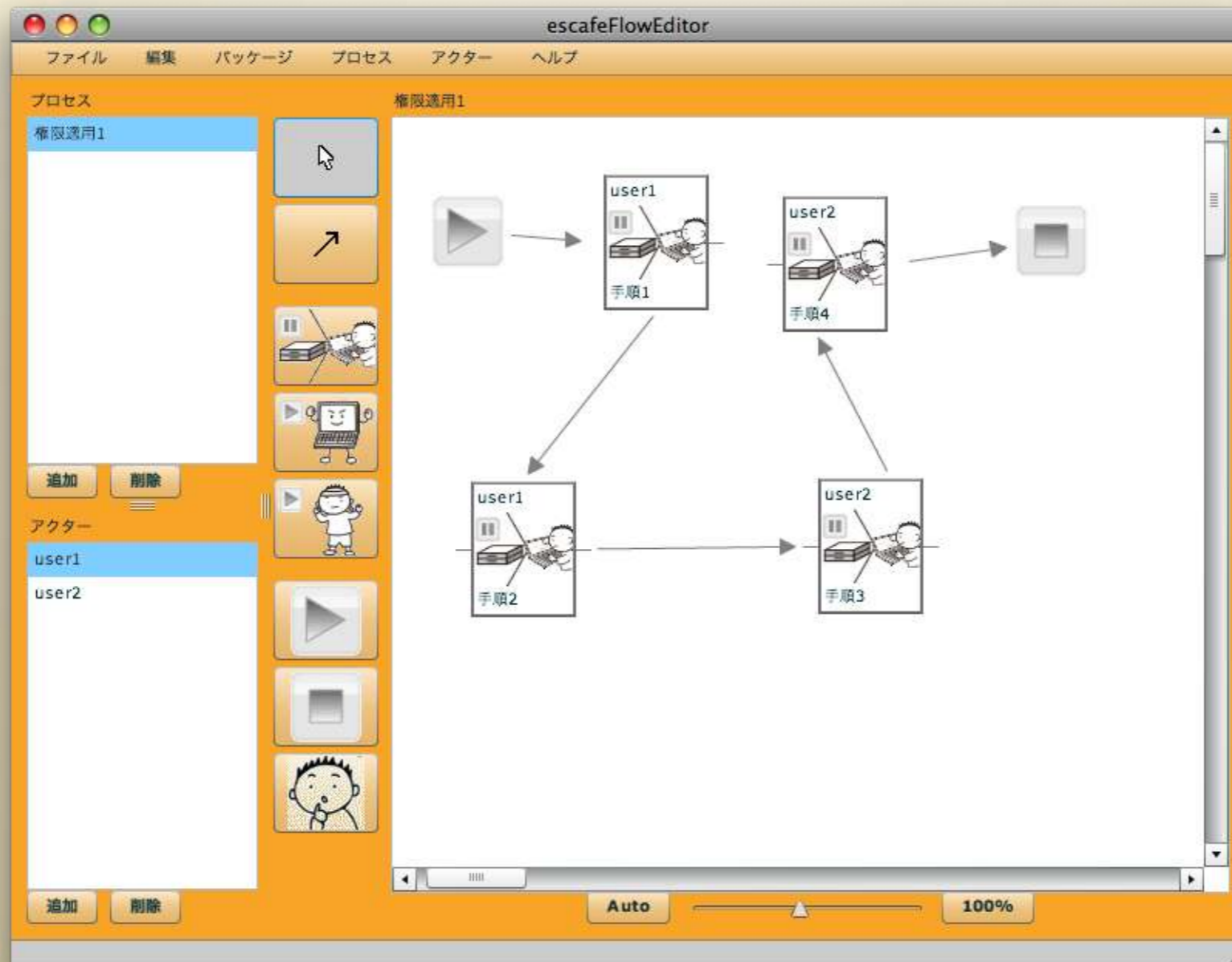
## XML Process Definition Language

ビジネスプロセス定義を行う為の標準形式



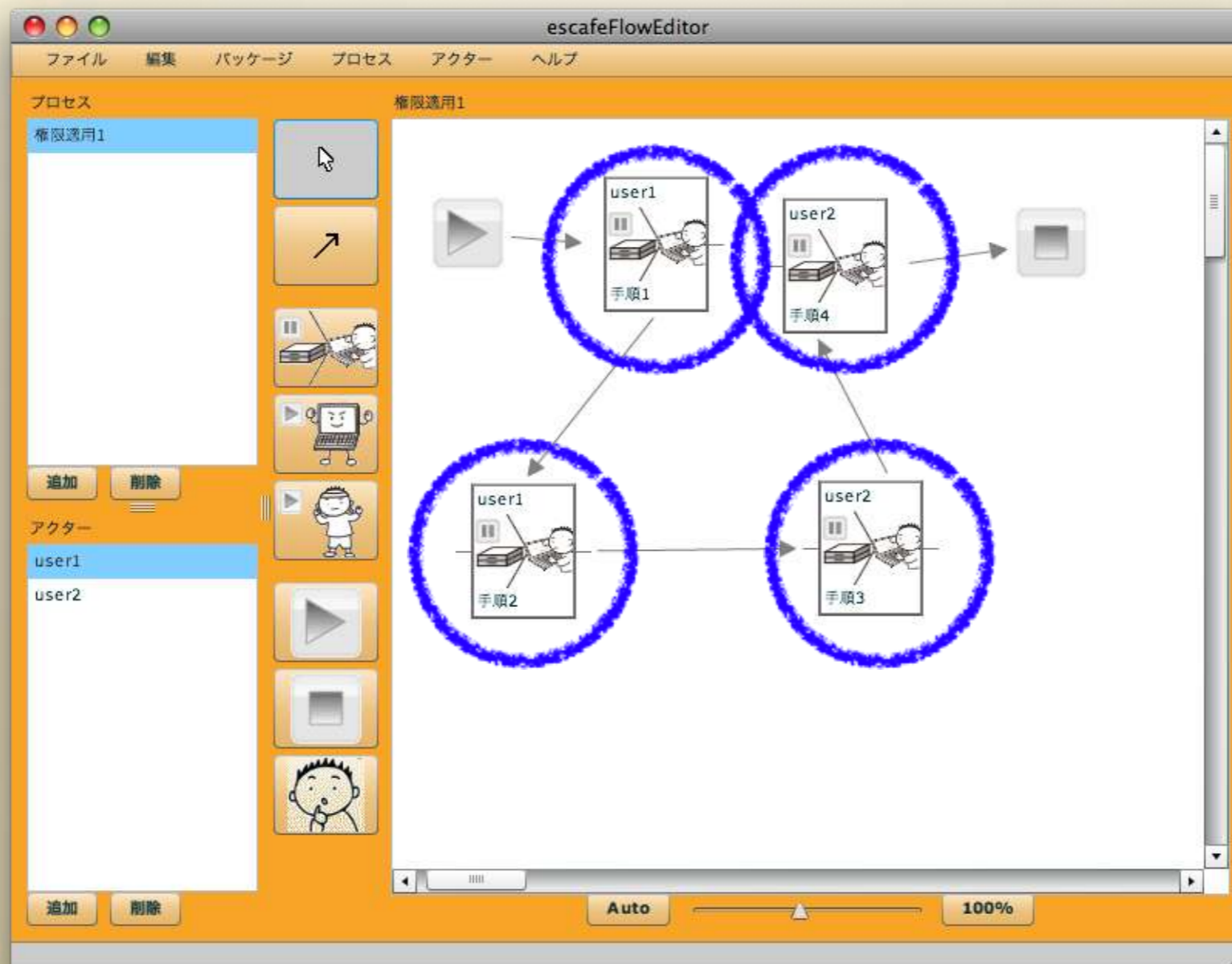
# フロー

- ぶりでは「プロセス」と表現します



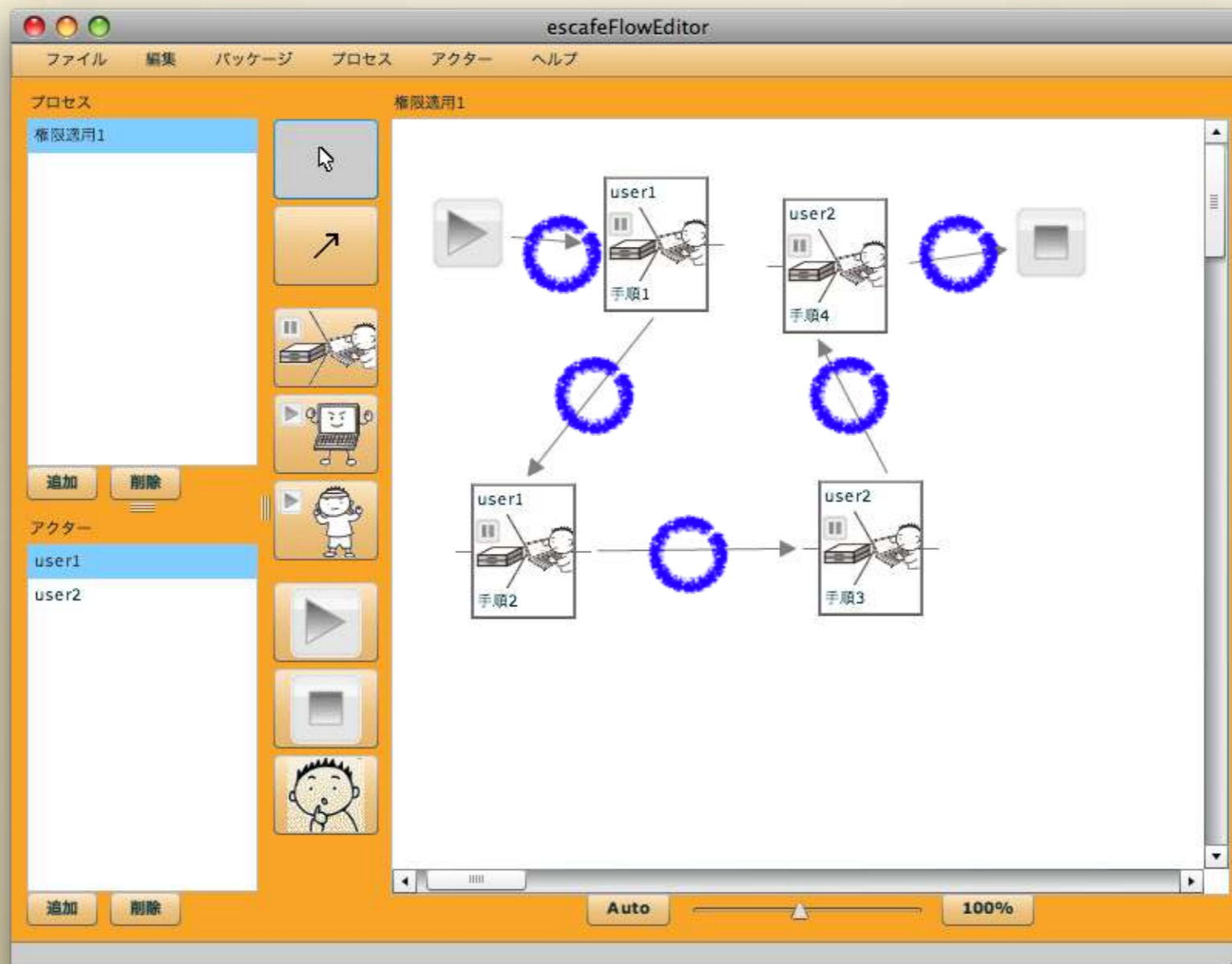
# アクティビティ

- フローに書かれた箱です



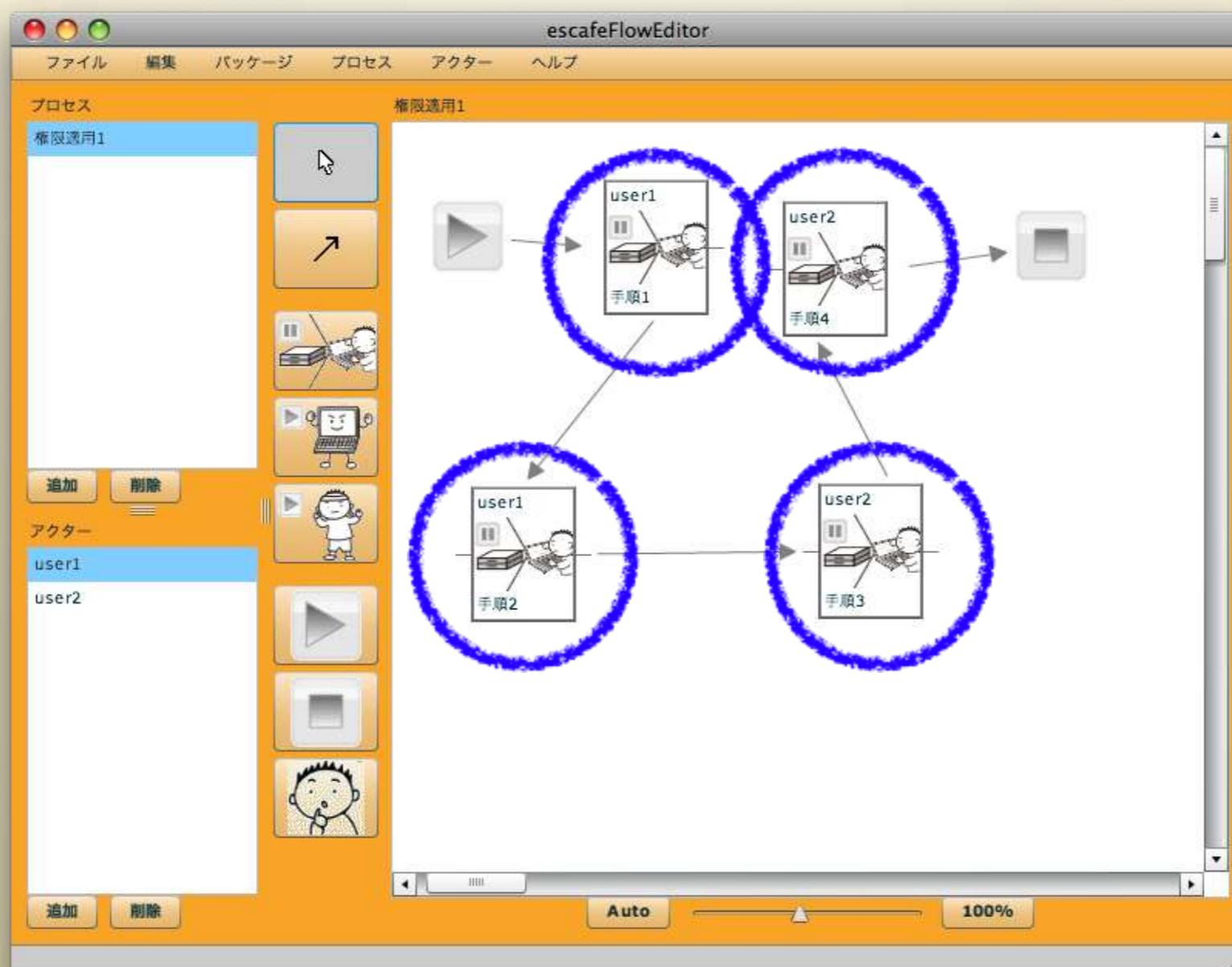
# トランジション

- 箱を繋ぐ「線」です



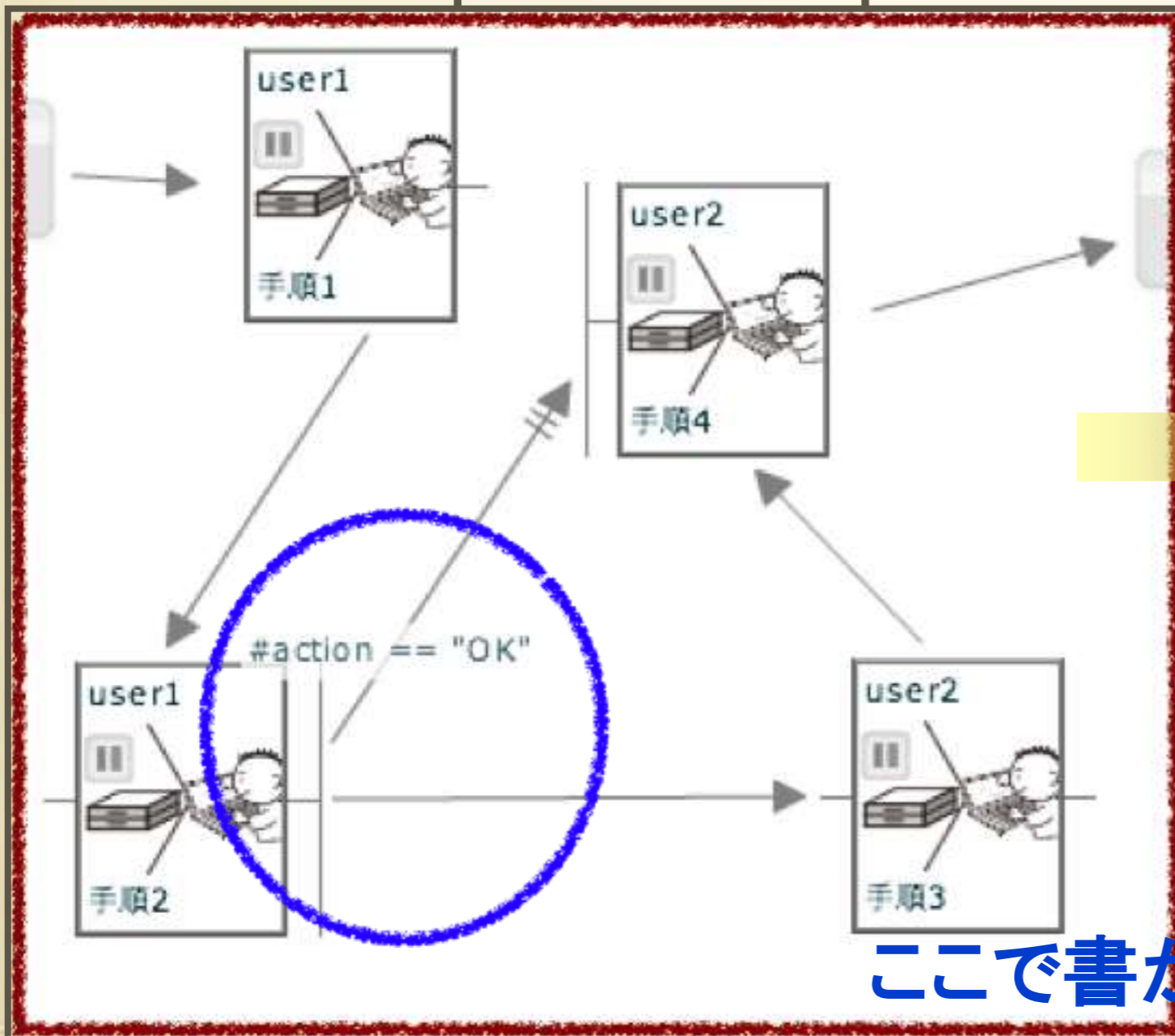
# 状態

- 「アクティビティ」です



# 分岐

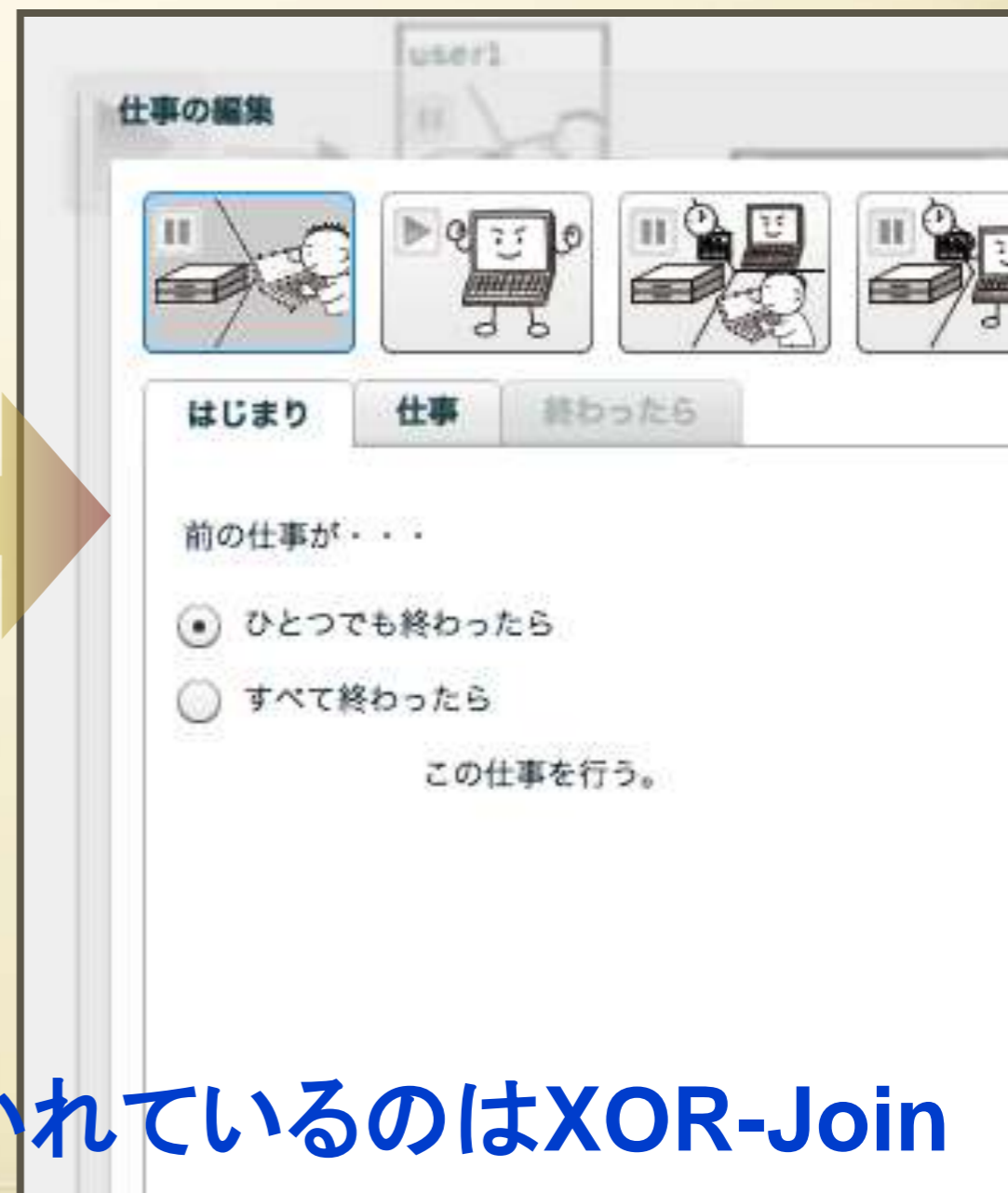
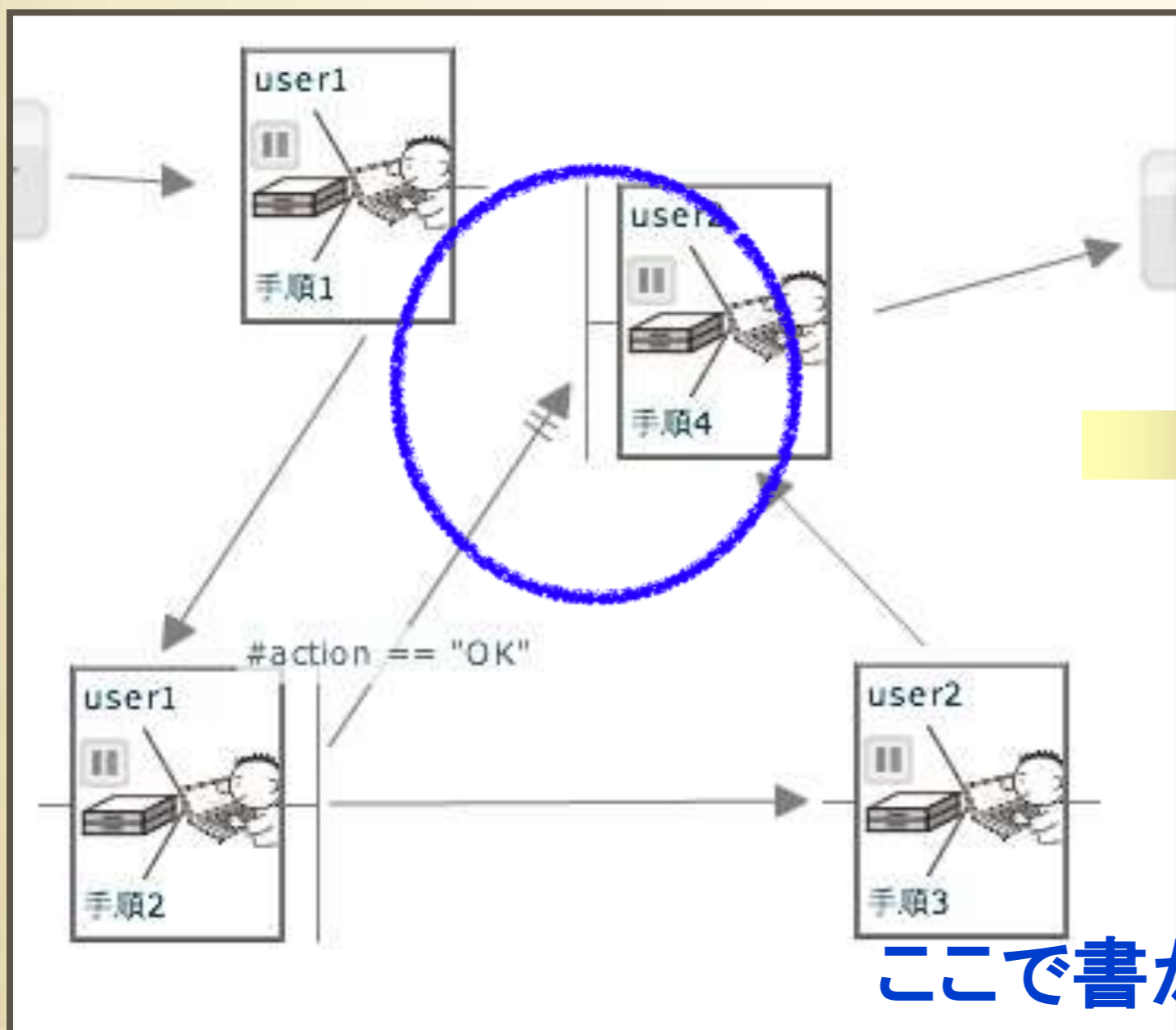
- AND-Split/XOR-Split



ここで書かれているのはXOR-Split

# 結合

- AND-Join/XOR-Join



ここで書かれているのはXOR-Join

- これらの表現形式を使って、
- buri/escapeFlowでは
- 「状態」の遷移フローを記述します
- そのフローはXPDLとして保存されます

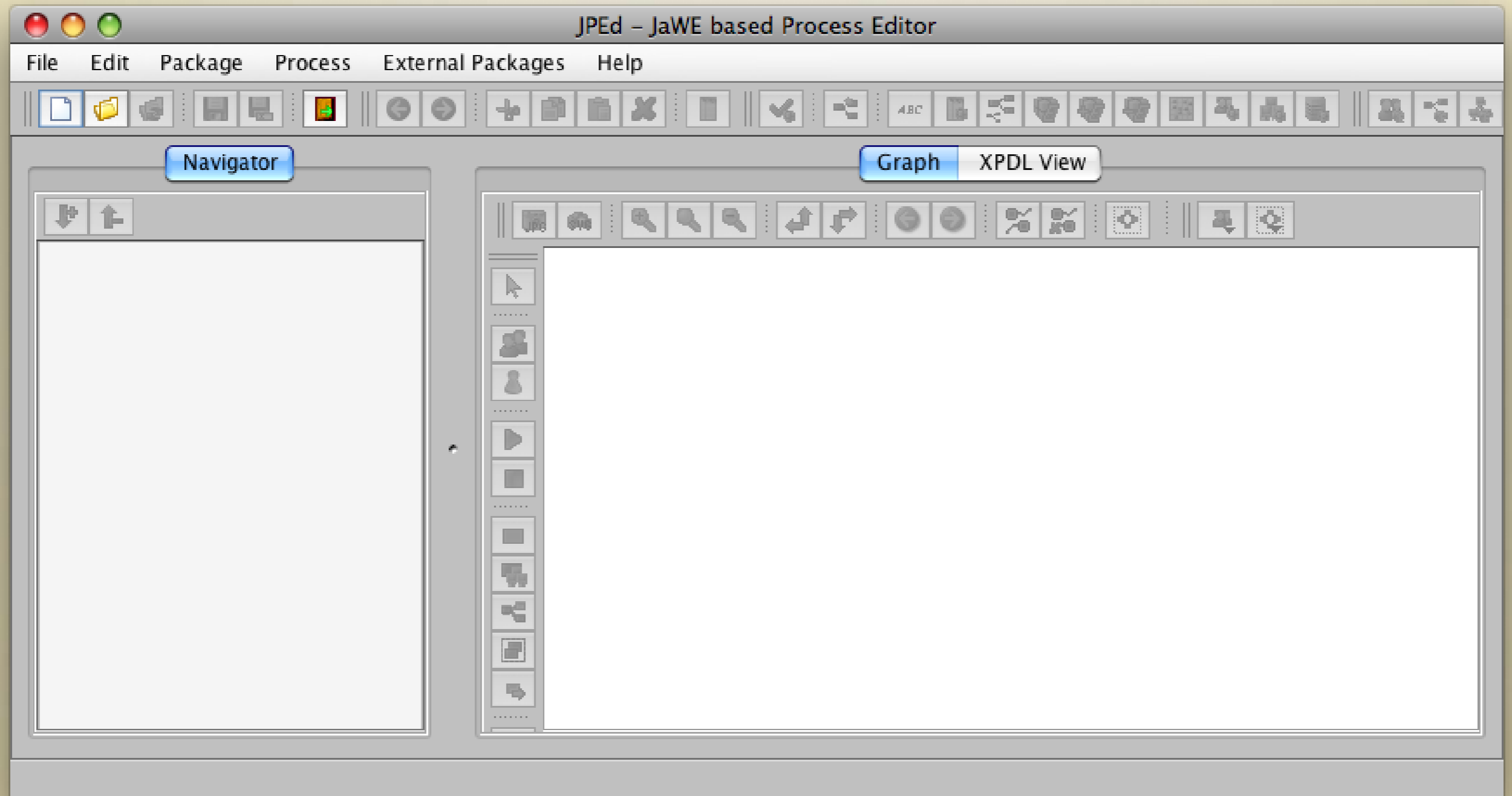


# ぶりの中のXPDL



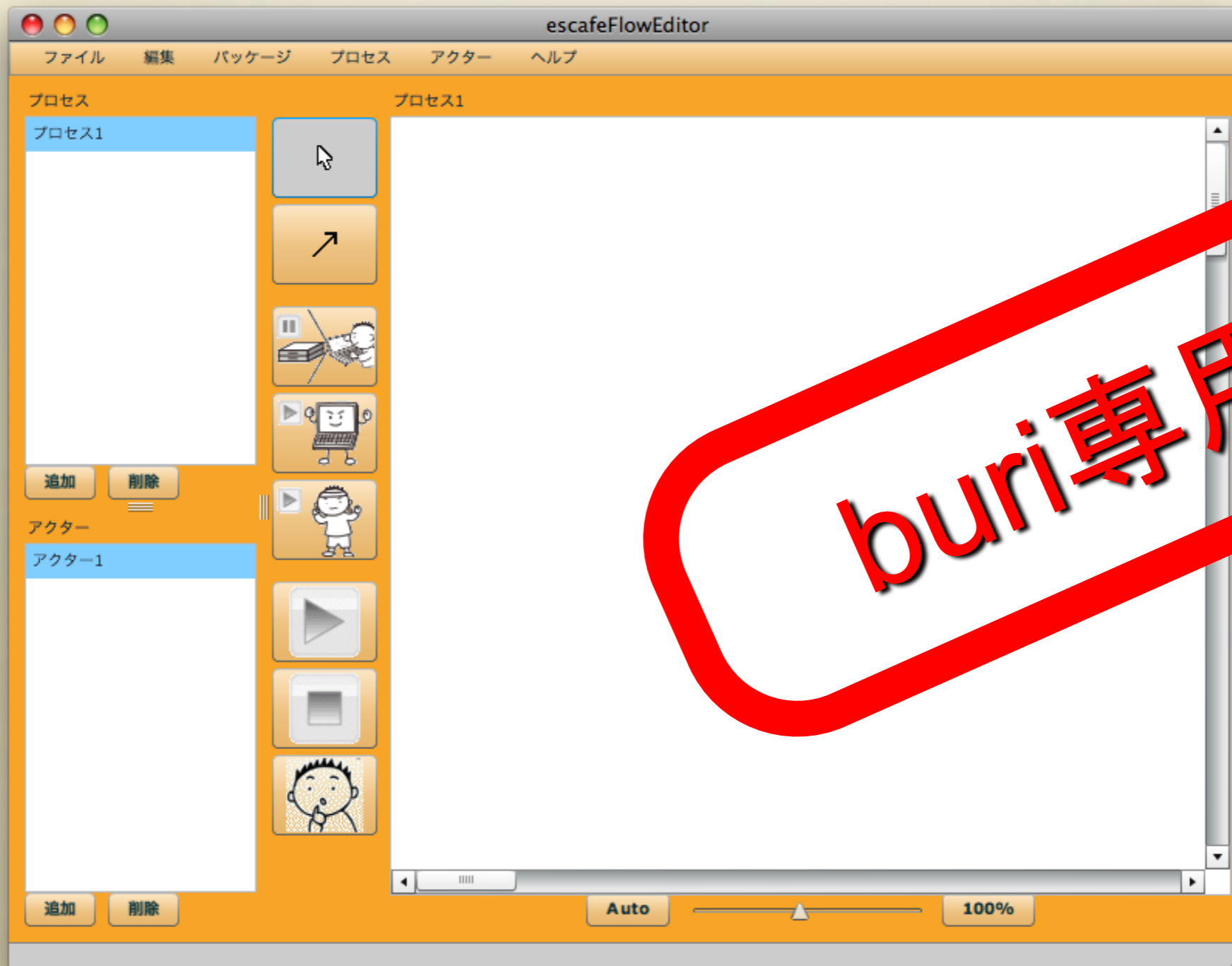
# XPDL-Editor

# JPEd



<http://sourceforge.net/projects/jped>

# escafeFlowEditor



huri専用

<http://www.escafe.org/escafeFlowEditor/>

考え方

## 〜[buri]設計思想

そういえば言ったこと無いなあ〜  
というより、そんなのあったっけ・・・  
あるとすれば・・・

1. IFとかフラグを減らすこと
2. 「アプリケーションの動き」と「ユーザの想像」の差を減らすこと

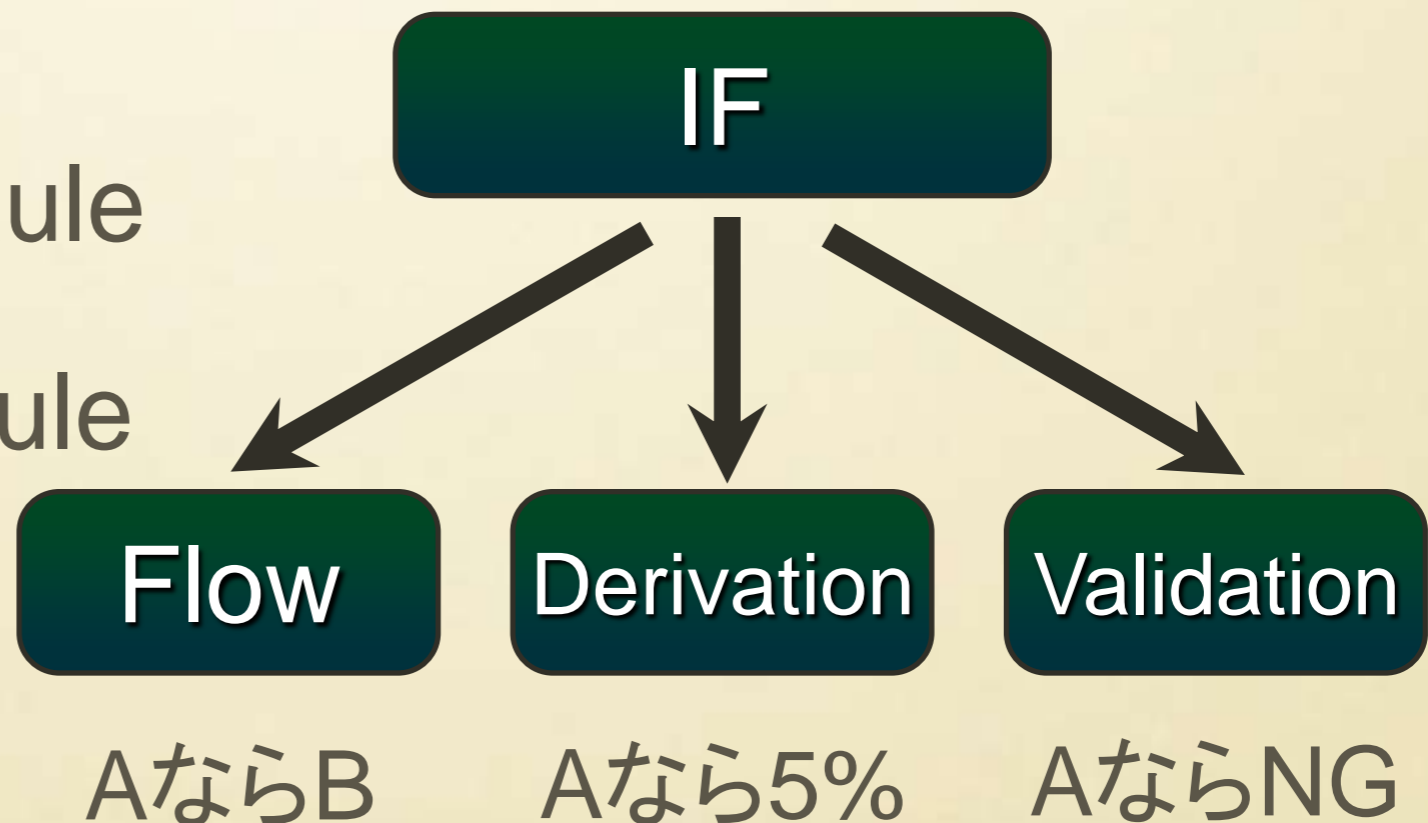
基本的にはこのくらいかな・・・つか、すくなっ！

## 〜[メモ]

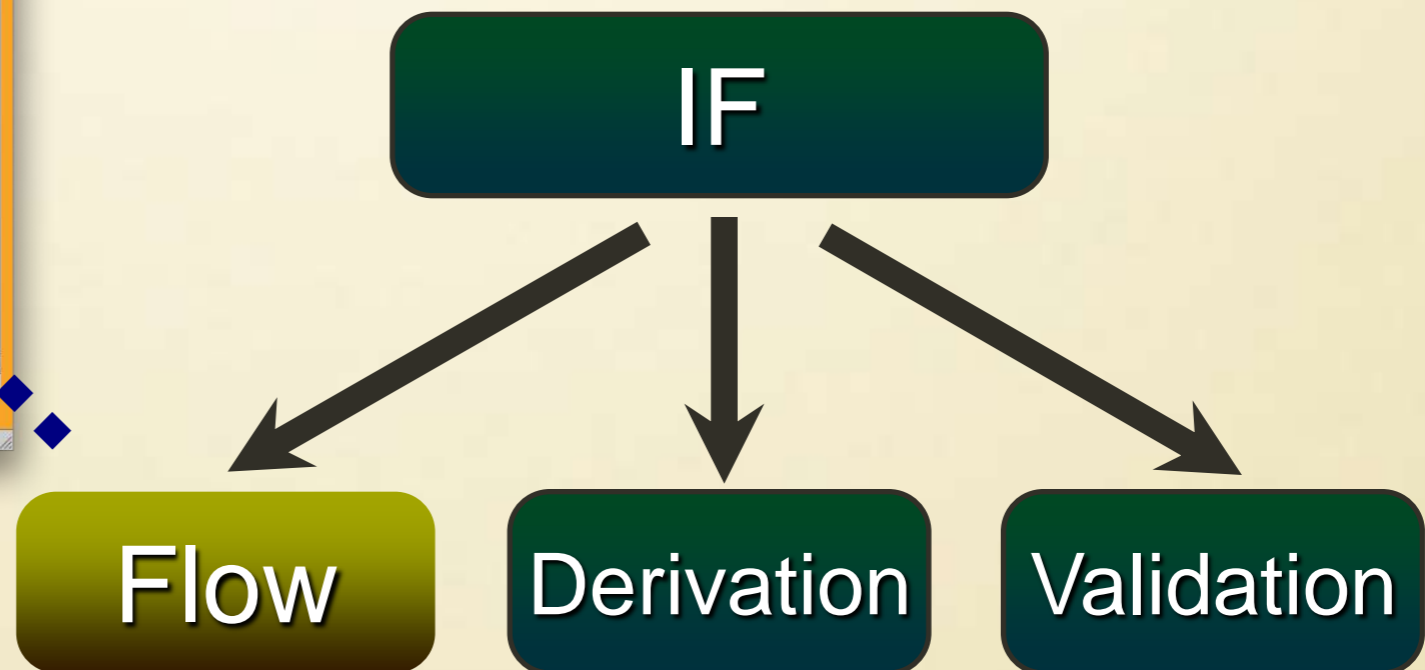
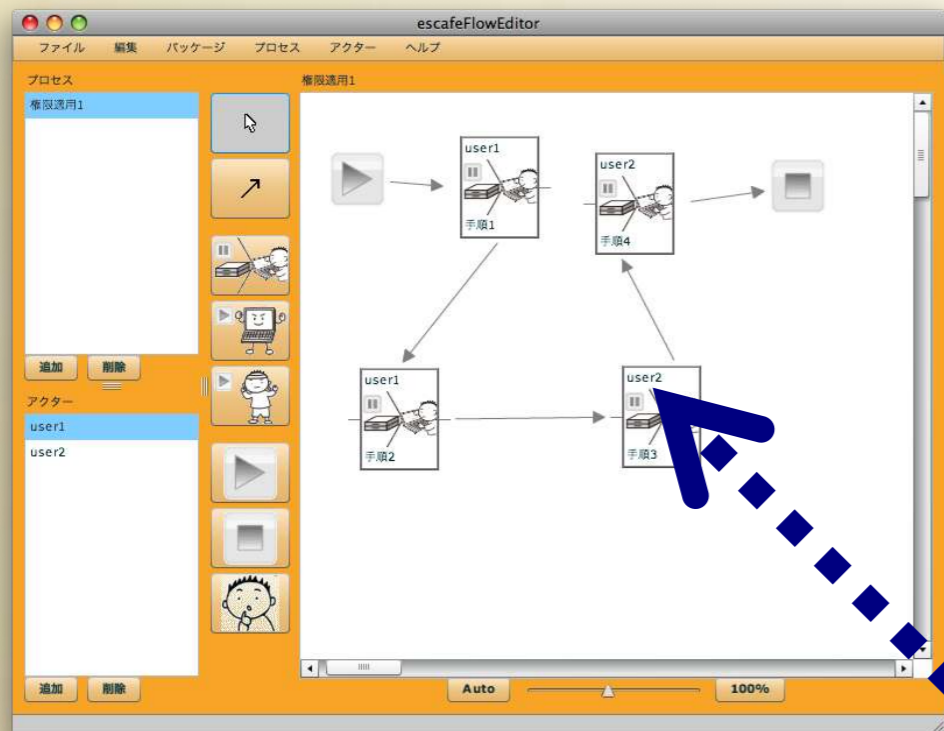
ファスト、Patriot製高速SSD「Torox」シリーズ - リード260MB/ライ

# そもそもIFとは・・・

- BusinessLogicの中のIFを整理すると3つの
- Ruleが見つかる
  - Flow
  - Derivation Rule
  - Validation Rule



- buri/escafeFlowは
- Flowを外部定義できるようにした





# 結果として

- 状態遷移は状態遷移で集中可能になる
- コード上のIFのフローに関するものが不要に
  - =テーブルレイアウトの「Xxフラグ」「Xxステータス」が不要になる

# 結果として

- ソフトウェアの構造がシンプルになる
  - 品質改善に繋がる
  - テストを楽にしていける
- 開発生産性が向上する
  - 楽なテストで済むようなコードを書けば済む

まとめ

# huri/escapeFlowという アプローチ

- IF文削減
  - その為の、状態遷移フロー
- ぶりを使う
  - =フローを書く+ちょっとした技術

# huri/escapeFlowという アプローチ

- このフローはテーブルと一対になります
  - ぶり独自の特徴
- つまりテーブル、フローの設計が非常に重要
  - ぶりを使わなくてもこれらは超重要

# テーブルとフローの設計

- 今回はさらっとしか扱いません
  - 時間が限られておりますので><
- このへんに興味のある方はぜひ・・・

**buri-ja勉強会へ！**

# まとめのまとめ

- **非正規形のプロセスを状態**
- **という観点で正規化する**
- **それがBuri/escapeFlow**