Independent
Verification &
Validation

Japan Manned Space Systems Corporation
JAMSS

STAMPとFRAM

1st STAMP Workshop Japan
2016.12.7
有人宇宙システム株式会社
野本秀樹

略歴

野本秀樹
有人宇宙システム株式会社

- 1996 宇宙ステーション計画で我が国初のIV&VをJAXAと開始し、現在IV&Vとりまとめ
- 2003-2004 MIT Nancy Leveson研究室勤務
- 2008-2016 HTV飛行管制官
- 2013-2014 JAXA理事長直轄プロジェクトで、次世代HTV概念設計をレジリエンス・エンジニアリングにより実施

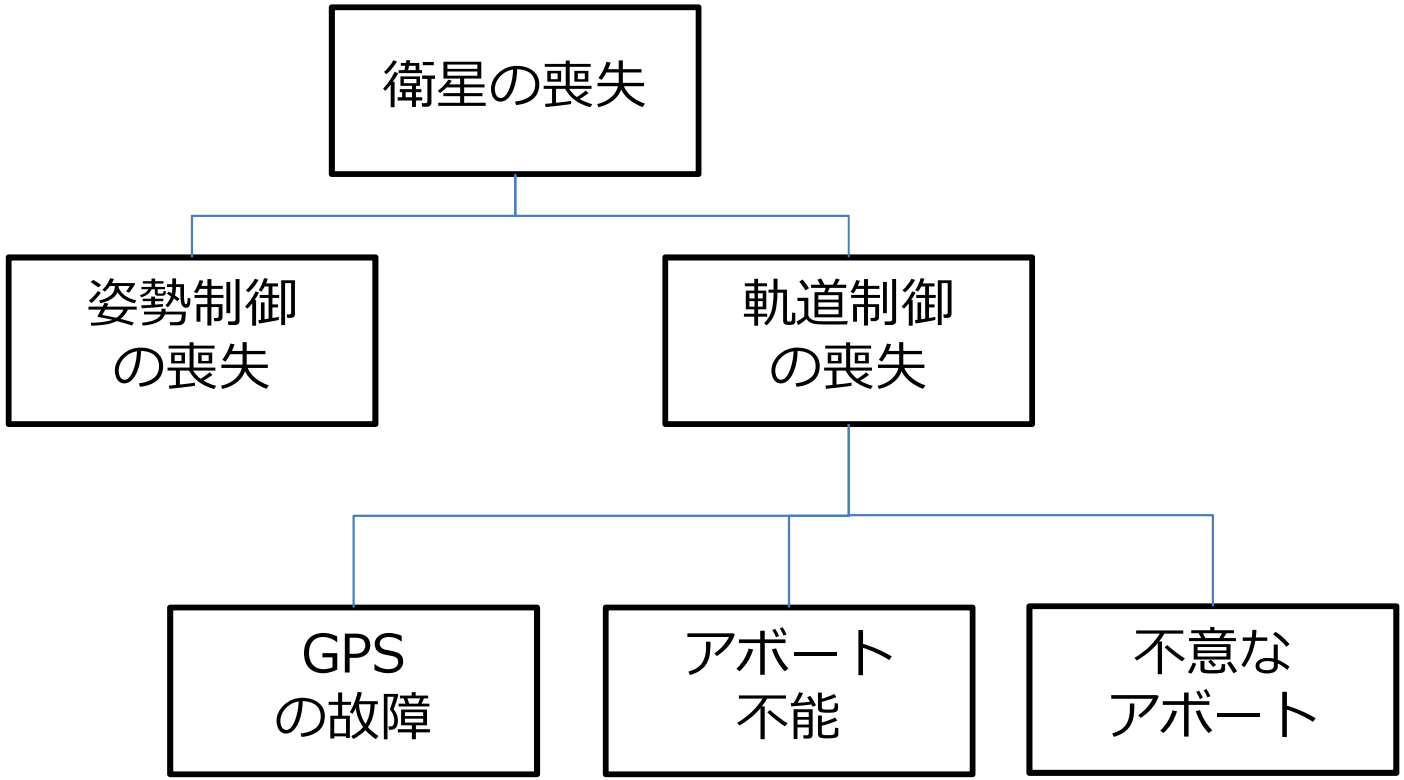
目的

テーマ：成功要因の分析による安全解析
Safety by Success Analysis

多くの安全の専門家には違和感があるはず。
何故成功要因の分析が安全に必要なのかを
お話したいと思います。

ツール：STAMPとFRAMのHybridで。

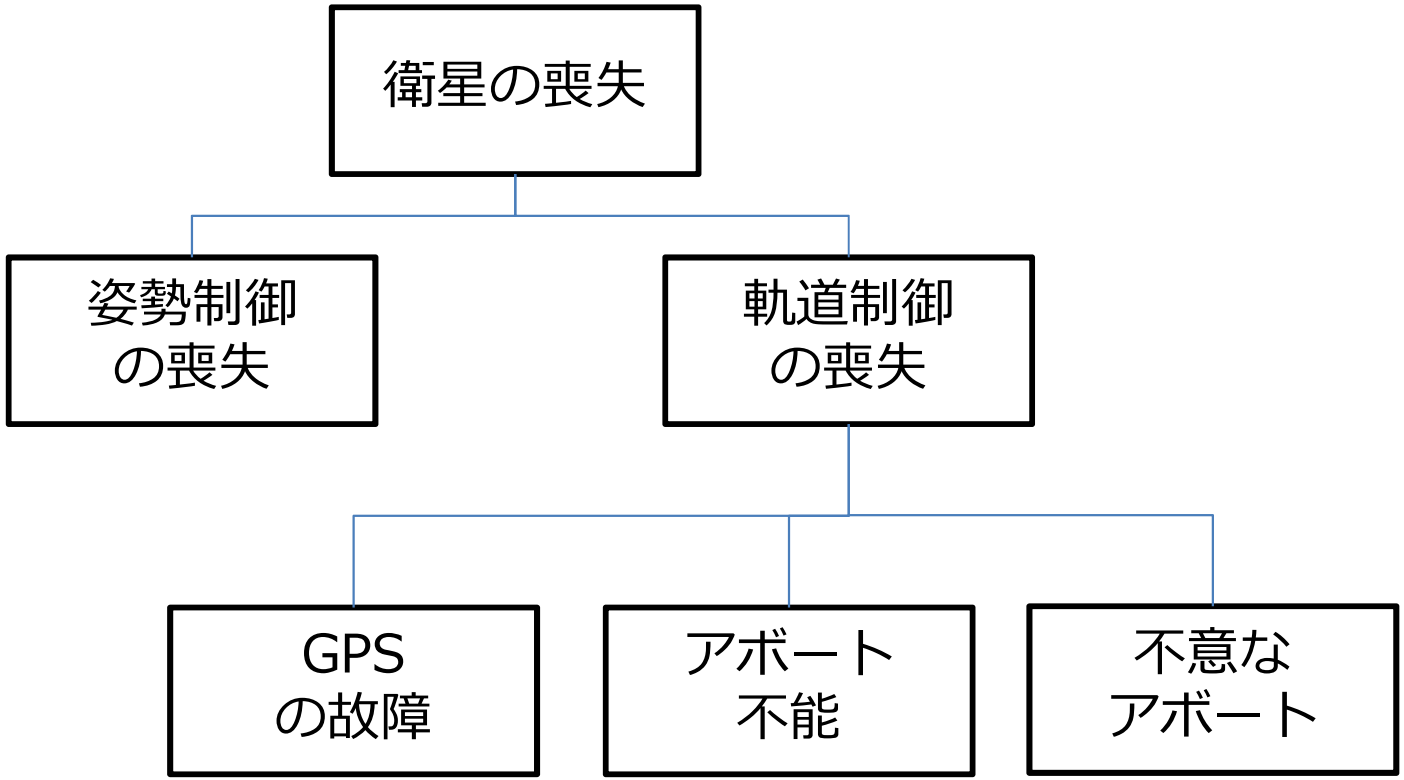
FTA



FTA

従来の安全解析は、**Failure**を分析。

Chain of Failuresによってハザードが発生すると仮定する。



New Paradigm

“Software does not fail.”

Nancy Leveson

“Engineering Safer World”

13th Workshop Of Critical Software System,

Jan 2016

New Paradigm

“Software does not fail.”

Nancy Leveson
“Engineering Sa
13th Workshop Of C
Jan 2016

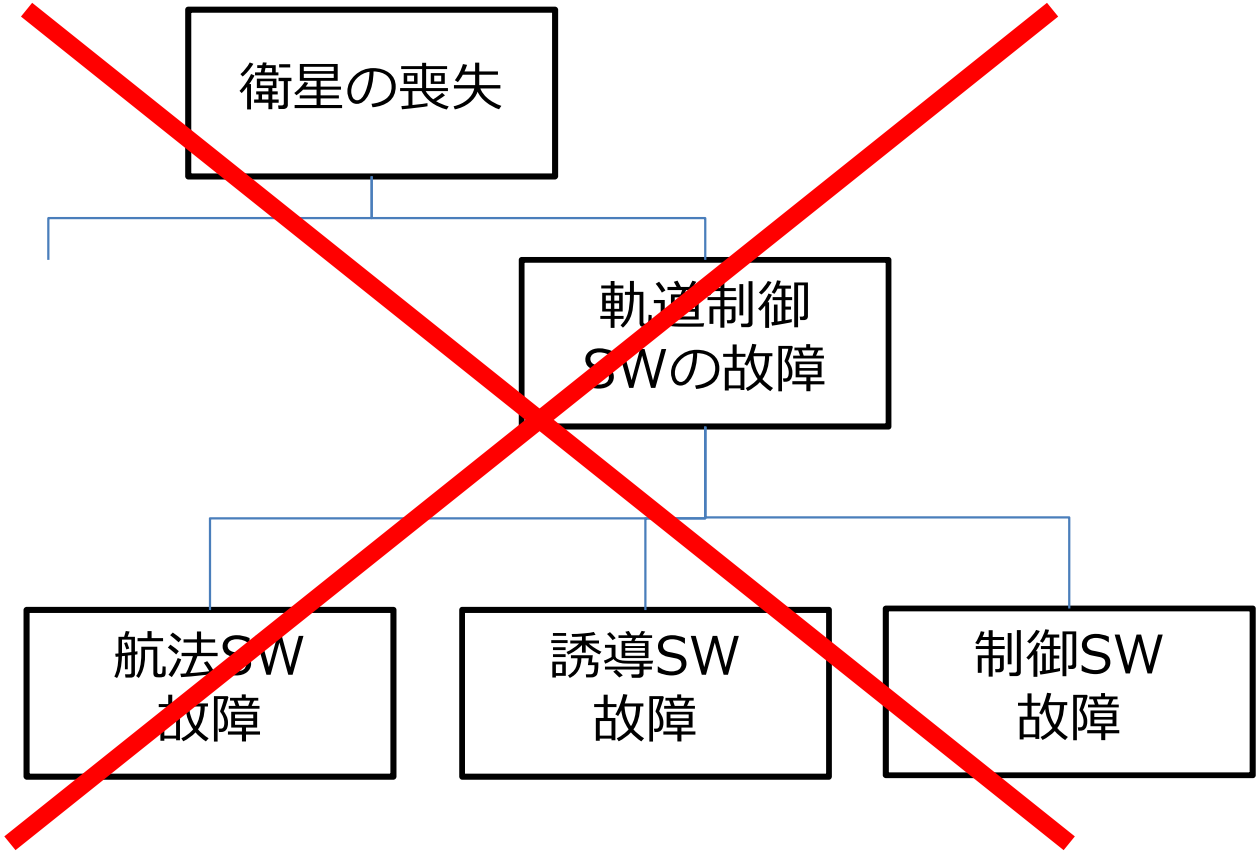
Mars Polar Lander Accident

“All components worked as required.”

John Thomas

New Paradigm

No FTA for software which does not fail!



New Paradigm

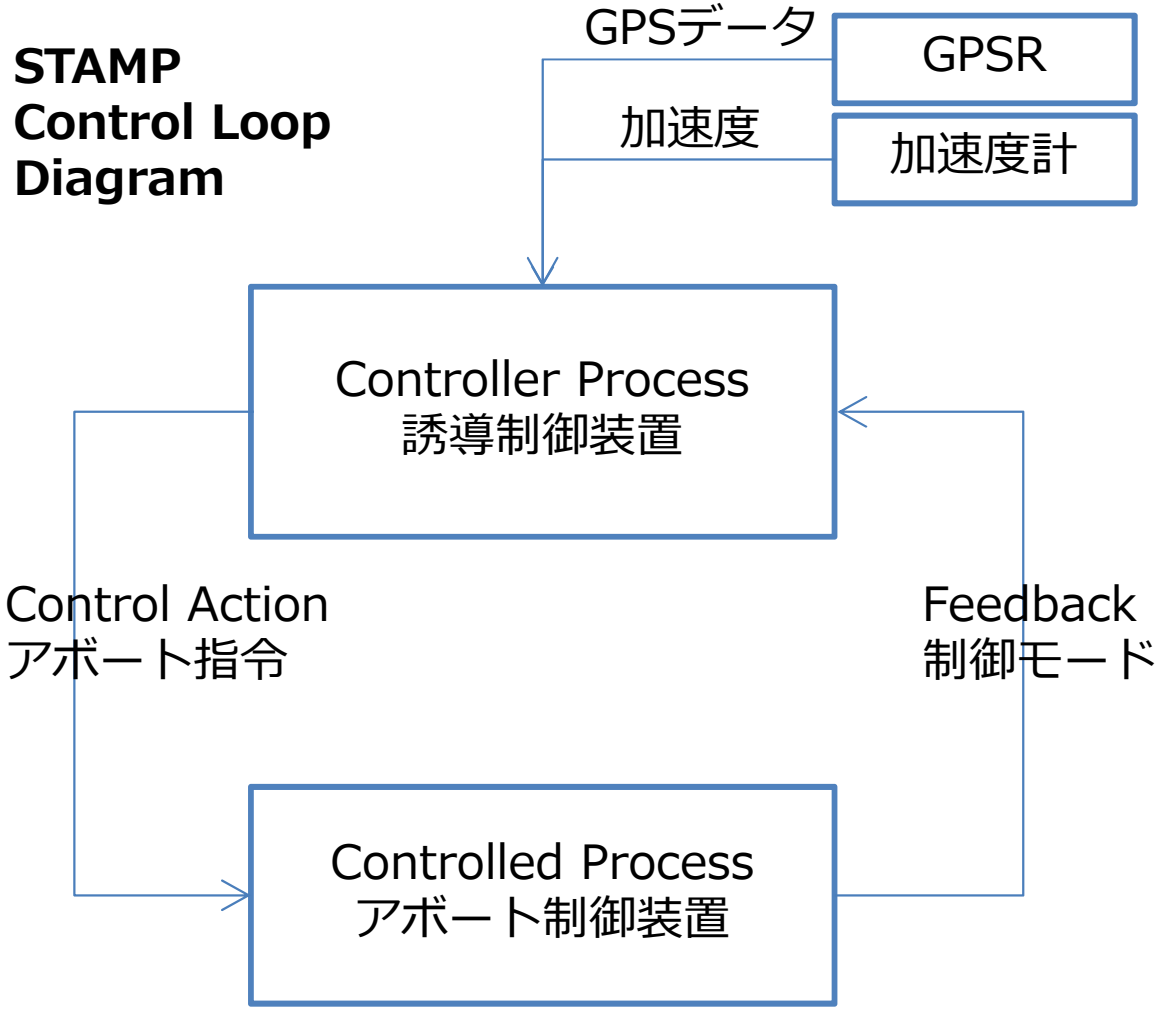
どうする？

New Paradigm

FTAのように、故障の連鎖をモデル化するのではなく、

Success Story（どのように安全が制御されているのか？）をモデル化する。

STAMP Control Loop Diagram

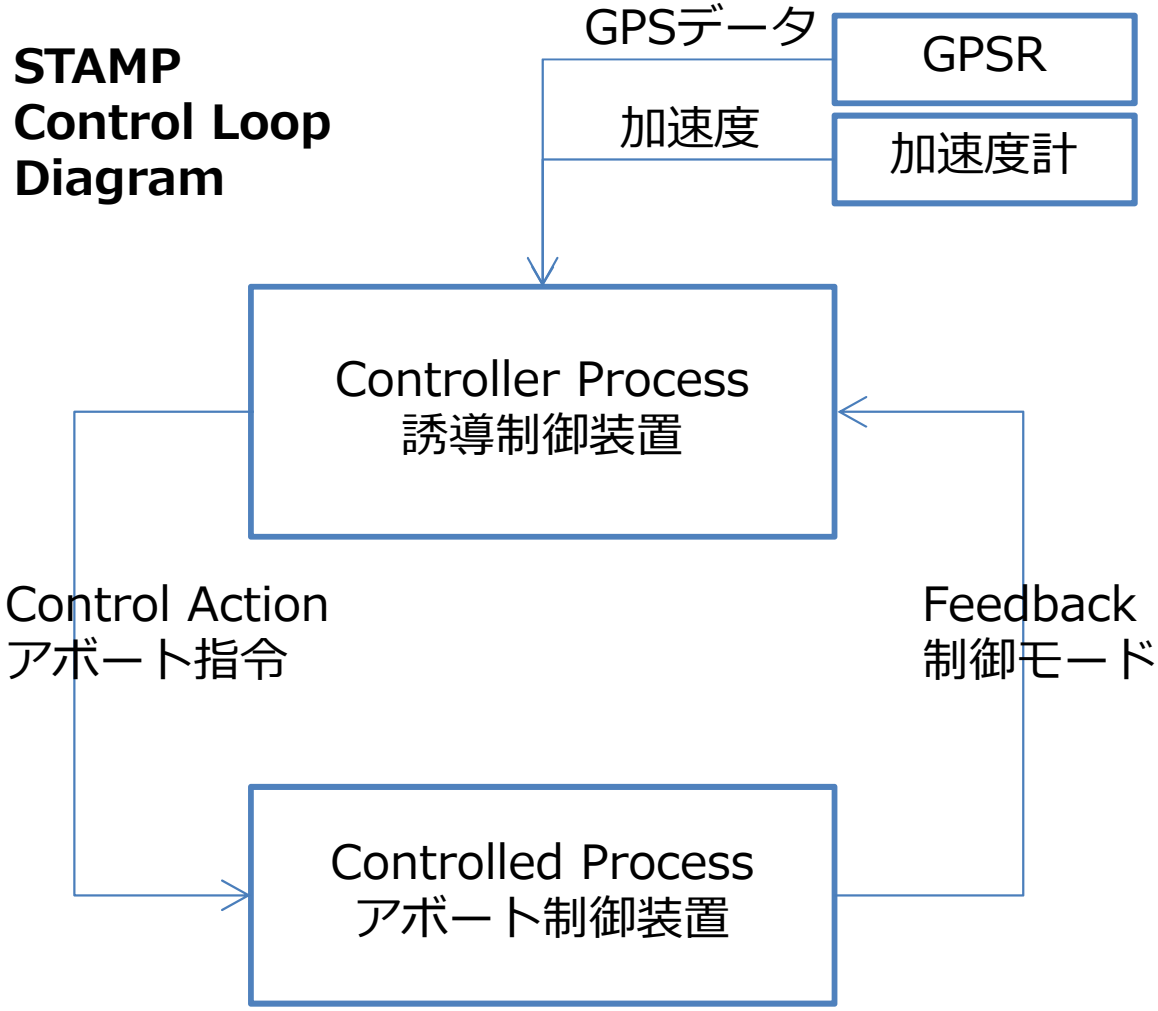


New Paradigm

Success Story（どのように安全が制御されているのか？）をモデル化し、それが破たんする理由を分析する。

Success Storyの分析こそが最も重要。

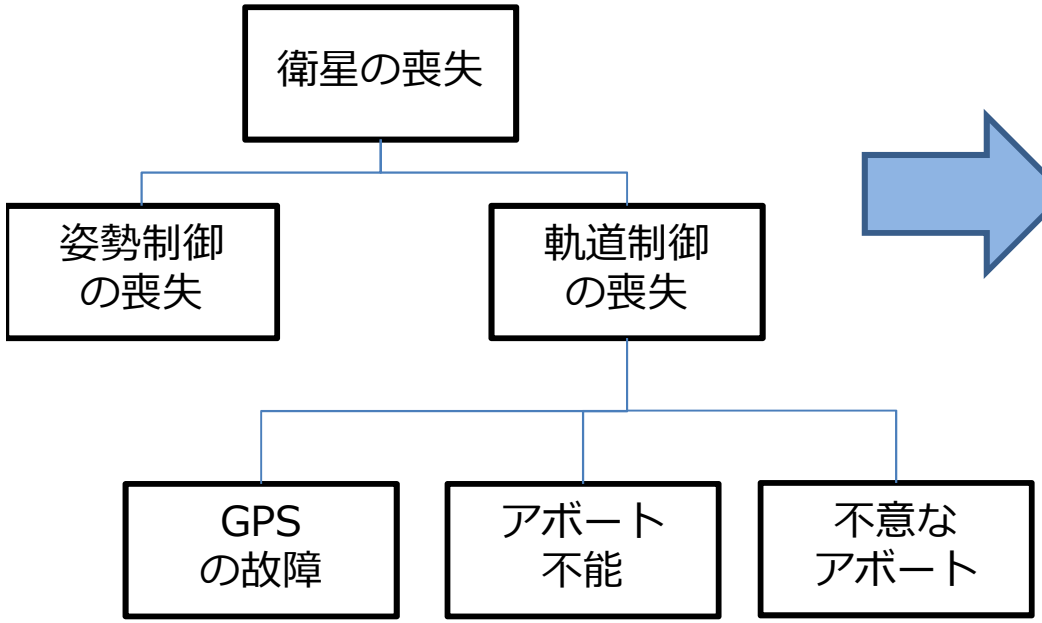
STAMP Control Loop Diagram



New Paradigm

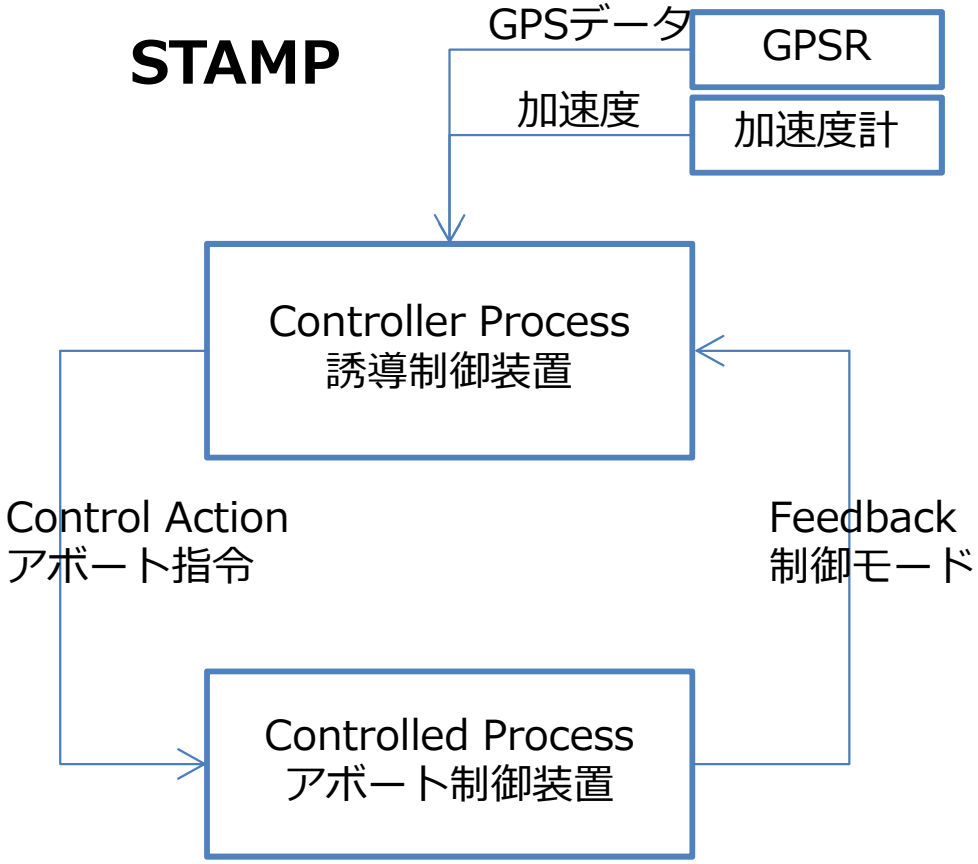
Failure Model

FTA



Success Model

STAMP



New Paradigm

人工知能時代到来

従来のシステムよりも、不確定性が高く、
Success Modelが見えにくい。

New Paradigm

人工知能時代到来

従来型：ゆらぎの排除が安全の鍵。

人工知能：ゆらぎこそが安全性の鍵。

New Paradigm

Markov Network

確率論的な推論ネットワーク

New Paradigm

Deep Learning

多重black box

レイヤ

New Paradigm

流動的で不確実性が高いシステムの
Success Modelは、作るのが難しい！

「人間の脳は、何故優秀なのか？」

It is hard to build the success model of non-deterministic systems as human brain.

New Paradigm

流動的で不確実性が高いシステムの
Success Modelは、作るのが難しい！

「人間の脳は、何故優秀なのか？」

複雑で非決定論的なシステムの成功要因を明らかにするために、もっと不確実なものの良さを見直そう！もっと成功に着目しよう！

New Paradigm

人間や動植物の不確実性・しなやかさに着目し、成功要因の分析を目指す安全工学

レジリエンス・エンジニアリング

New Paradigm

"We know something about what goes **wrong**, but almost nothing about what goes **right**!"

Erik Hollnagel
"Resilience Engineering and
the next generation of safety"
Jan 2016 @JAXA

FRAM

FRAM（Functional Resonance Analysis Method：機能共鳴分析手法）

＝ レジリエンスエンジニアリングのモデリング手法

物事が失敗する理由ではなく、

物事が成功する理由を分析する安全解析

FRAM

FRAM（Functional Resonance Analysis Method：機能共鳴分析手法）

物事が失敗する理由ではなく、
物事が成功する理由を分析する安全解析

実際にどのように成功要因の分析をする？

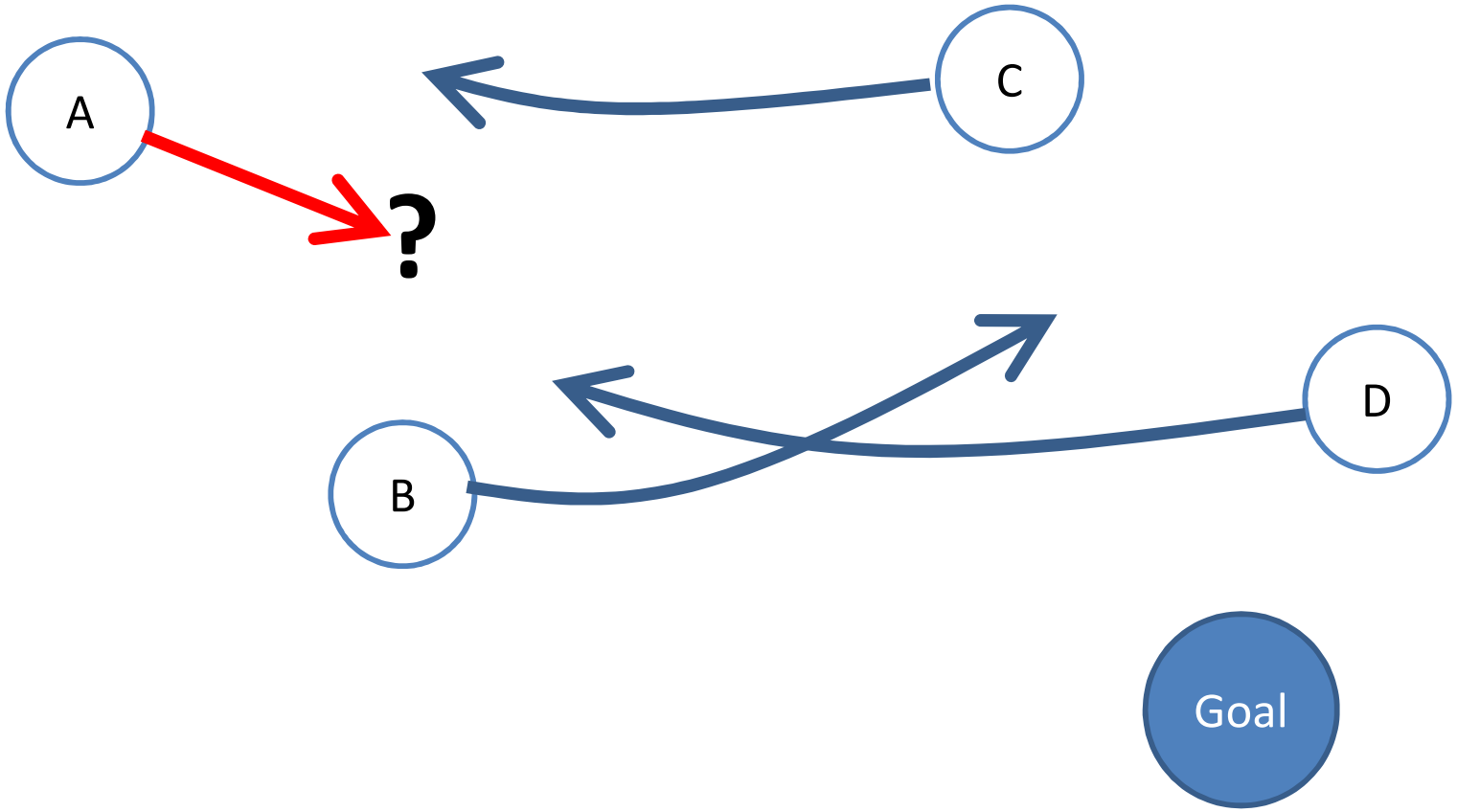
Analysis Example

新宿駅のコンコース

- No Fixed Rules
- No Fixed Safe Paths
- No Fixed Actors

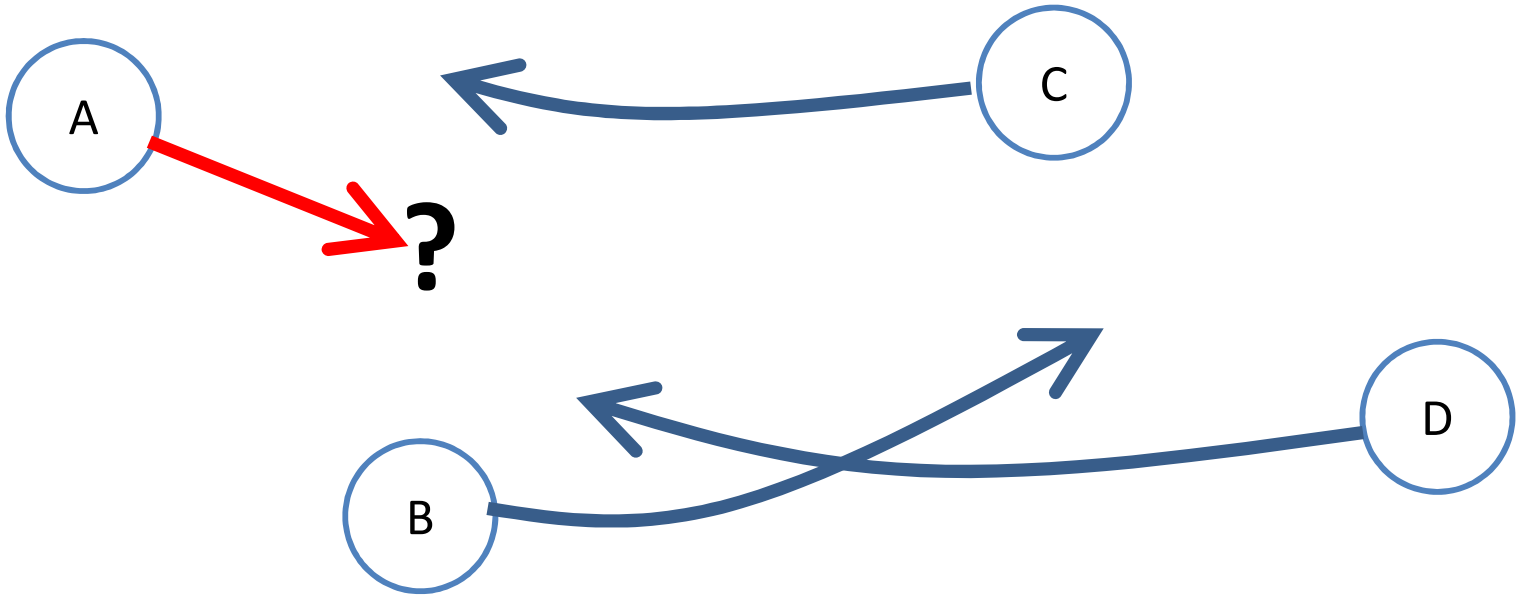
- どうやって衝突せずに、歩けるか？
- How to walk safely in Shinjuku station?

How to Survive in Shinjuku St



How to Survive in Shinjuku St

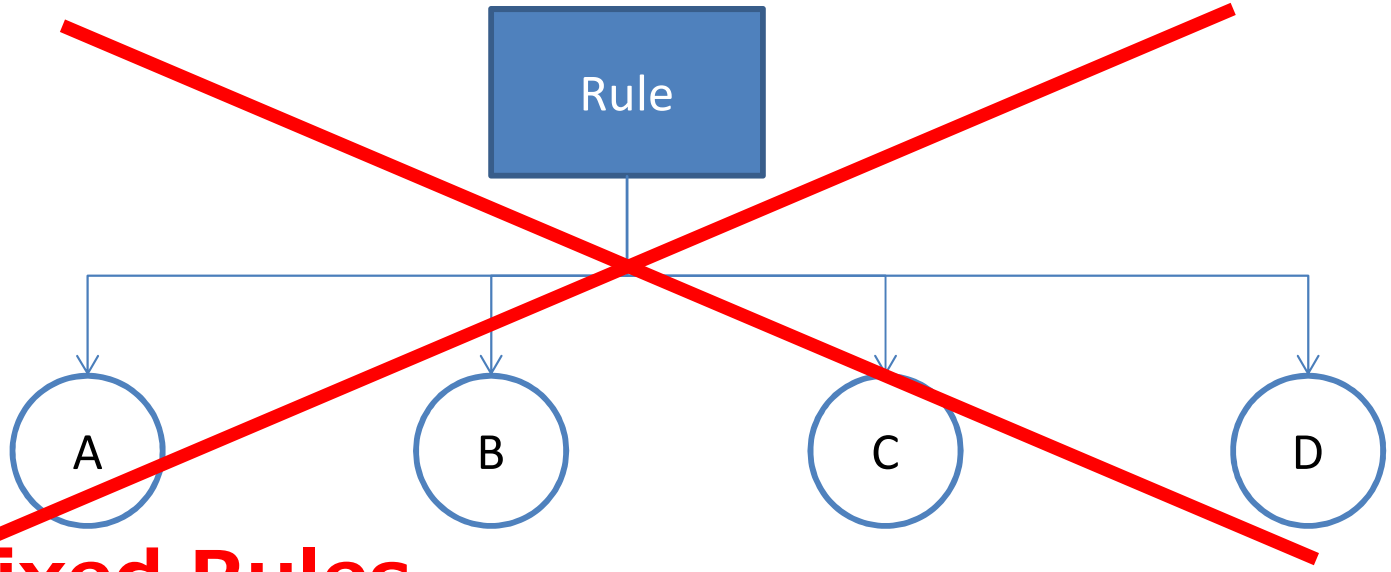
How A can succeed?



- No Fixed Rules
- No Fixed Safe Paths
- No Safety Controller

How to Survive in Shinjuku St

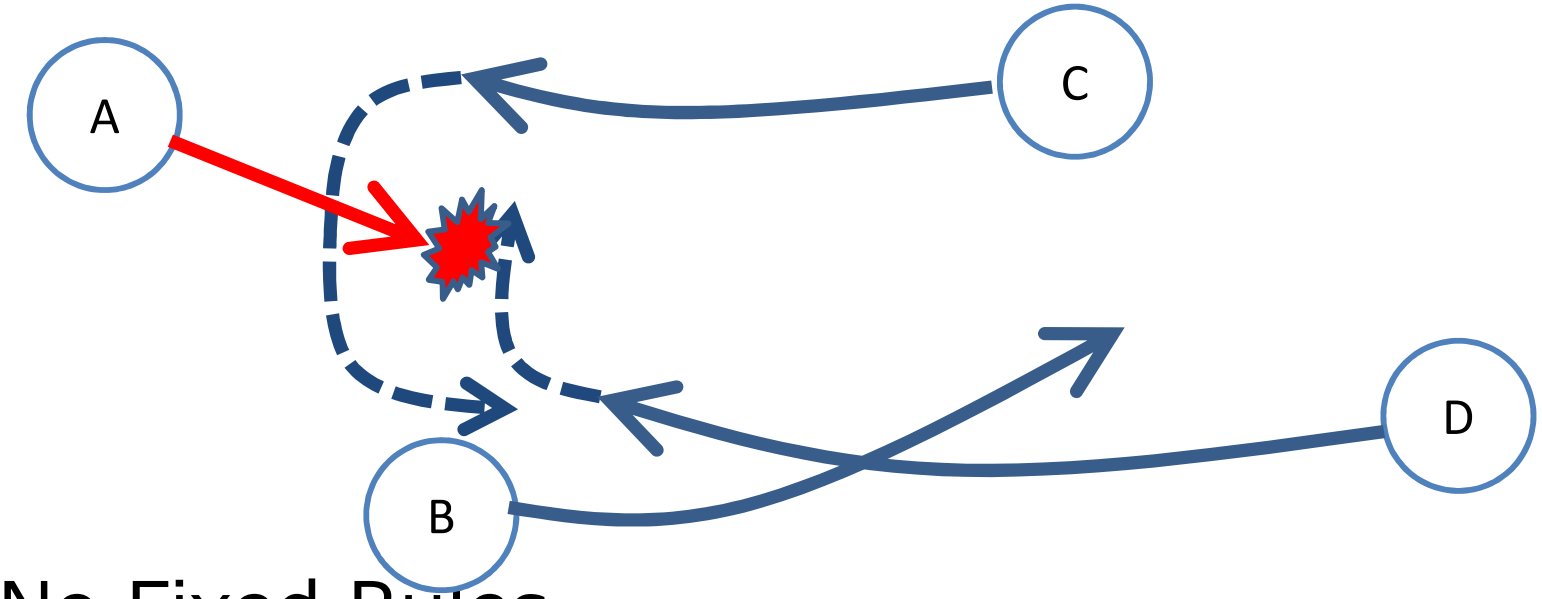
How A can succeed?



- **No Fixed Rules**
- No Fixed Safe Paths
- No Safety Actors

How to Survive in Shinjuku St

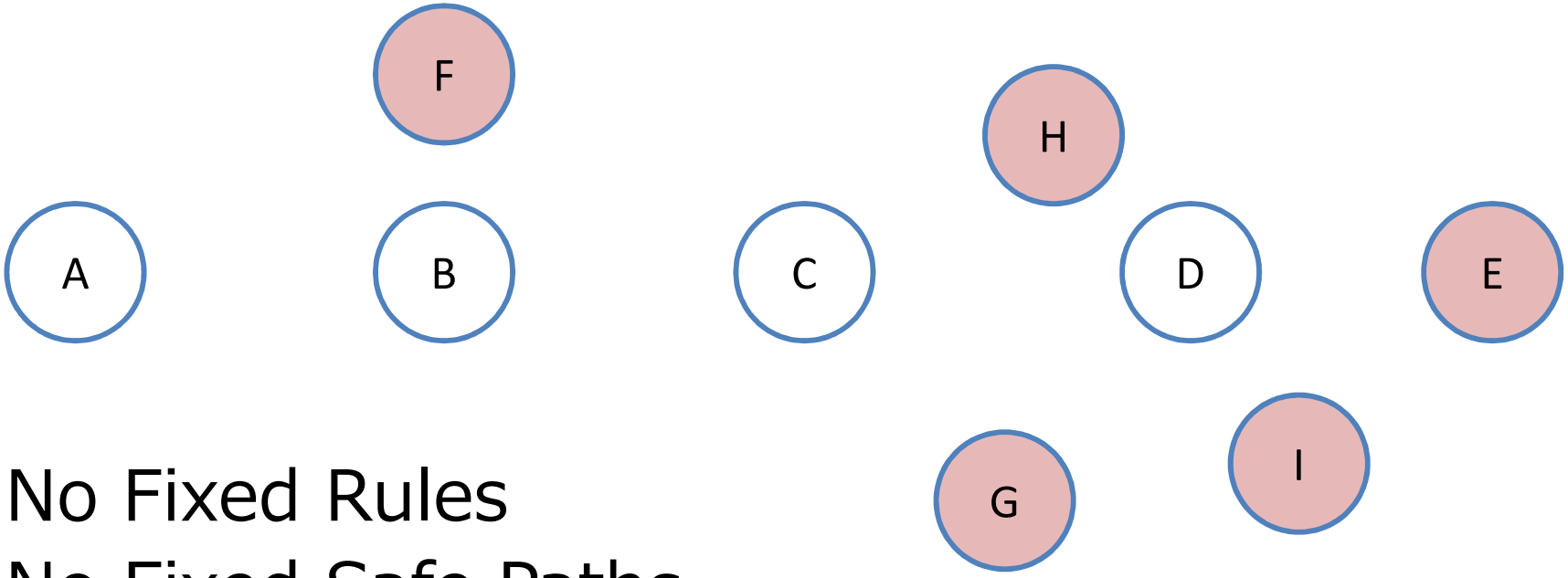
How A can succeed?



- No Fixed Rules
- **No Fixed Safe Paths**
- No Safety Actors

How to Survive in Shinjuku St

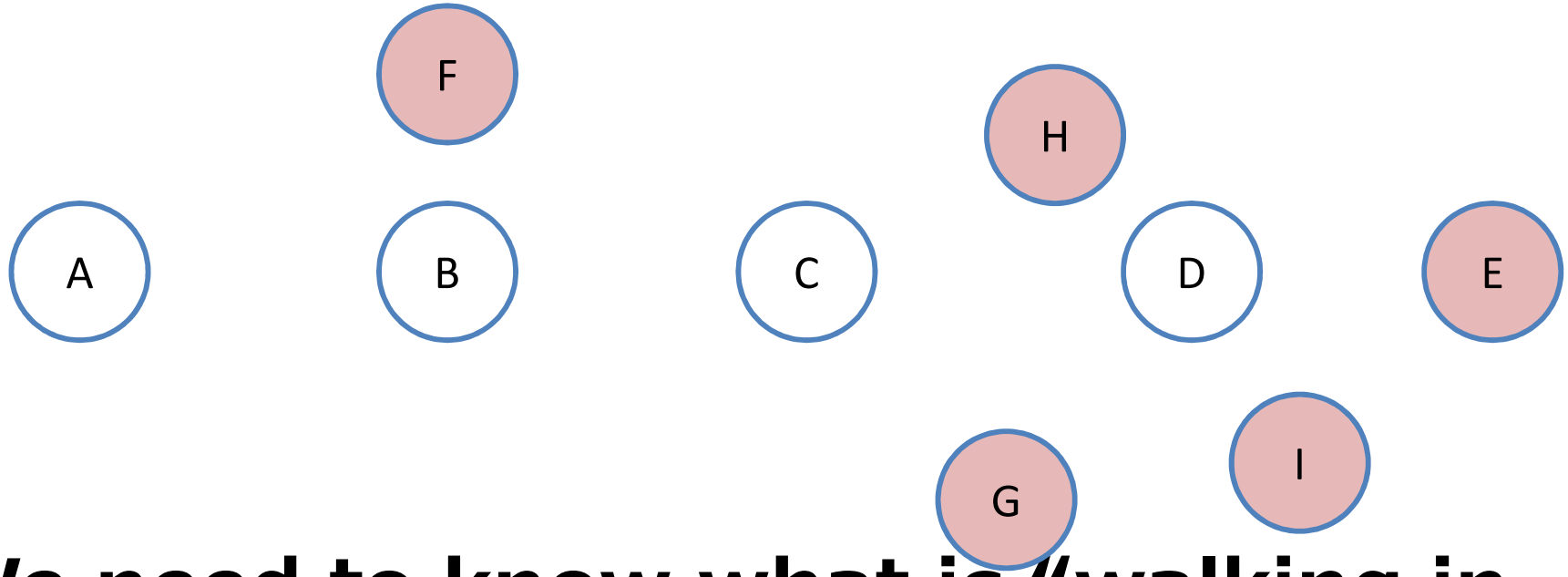
How A can succeed?



- No Fixed Rules
- No Fixed Safe Paths
- **No Fixed Actors**

How to Survive in Shinjuku St

It is very hard to define control structure of this kind of system.



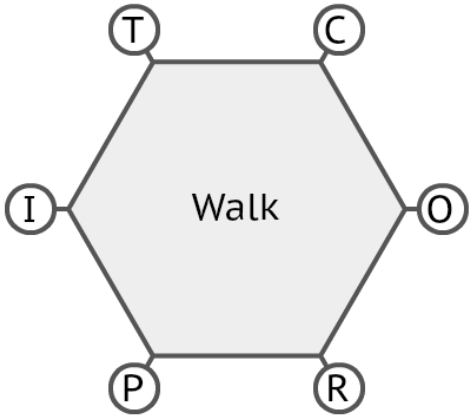
We need to know what is “walking in Shinjuku” to define that.

How to Survive in Shinjuku St

What is “walking in Shinjuku”?

「新宿駅で歩く」ことのProcess Model Variableは？

What are the Process Model Variables of “Walking in Shinjuku?”

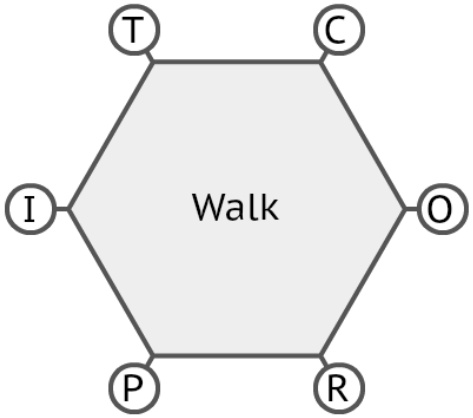


How to Survive in Shinjuku St

What is “walking in Shinjuku”?

「新宿駅で歩く」ことのProcess Model Variableは？

I	INPUT
P	PRECONDITION
R	RESOURCE
T	TIME
C	CONTROL
O	OUTPUT

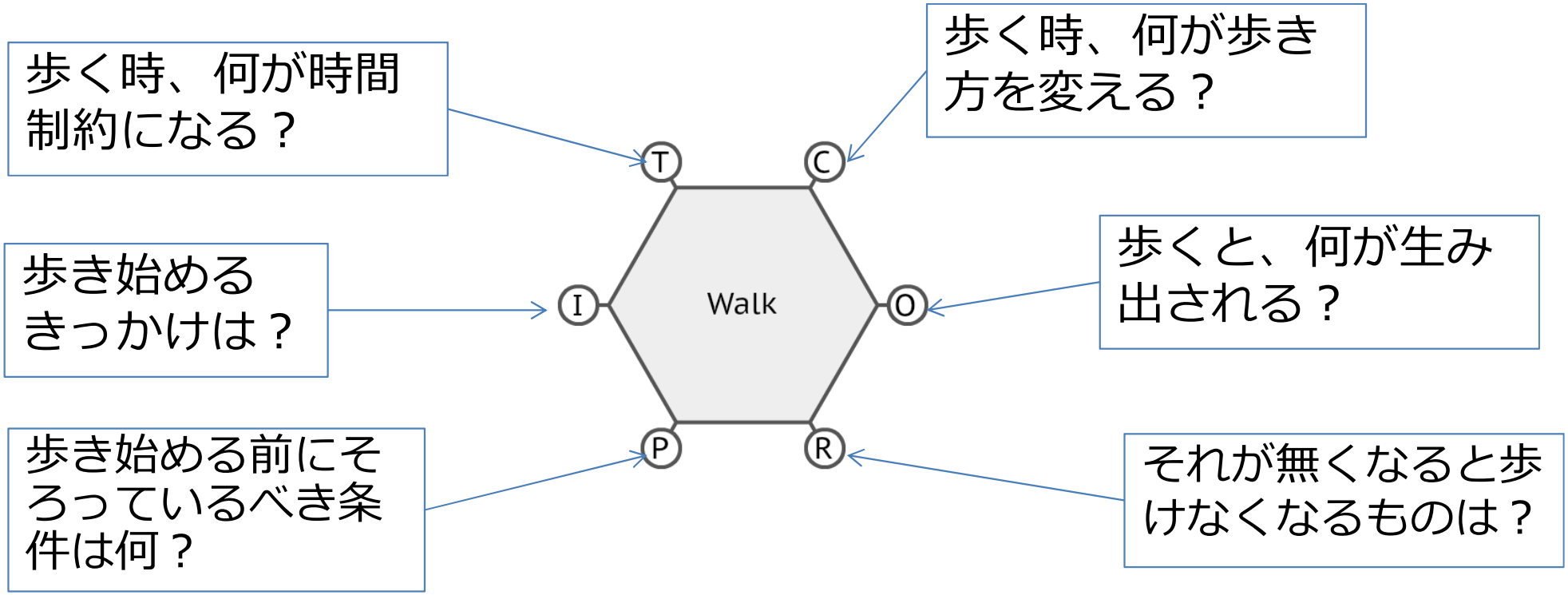


I	処理のトリガー
P	前提条件
R	資源
T	時間制約
C	制御
O	出力

How to Survive in Shinjuku St

What is “walking in Shinjuku”?

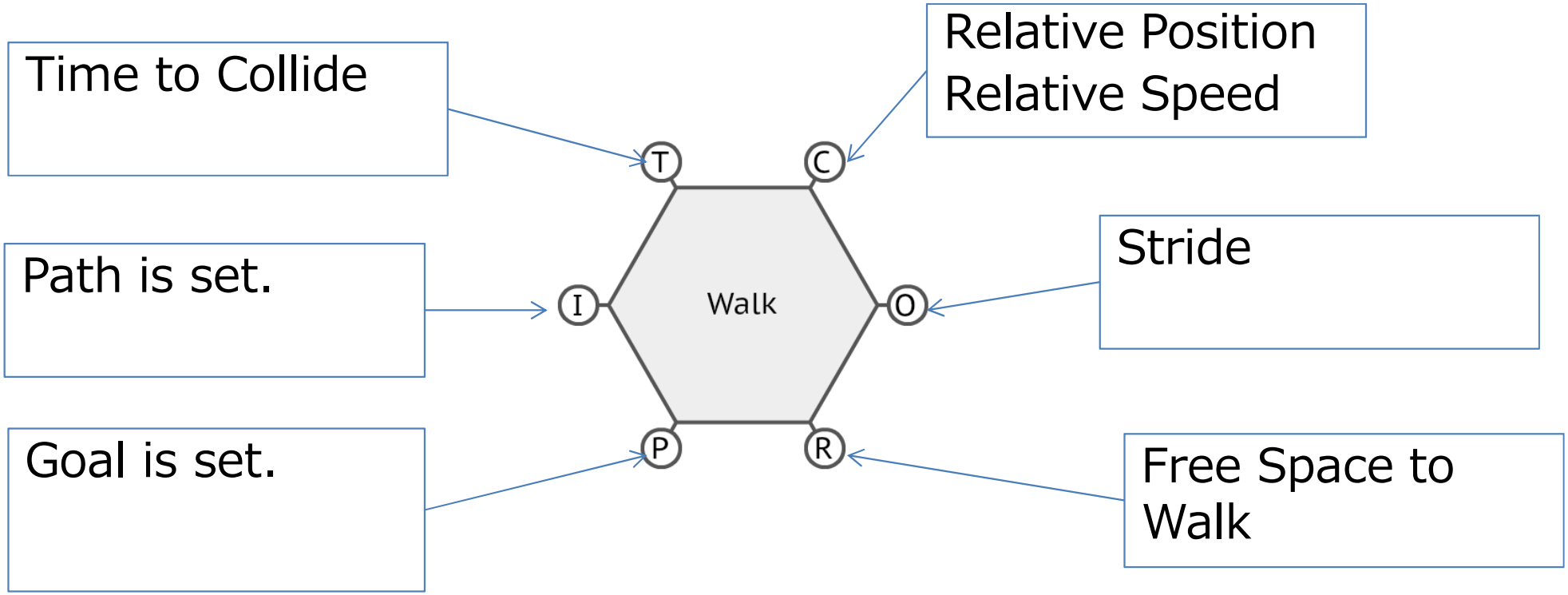
「新宿駅で歩く」ことのProcess Model Variableは？



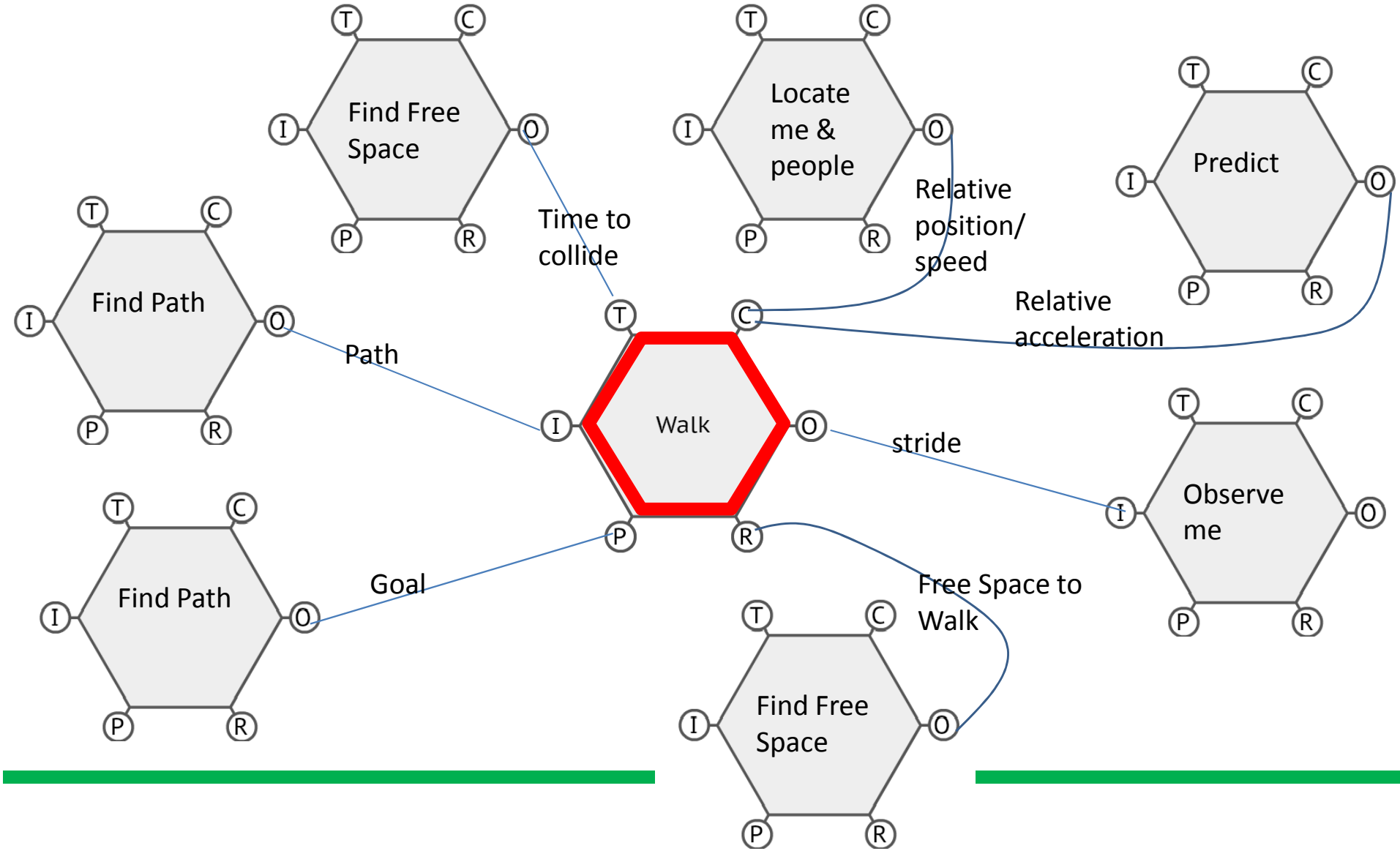
How to Survive in Shinjuku St

What is “walking in Shinjuku”?

「新宿駅で歩く」ことのProcess Model Variableは？

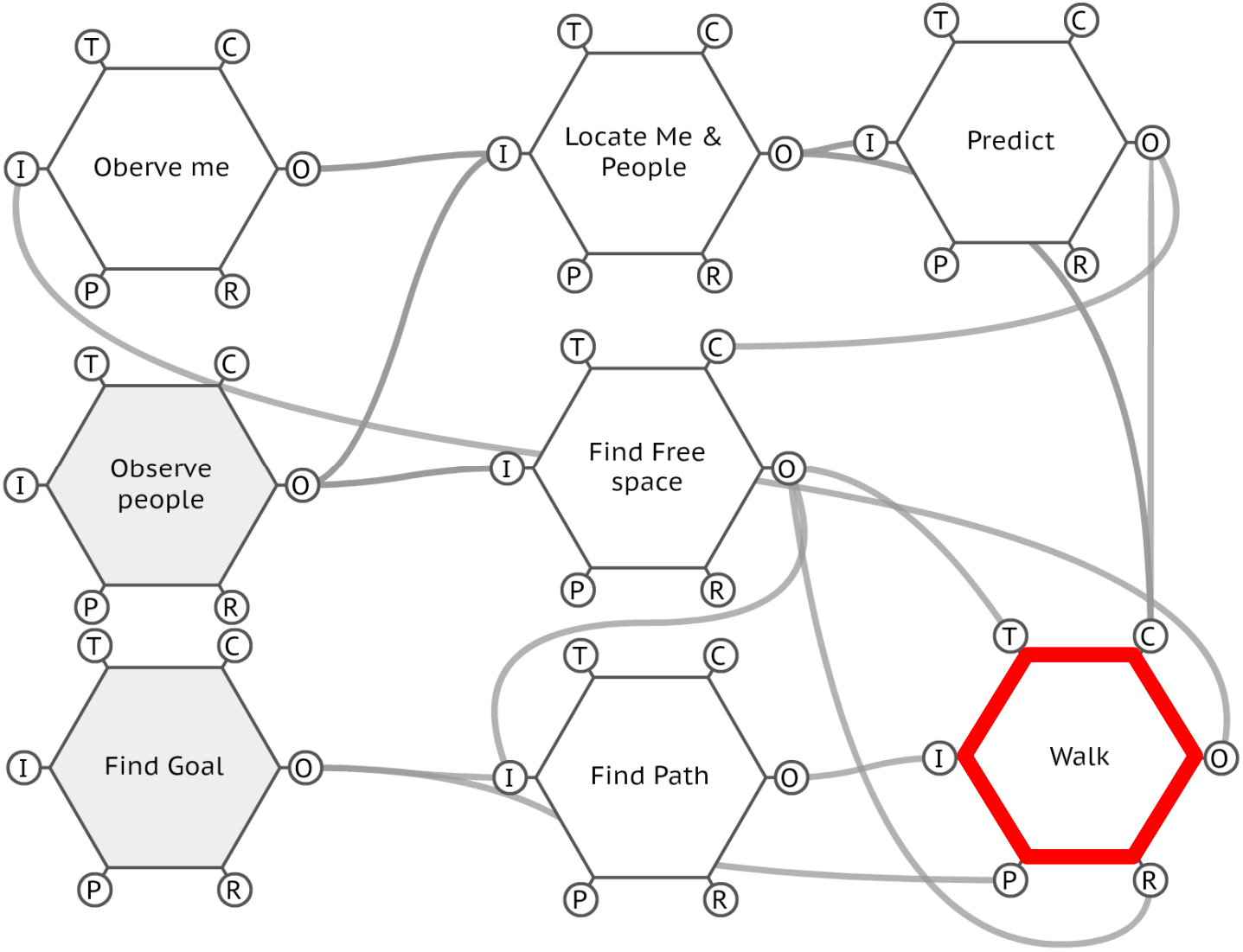


How to Survive in Shinjuku St



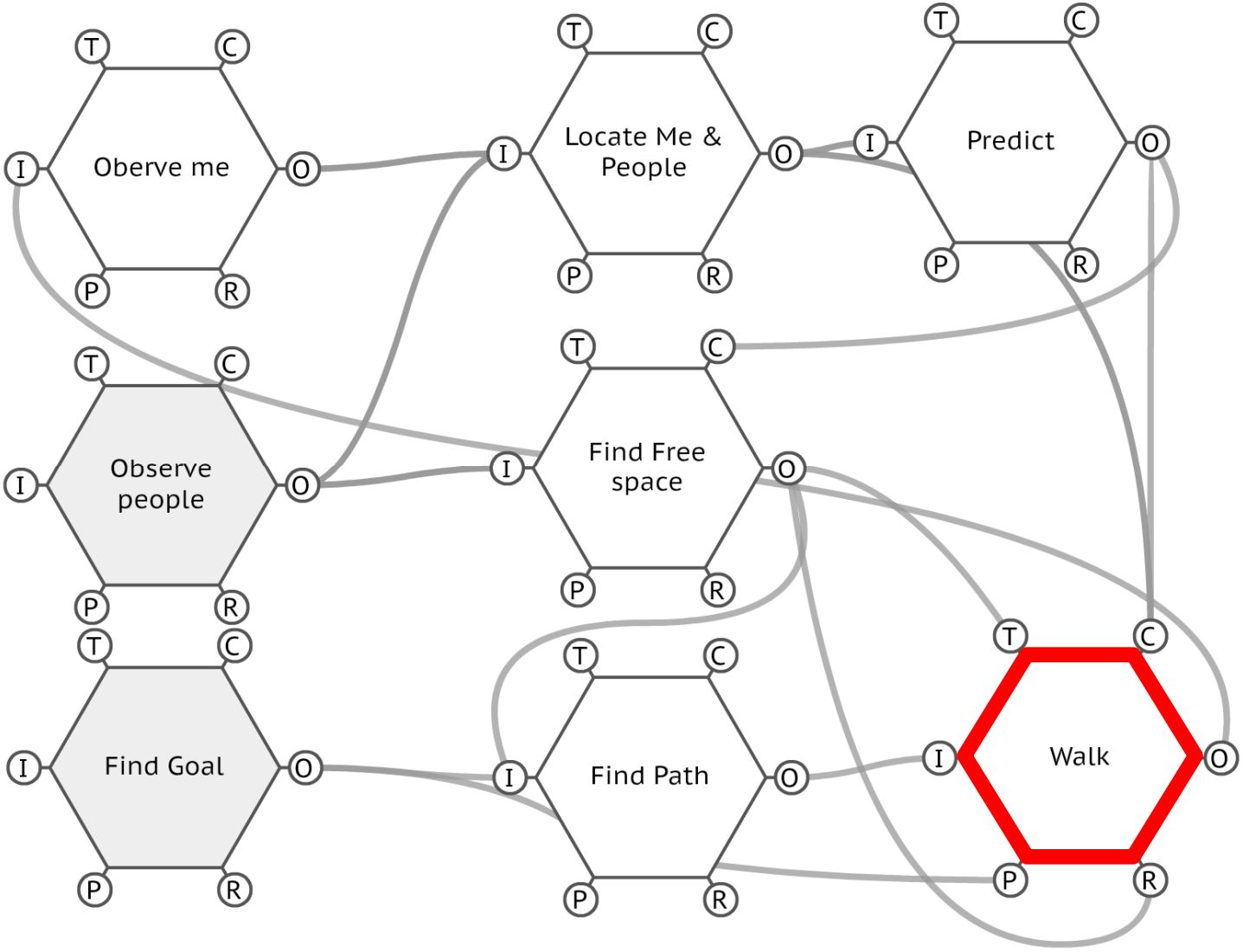
How to Survive in Shinjuku St

機能間の依存関係と前後関係でシステムをModeling



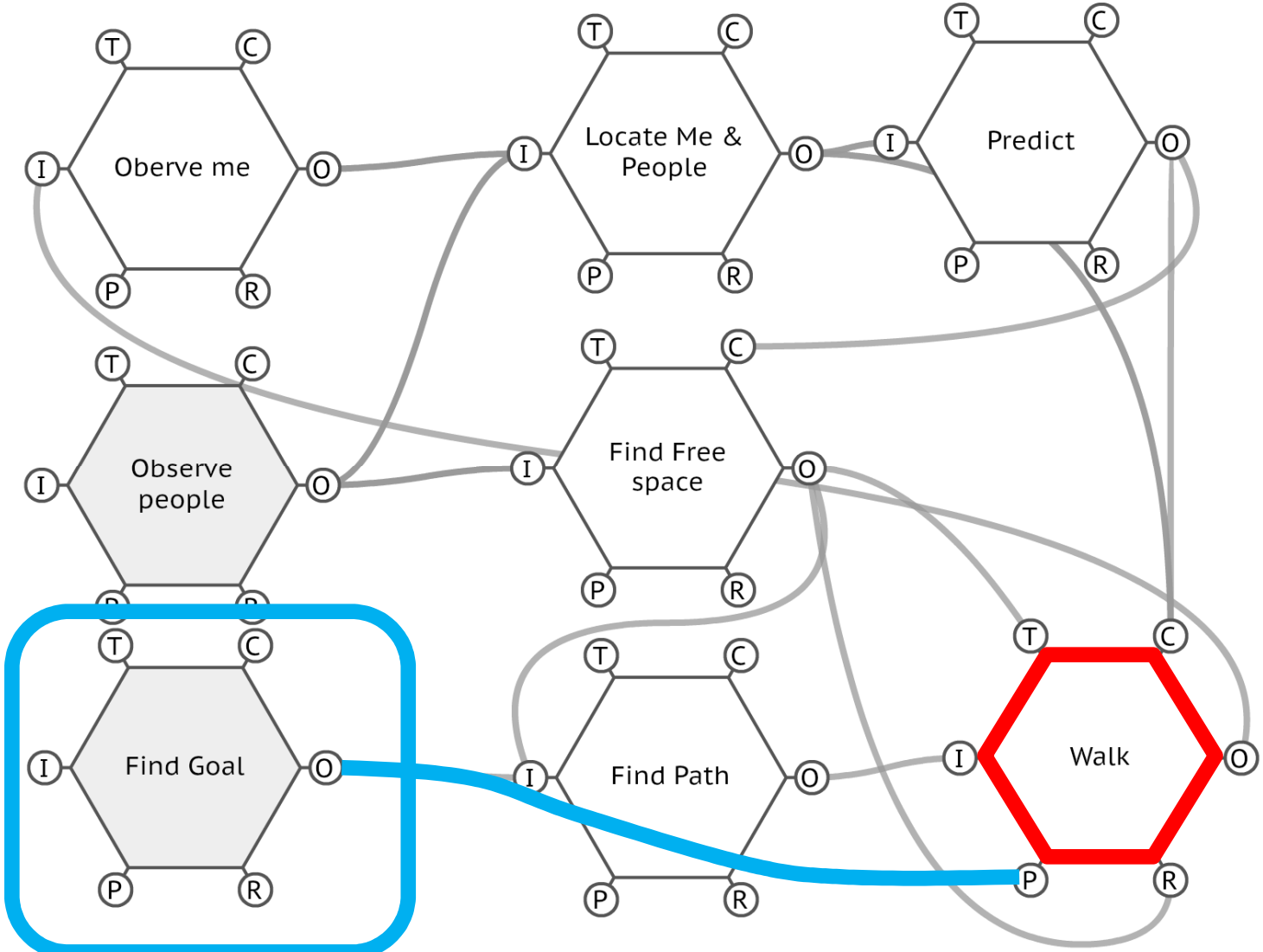
How to Survive in Shinjuku St

機能間の依存関係と前後関係でシステムをModeling



How to Survive in Shinjuku St

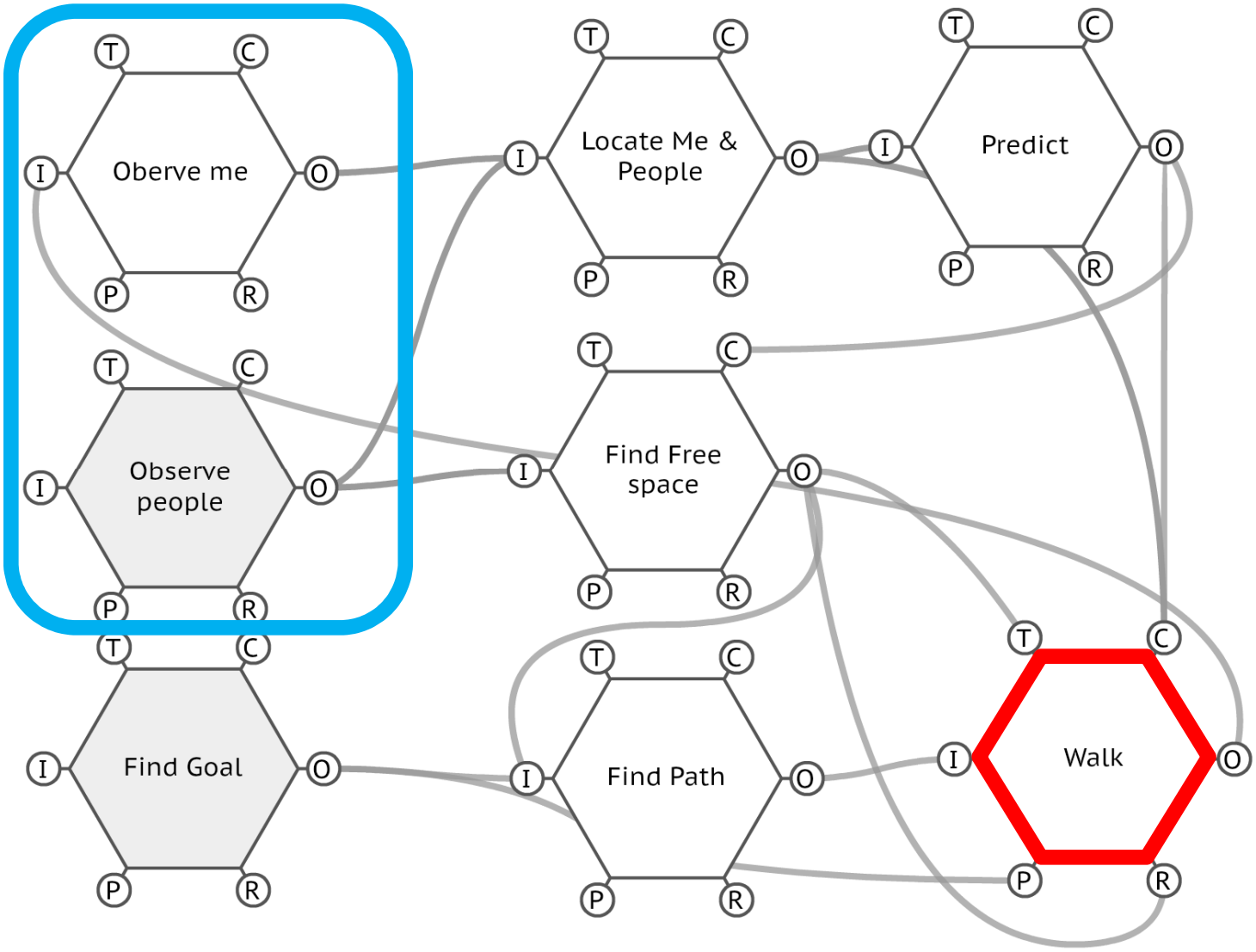
機能間の依存関係と前後関係でシステムをModeling



Find Goal
目的地を決める

How to Survive in Shinjuku St

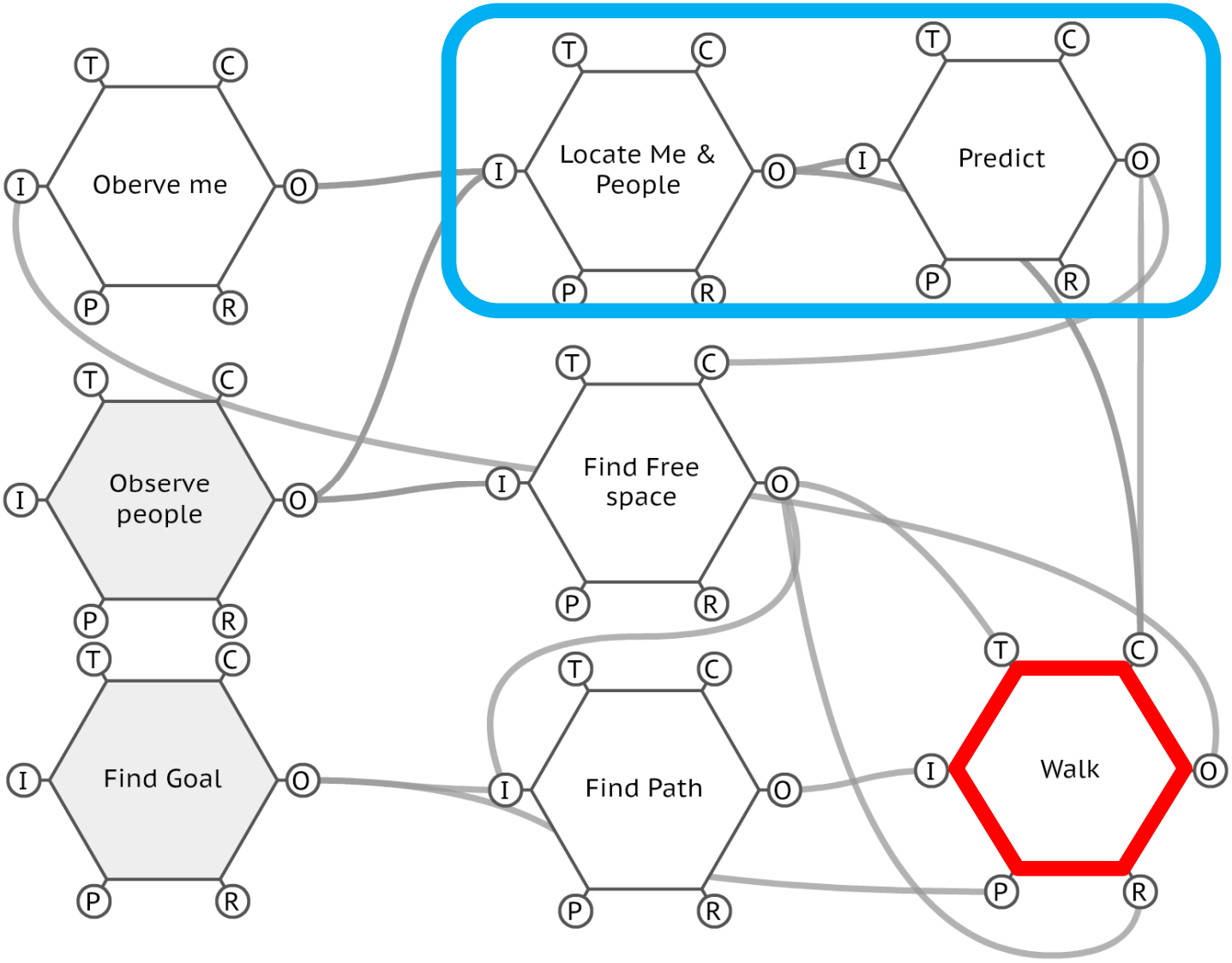
機能間の依存関係と前後関係でシステムをModeling



Get Information
観察する

How to Survive in Shinjuku St

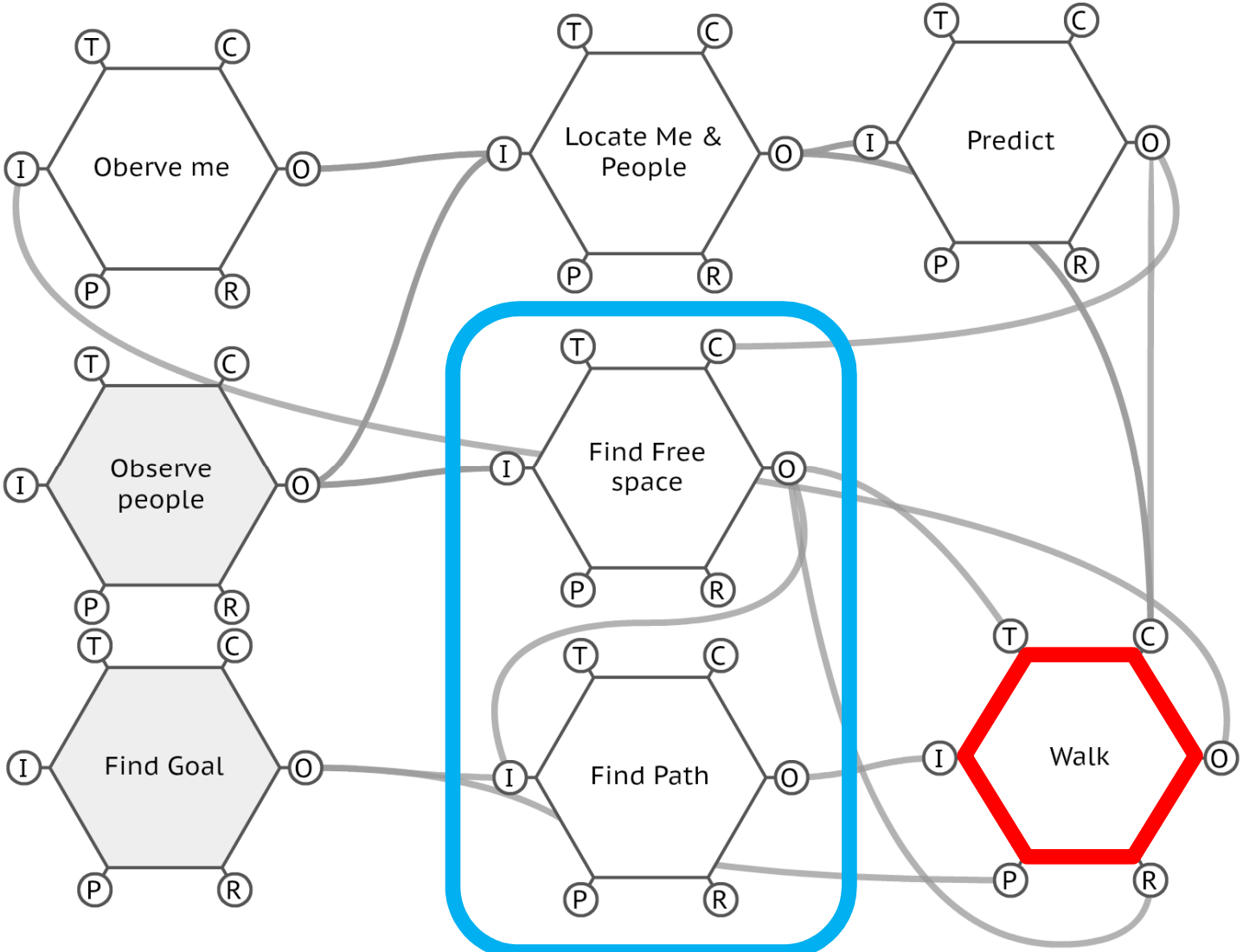
機能間の依存関係と前後関係でシステムをModeling



Predict Future
将来予測

How to Survive in Shinjuku St

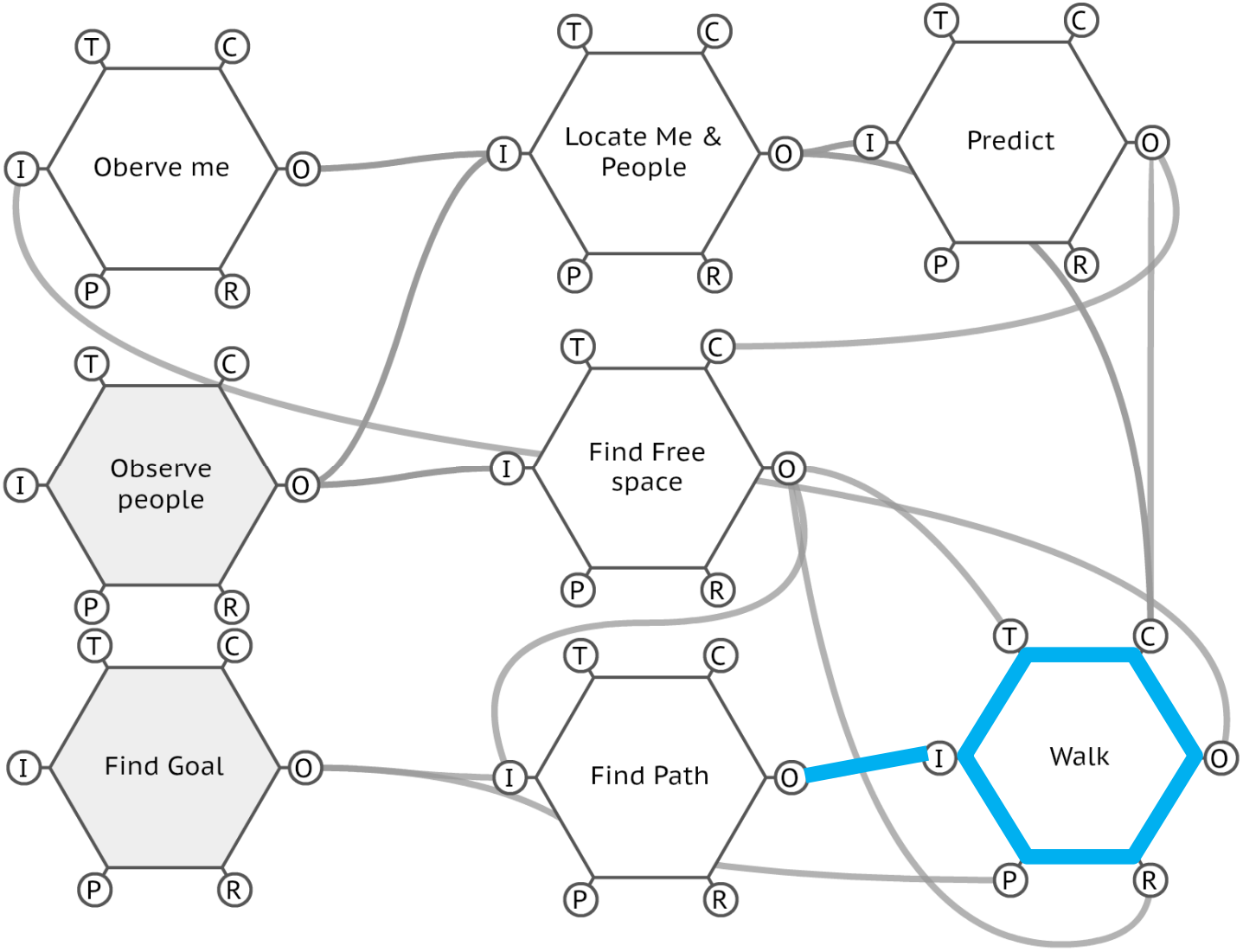
機能間の依存関係と前後関係でシステムをModeling



Find Path
方針の決定

How to Survive in Shinjuku St

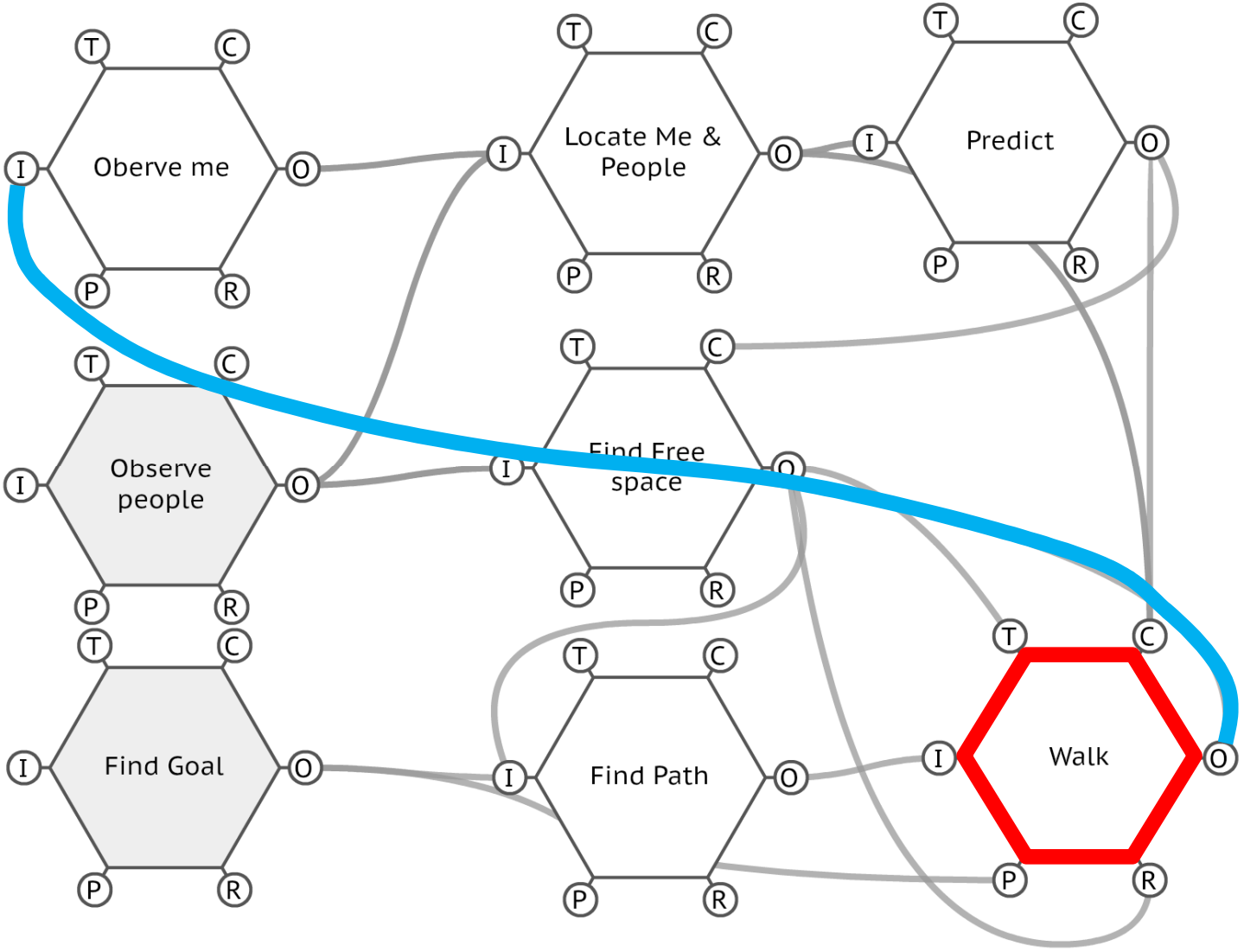
機能間の依存関係と前後関係でシステムをModeling



Walk
歩く

How to Survive in Shinjuku St

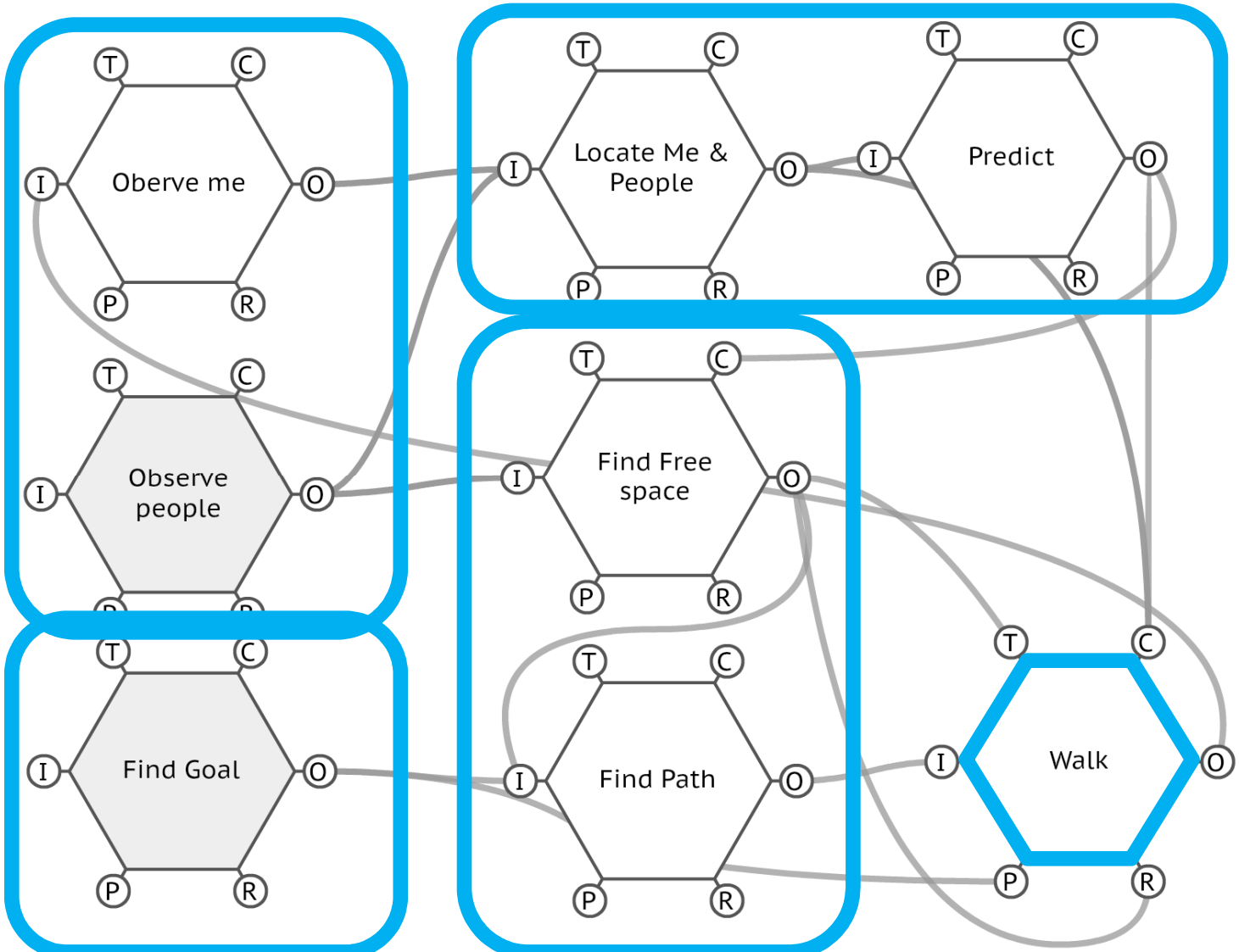
機能間の依存関係と前後関係でシステムをModeling



Feedback
次のサイクルへ

How to Survive in Shinjuku St

機能間の依存関係と前後関係でシステムをModeling



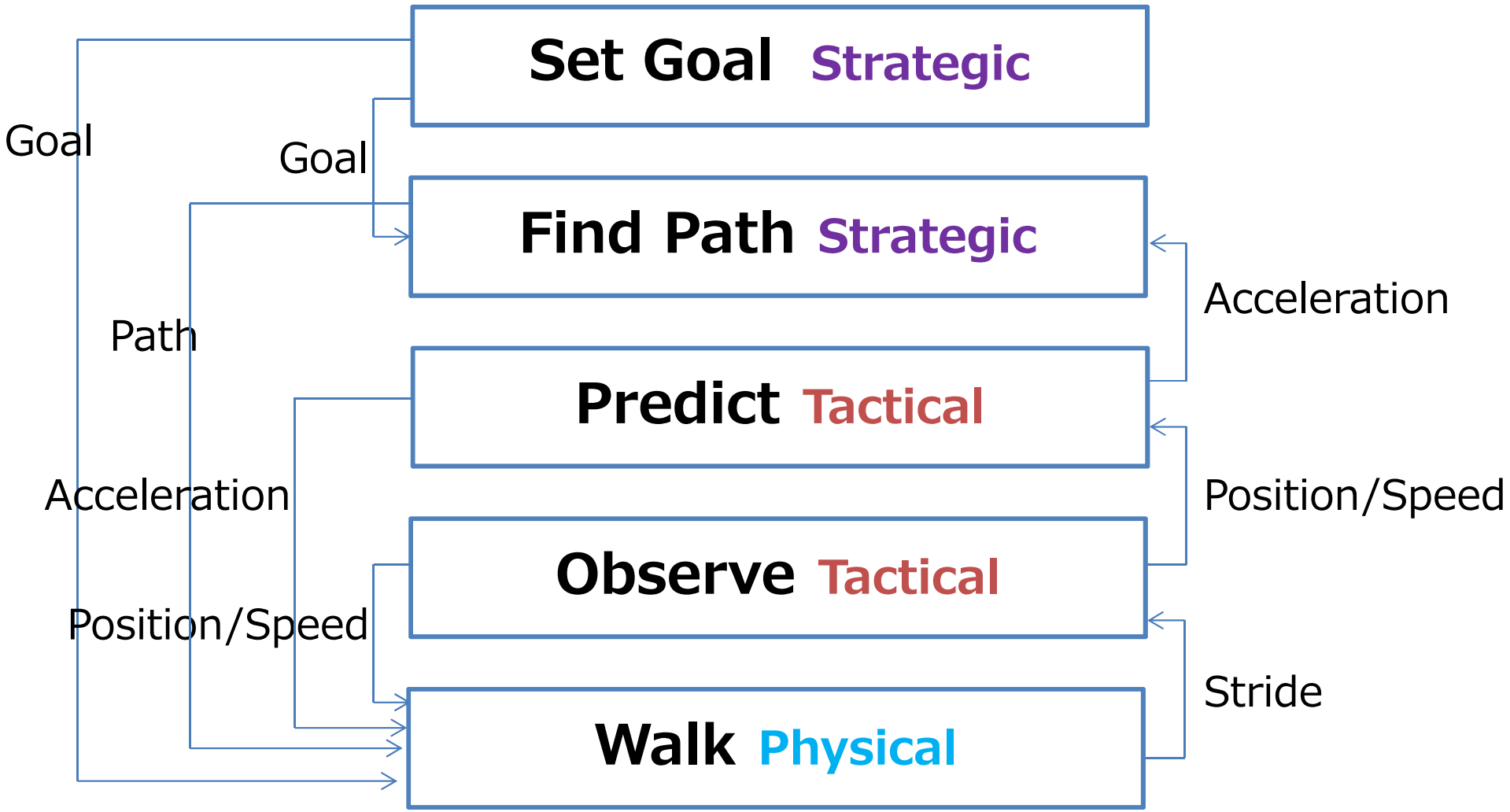
階層構造(Layered structure)のようなものが存在している。

How to Survive in Shinjuku St

見えてきた階層構造にそって、STAMPのcontrol structureにしてみる。（FRAMモデルをそのままSTAMPモデルに射影してみる）

Let’s do projection-mapping from FRAM to STAMP control structure.

How to Survive in Shinjuku St

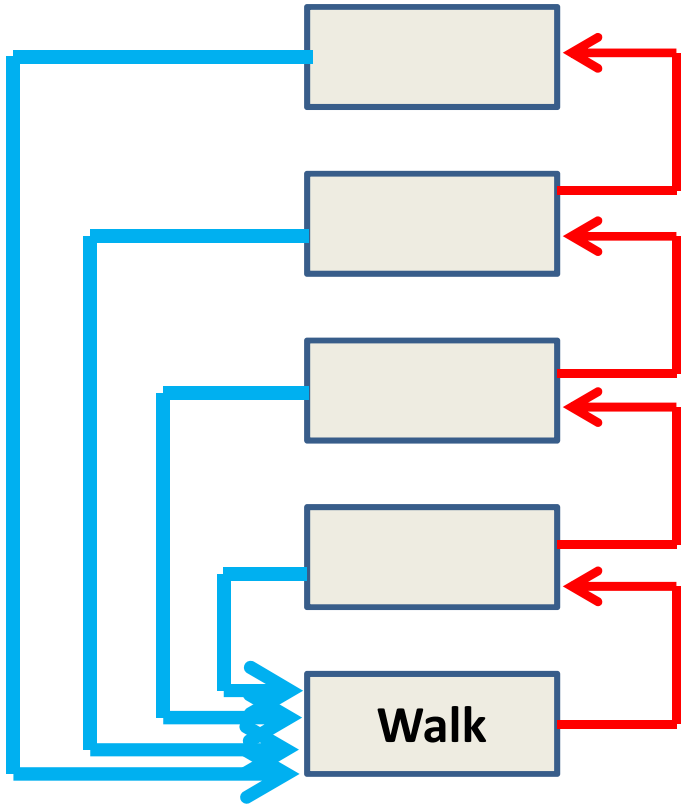


Nicely layered. But not very much systemic.

How to Survive in Shinjuku St

Less Top-down structure.

More Bottom-up structure.



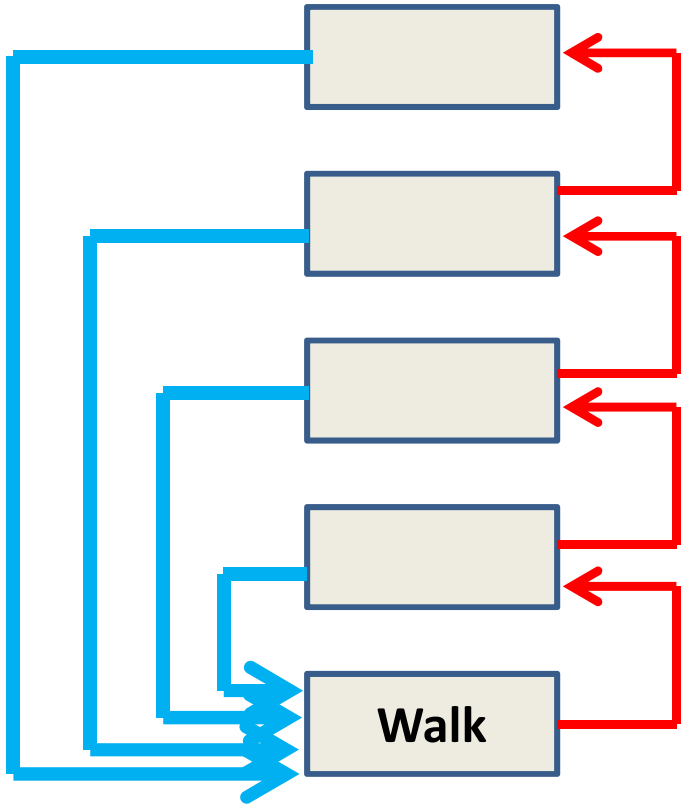
新宿駅での歩行は、かなりボトムアップなプロセス

大きな戦略の元に戦術を立てて実行するという計画性はなく、動物の群れが戦略を立てなくても全体システムとして動けることと似ている。

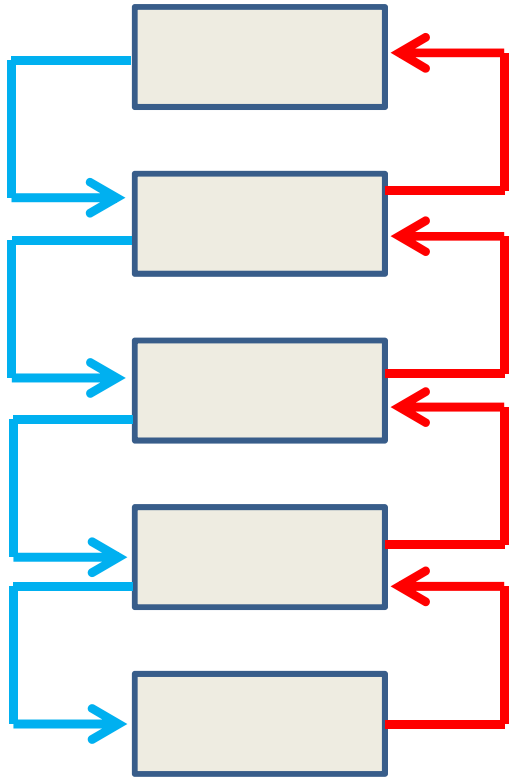
How to Survive in Shinjuku St

Less Top-down structure.

More Bottom-up structure.



Walk in Shinjuku



Typical Control Structure

How to Survive in Shinjuku St

新宿駅での歩行は、かなりボトムアップなプロセス

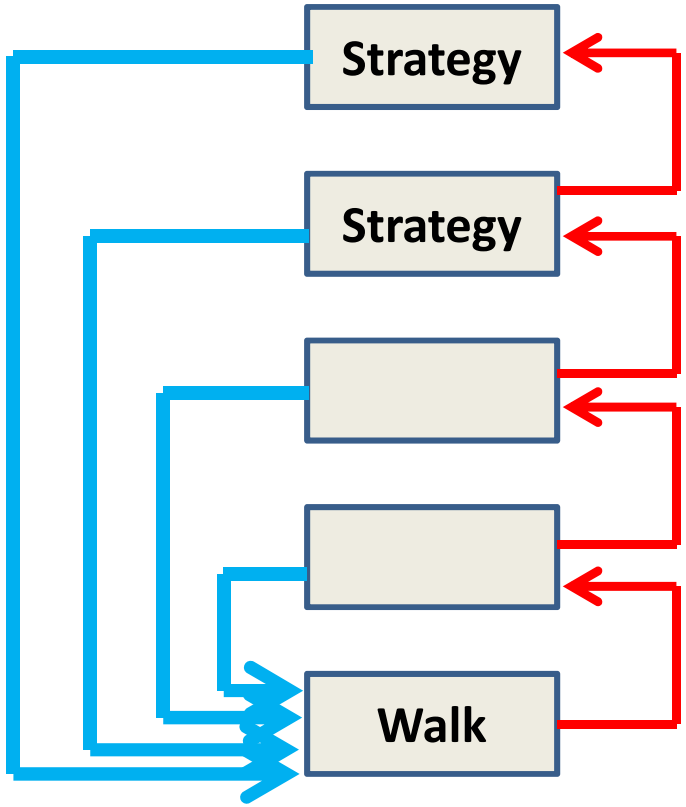
大きな戦略の元に戦術を立てて実行するという計画性はなく、動物の群れが戦略を立てなくても全体システムとして動けることと似ている。

Beautifully organized flock of fish.
Organized. But no Leader or Top down hierarchy.
The emergent formation from bottom-up process.

How to Survive in Shinjuku St

Why people can succeed in Shinjuku?

Everybody walks by the strategy emerged in bottom-up manner.
The strategy can be adjusted to changing situations because it is built bottom-up.



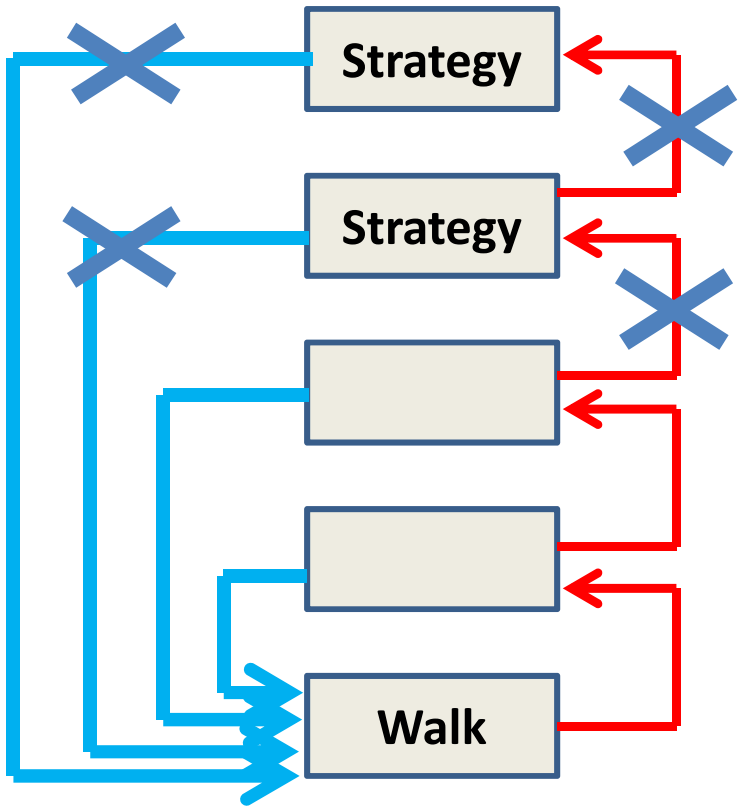
新宿駅コンコースの成功要因：

全ての歩行者が、誰の指示にもよらず、同じ**戦略**をつかって、**Walk**する。
トップダウンで戦略を立てず、ボトムアップに立てるので、状況に応じて戦略が少しずつ自然に変化できる。
レジリエント。

How to Survive in Shinjuku St

Why people can succeed in Shinjuku?

Everybody walks by the strategy emerged in bottom-up manner.
The strategy can be adjusted to changing situations because it is built bottom-up.



リスク要因：
途中のデータが喪失すると、それより上位の機能が全て喪失する。
戦略無しで制御が継続

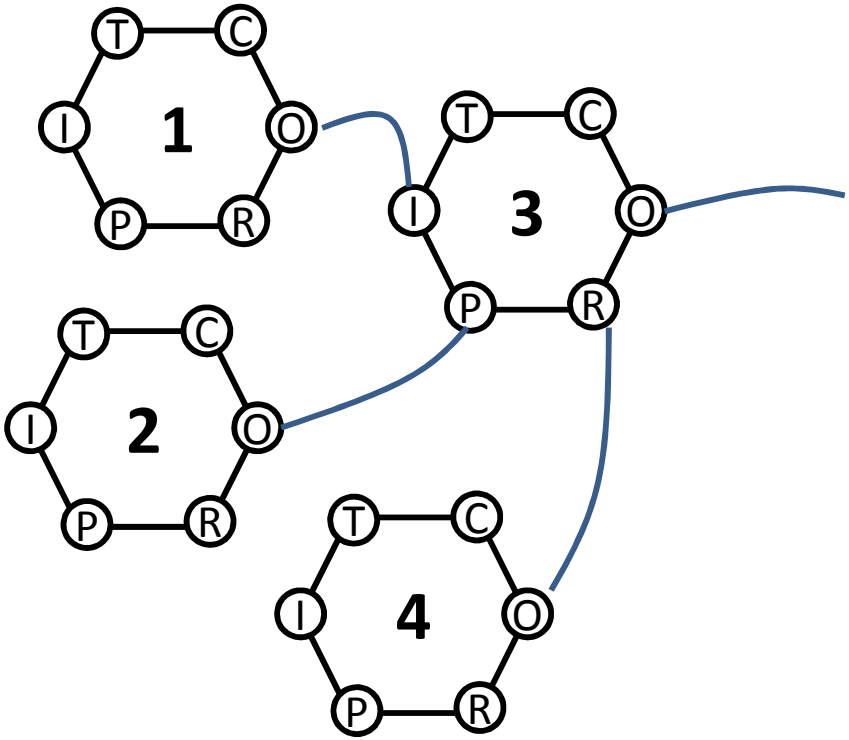
Risk Factor:
Loss of a feedback will loose all upper layer functions.
Keep operating without strategy.

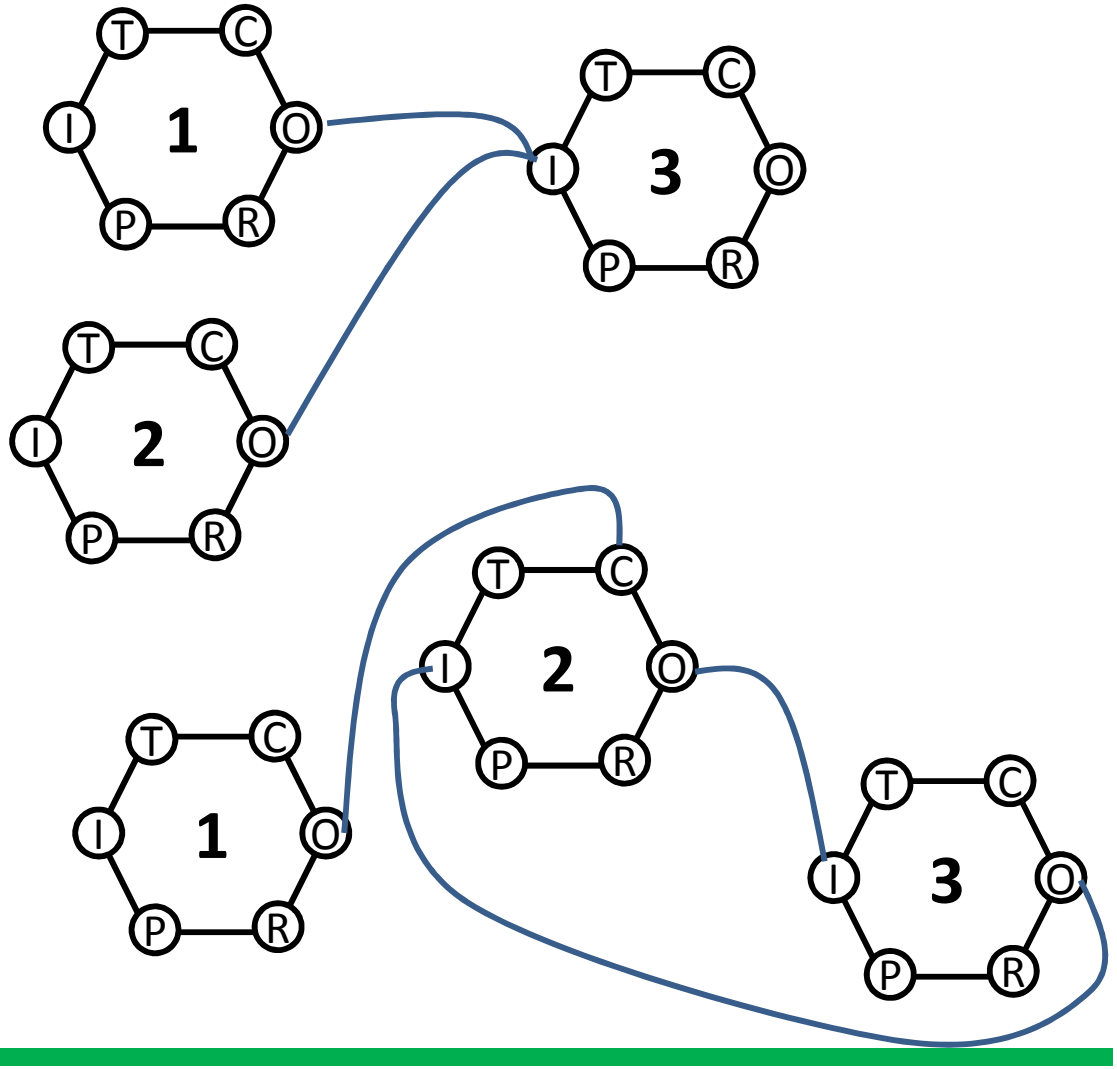
まとめ

- STAMPとFRAMのHybridを試した。
- FRAMでSuccess StoryをModelingし、STAMPのControl Structure DiagramにProjection Mappingした。
- その結果、面白いBottom-up strategic group management というarchitectureが見えてきた。
- 何故鳥の群れや魚の群れが、あれほどきれいに一団となって同時に方向転換できるのか？
- The model might shows why flock of birds or fish can maneuver beautifully organized without strategic planning.

Back up slides

何故成功要因の分析にFRAMが有効？

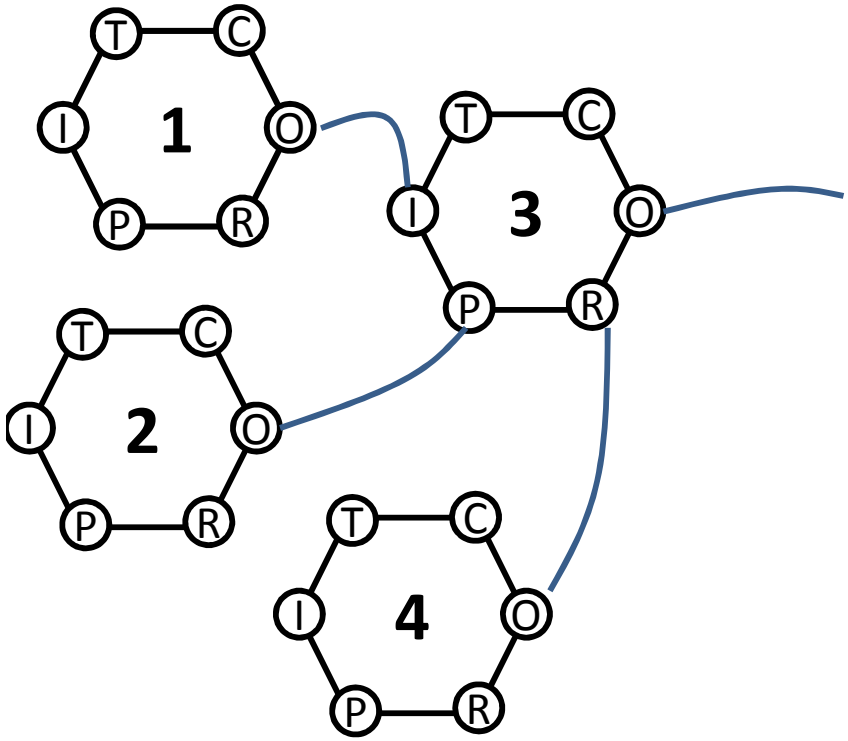




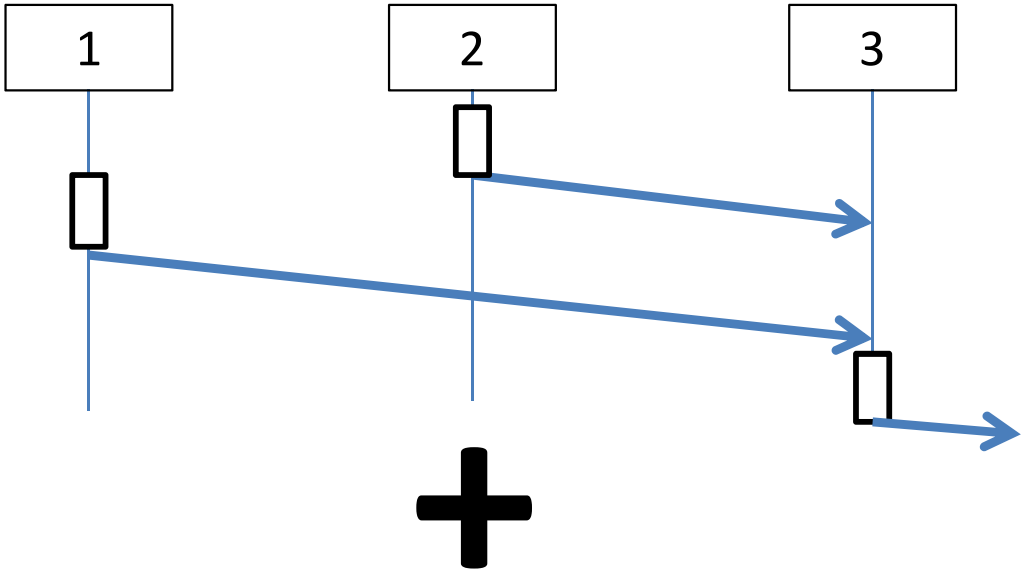
何故成功要因の分析にFRAMが有効？

FRAMモデルは見た目より情報量が多い

入力シーケンスと依存関係を同時に表現



=

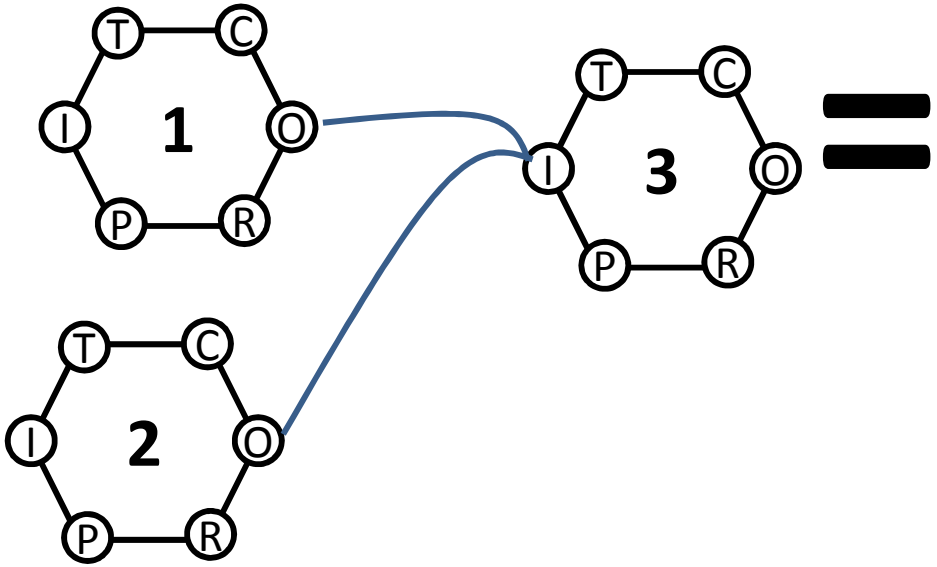


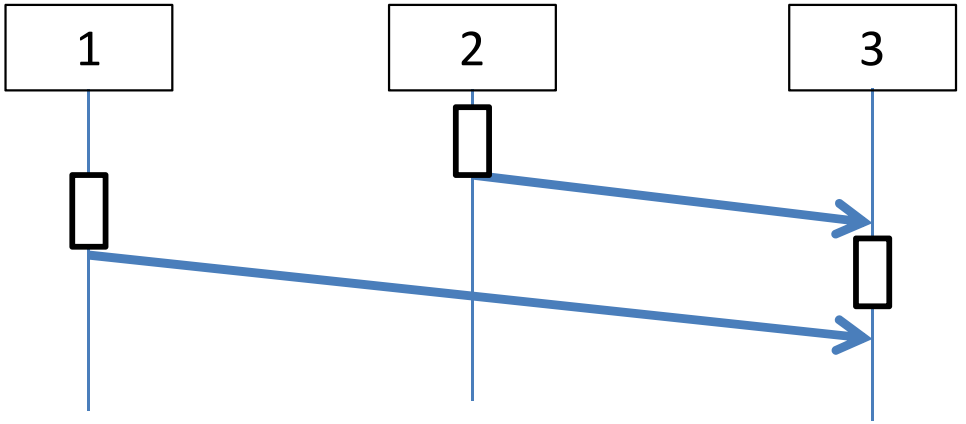
Use Case	3の実行
Goal	3が出力を行う
Pre cnd	2が出力を行っていること
Flow	1入力後、2を検査し、3を実行
Post cnd	4のリソースが足りていること

何故成功要因の分析にFRAMが有効？

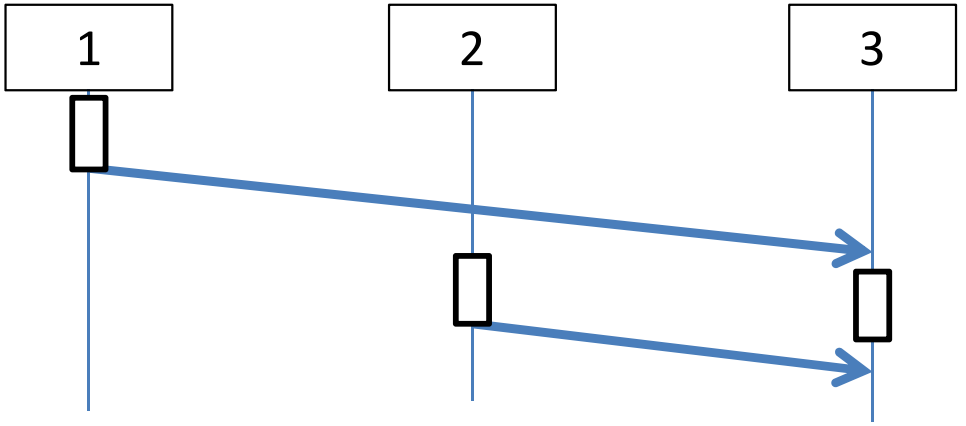
非決定論定期な関係をシンプルに表現

「1と2はどちらが先に来るか不明」という状態を表現できる（明らかなリスク）





OR

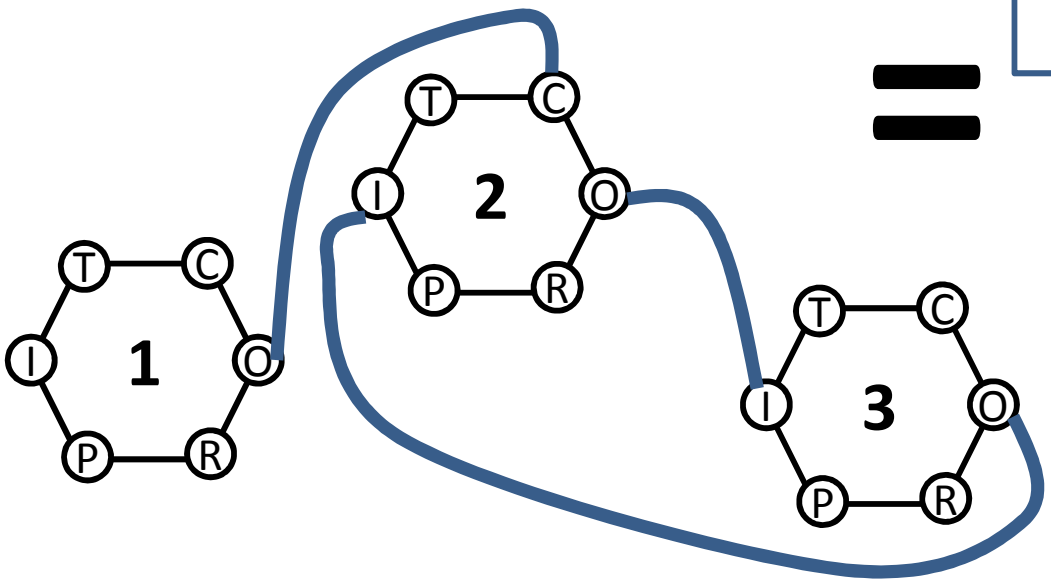


何故成功要因の分析にFRAMが有効？

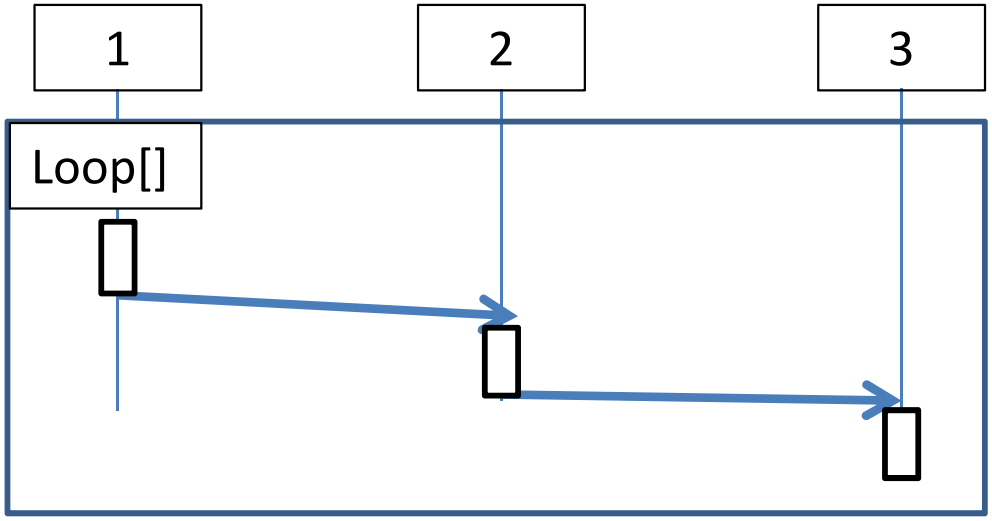
時間経過による変化を表現

2と3のループ中に、1が2を変化させ、

エスカレーションを発生させる



=



+

Const1

2

<<<constraints>>>

Const1

Constraints

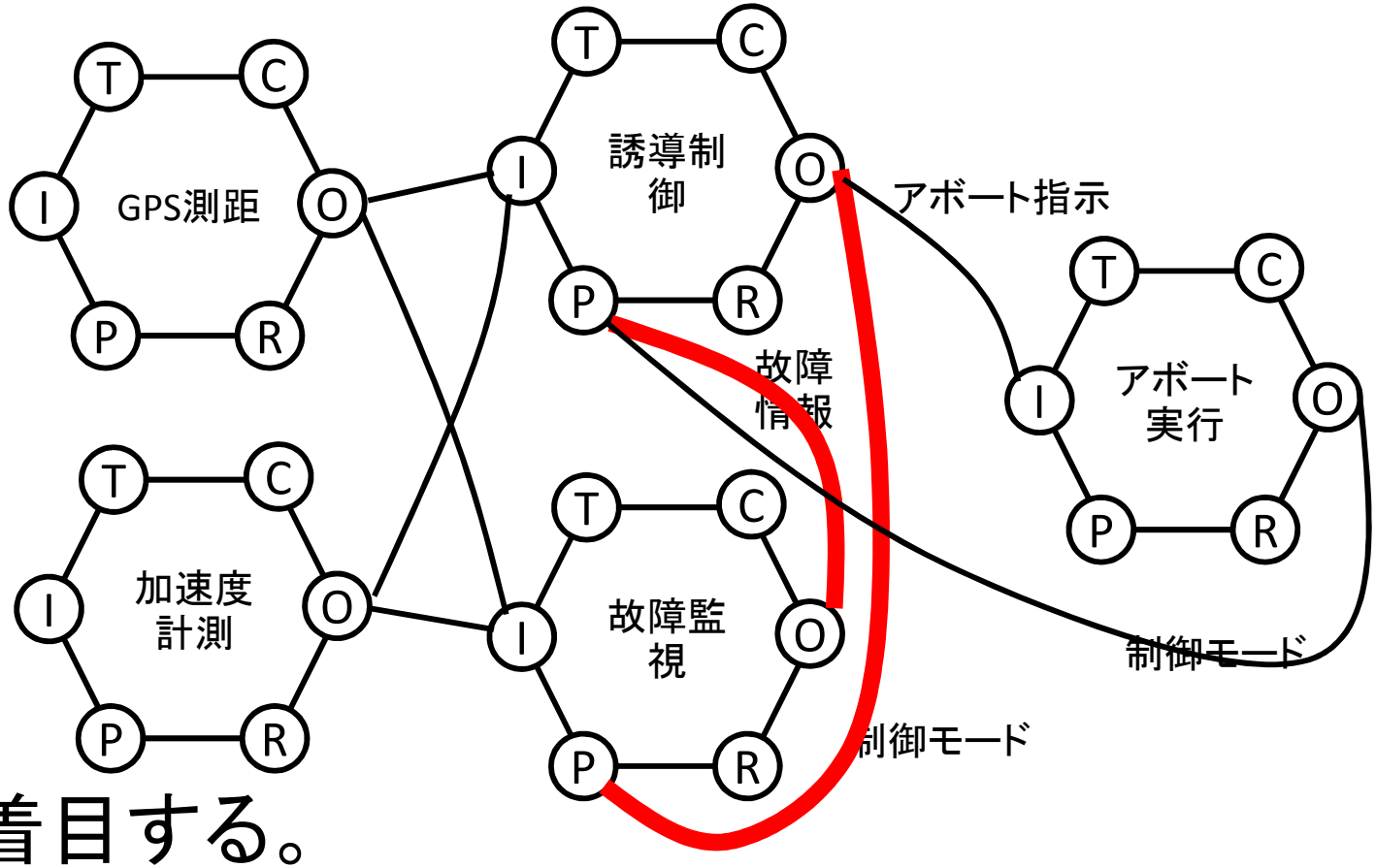
Parameters

Output_from_1

成功要因からリスクを見出す

FRAM分析とは:
共鳴ポイント
の分析

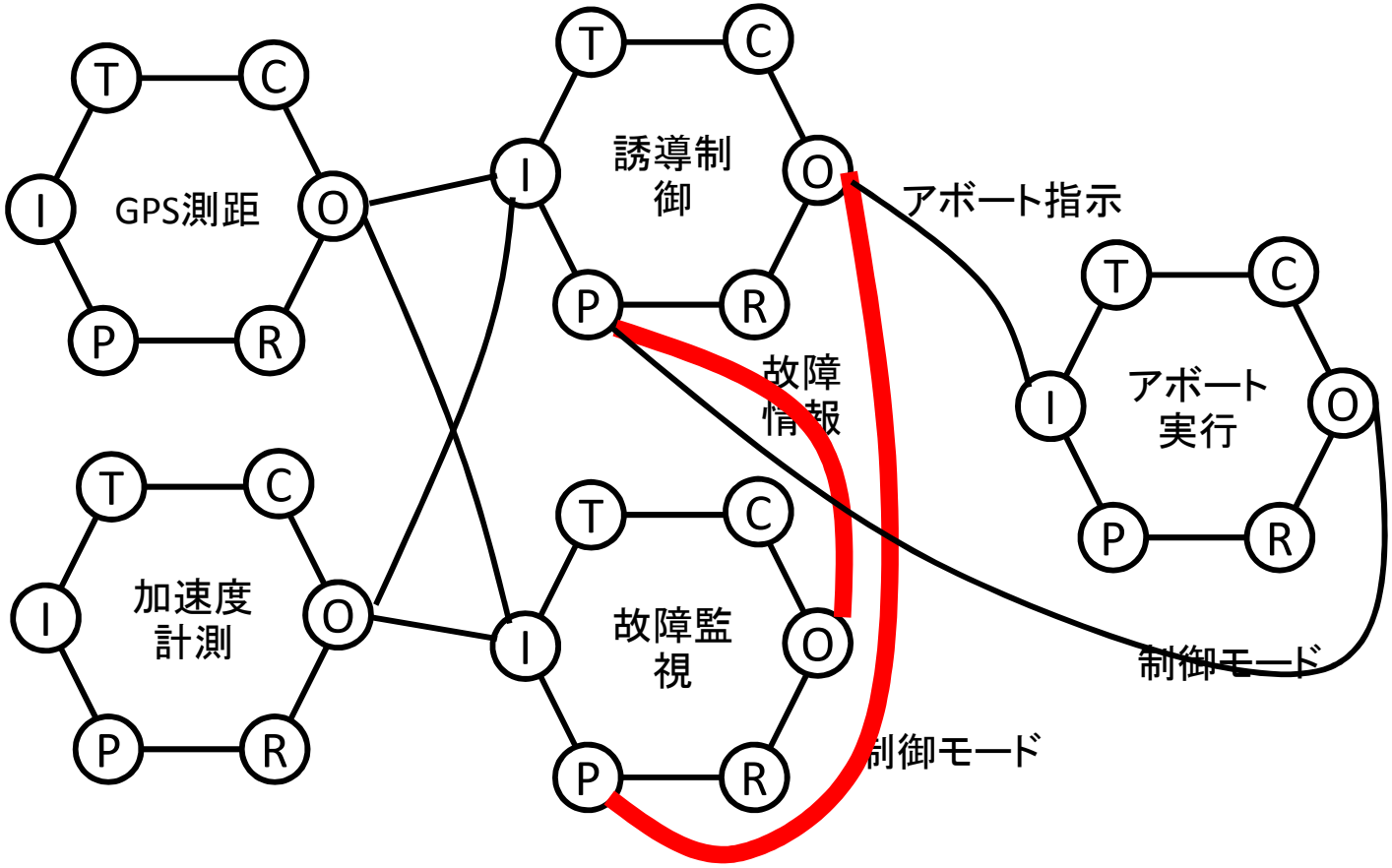
共鳴現象は、
ループ構造や
入出力の集中
箇所に発生。
FRAM分析では
共鳴ポイントに着目する。



成功要因からリスクを見出す

共鳴ポイント:
変わった
ループ構造

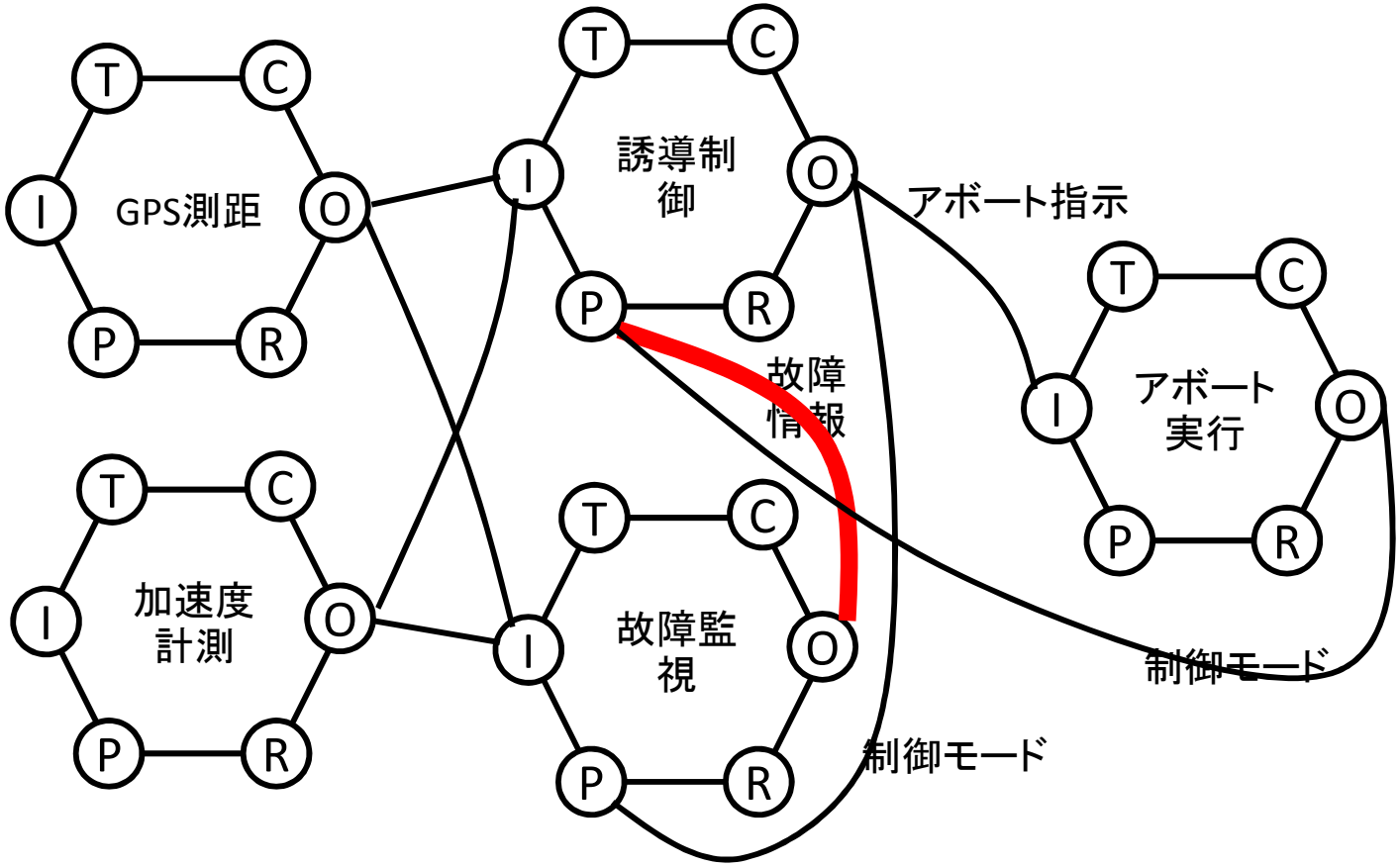
誘導制御と
故障監視が
互いに前提
条件を提供
（互いに依存
しあう関係）



成功要因からリスクを見出す

共鳴ポイント:
変わった
ループ構造

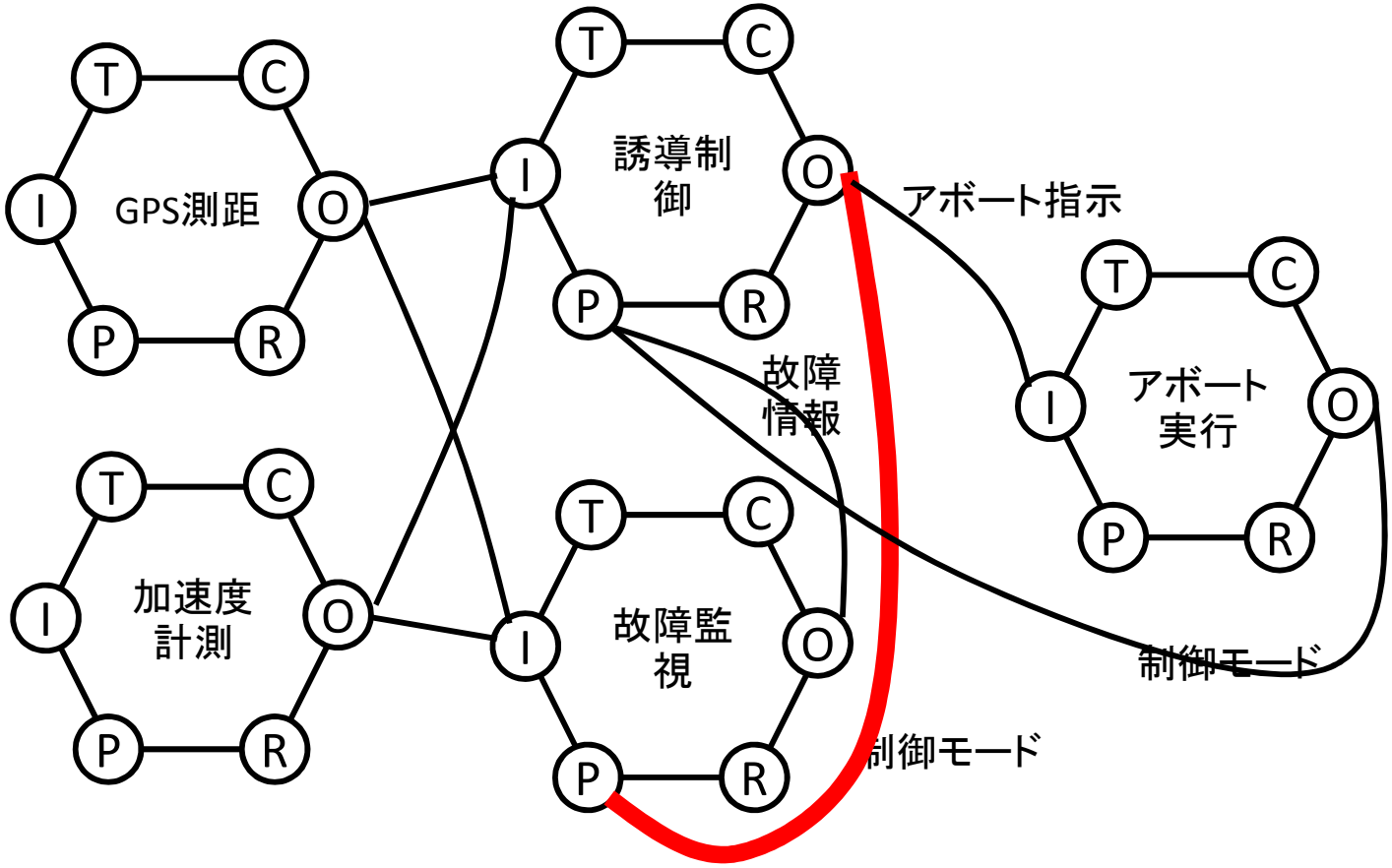
誘導制御は
故障していない
センサのみ
使いたい



成功要因からリスクを見出す

共鳴ポイント:
変わった
ループ構造

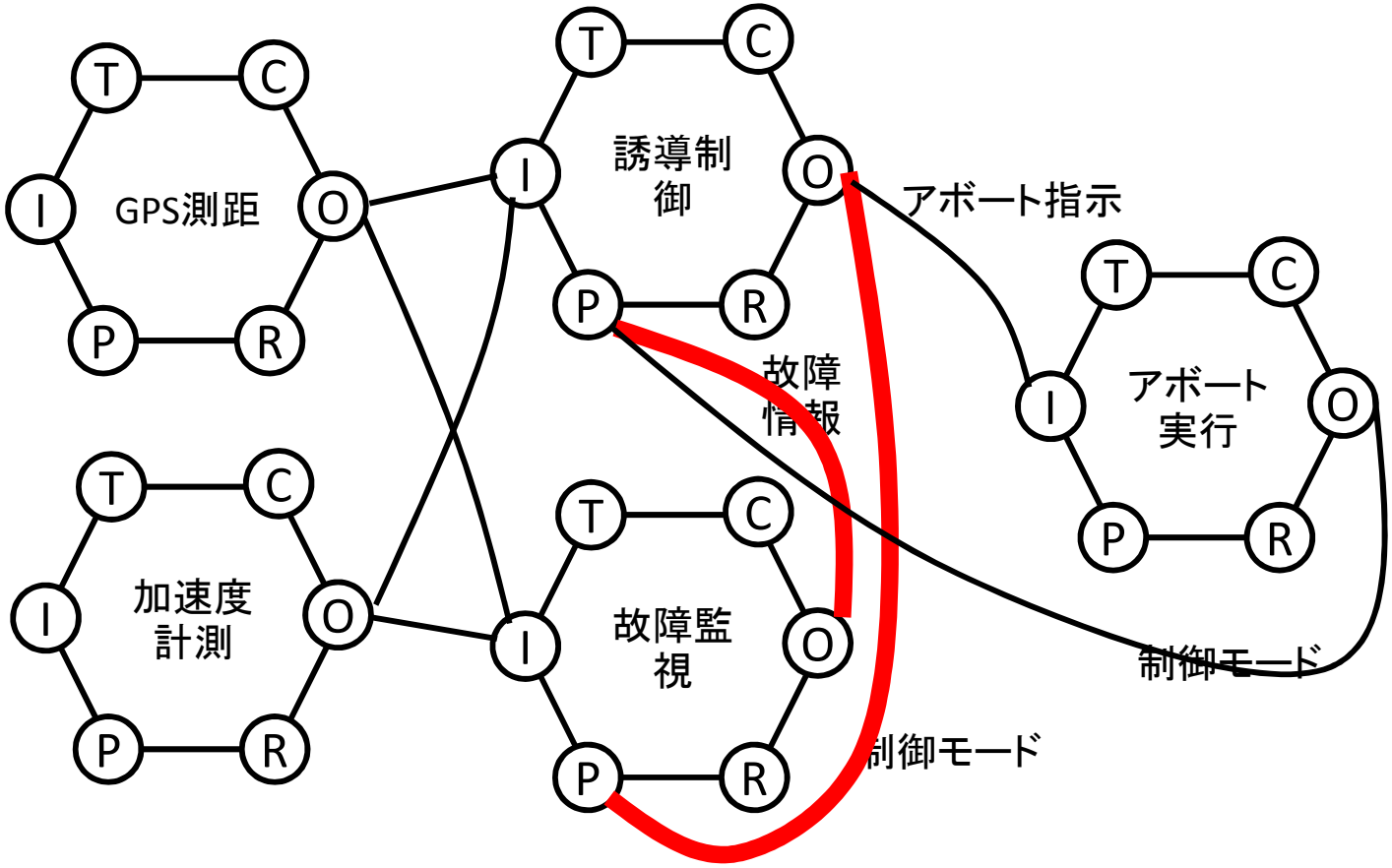
故障監視は
特定の制御
モード時のみ
故障監視したい



成功要因からリスクを見出す

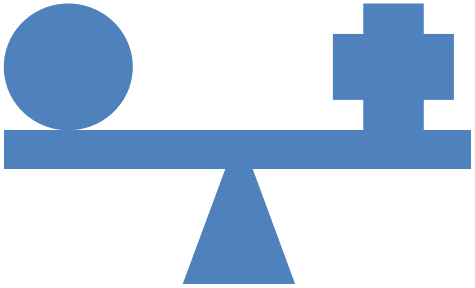
互いに制約し
あうことにより、
監視すべき時に
監視し、
誘導すべき時に
誘導できる。

これが成功要因

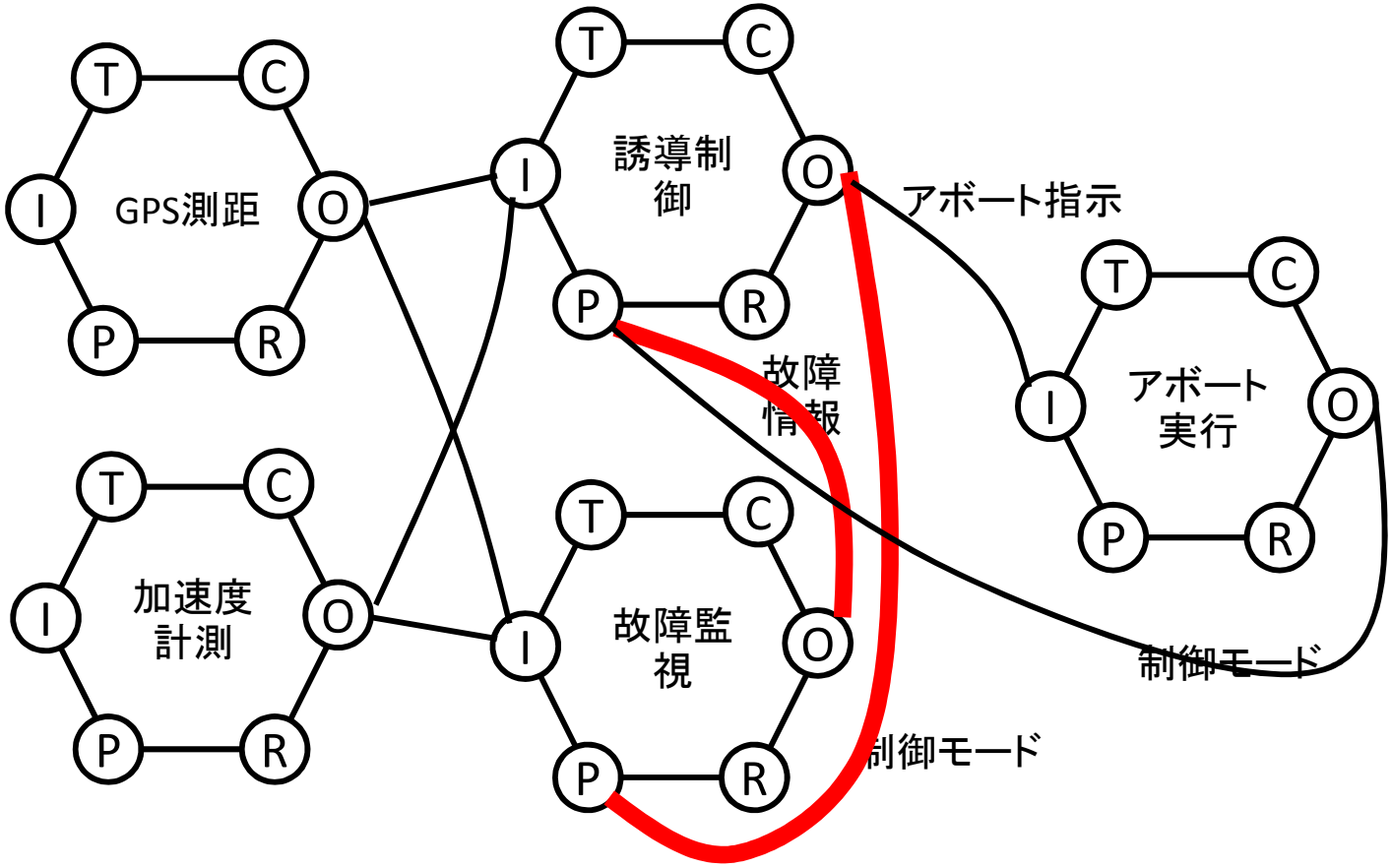


成功要因からリスクを見出す

成功要因は

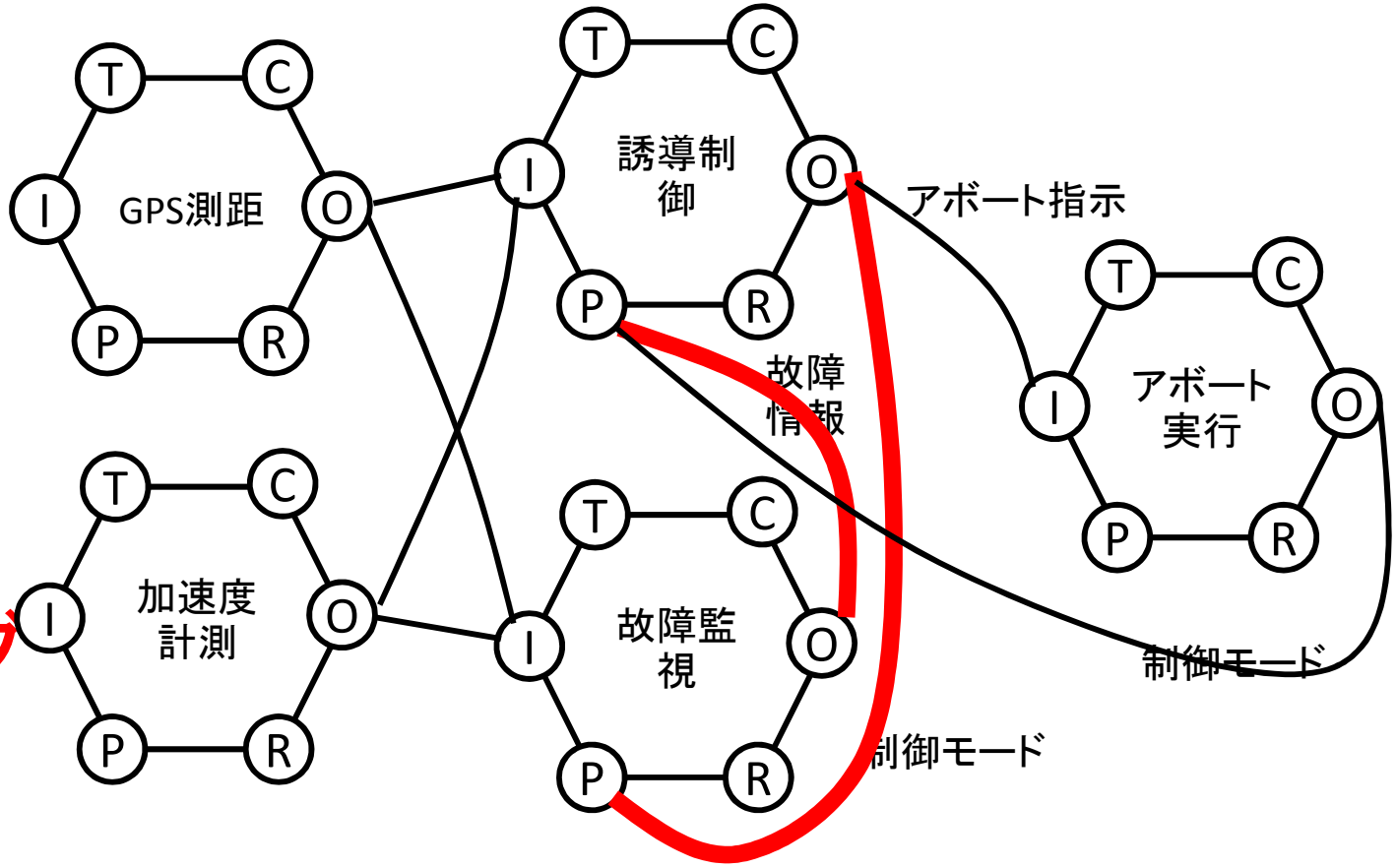


相互依存関係
にある

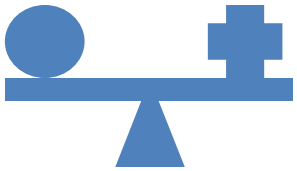


成功要因からリスクを見出す

成功要因の
裏には
リスク要因が
ある。
→レジリエンス
エンジニアリング

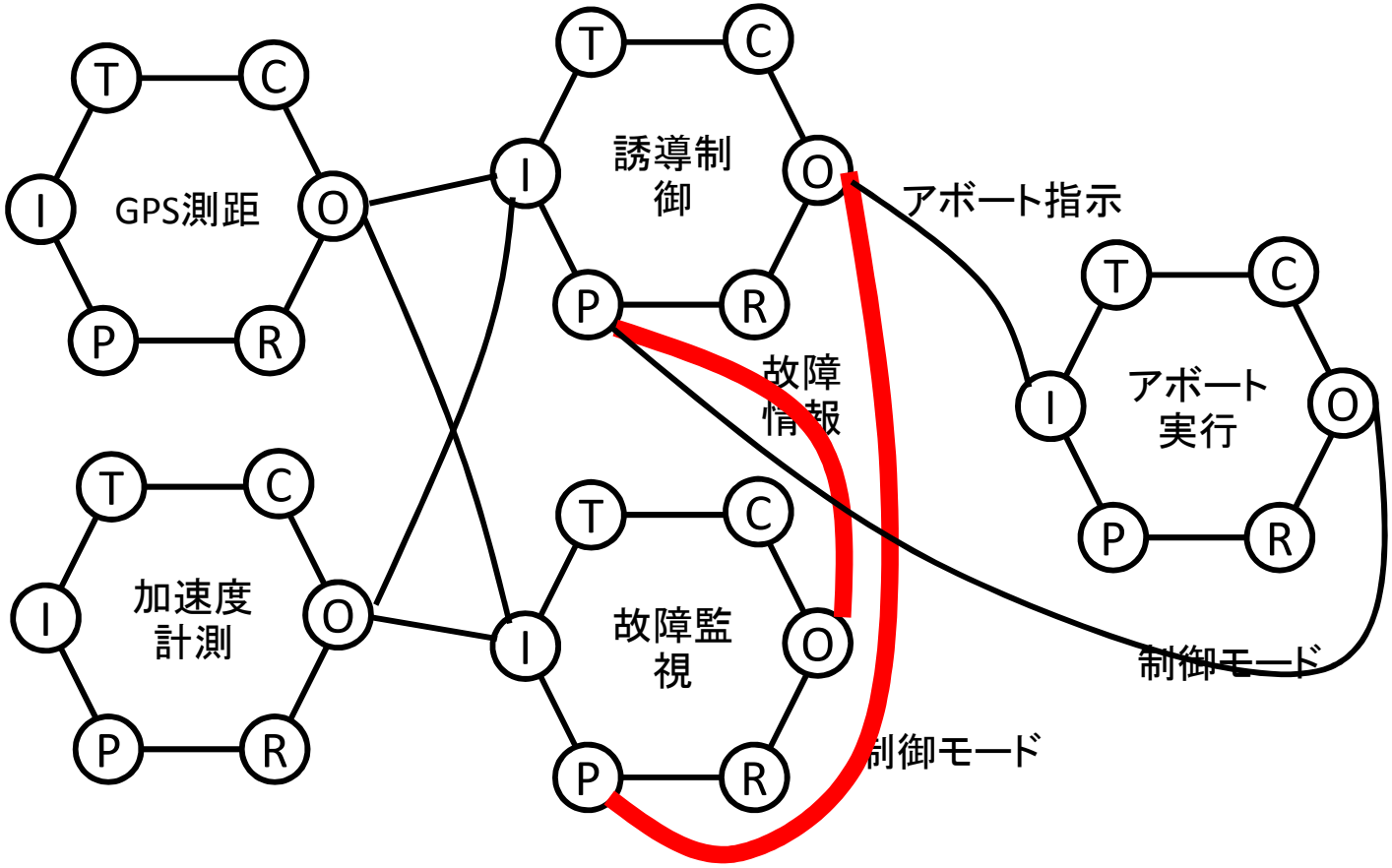


成功要因からリスクを見出す



相互依存関係の特徴

どちらが先に処理されるかによって、最終状態が変わる。

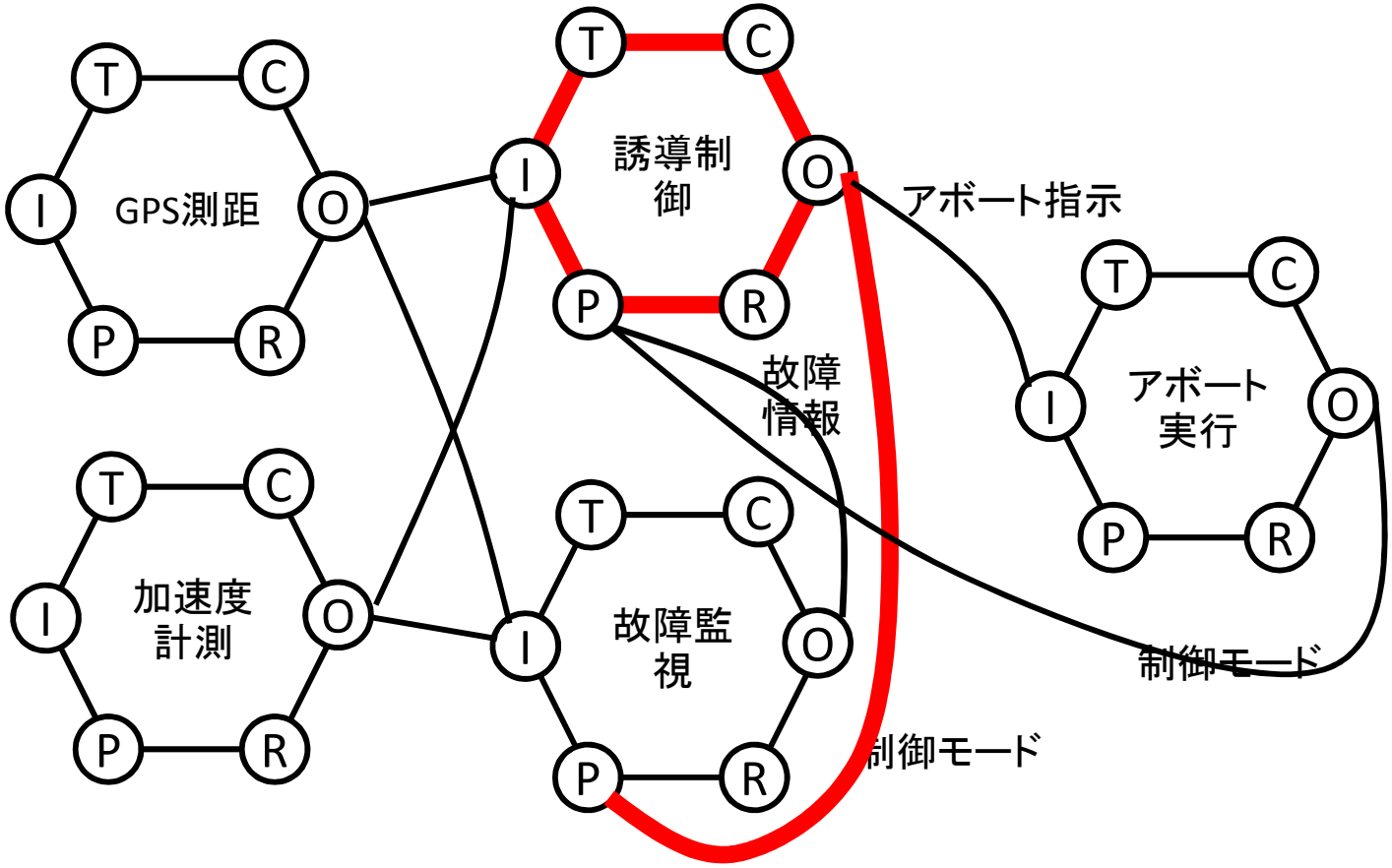


成功要因からリスクを見出す

誘導制御が
先に動作:

故障監視して
いないデータ
を使って誘導
してしまう

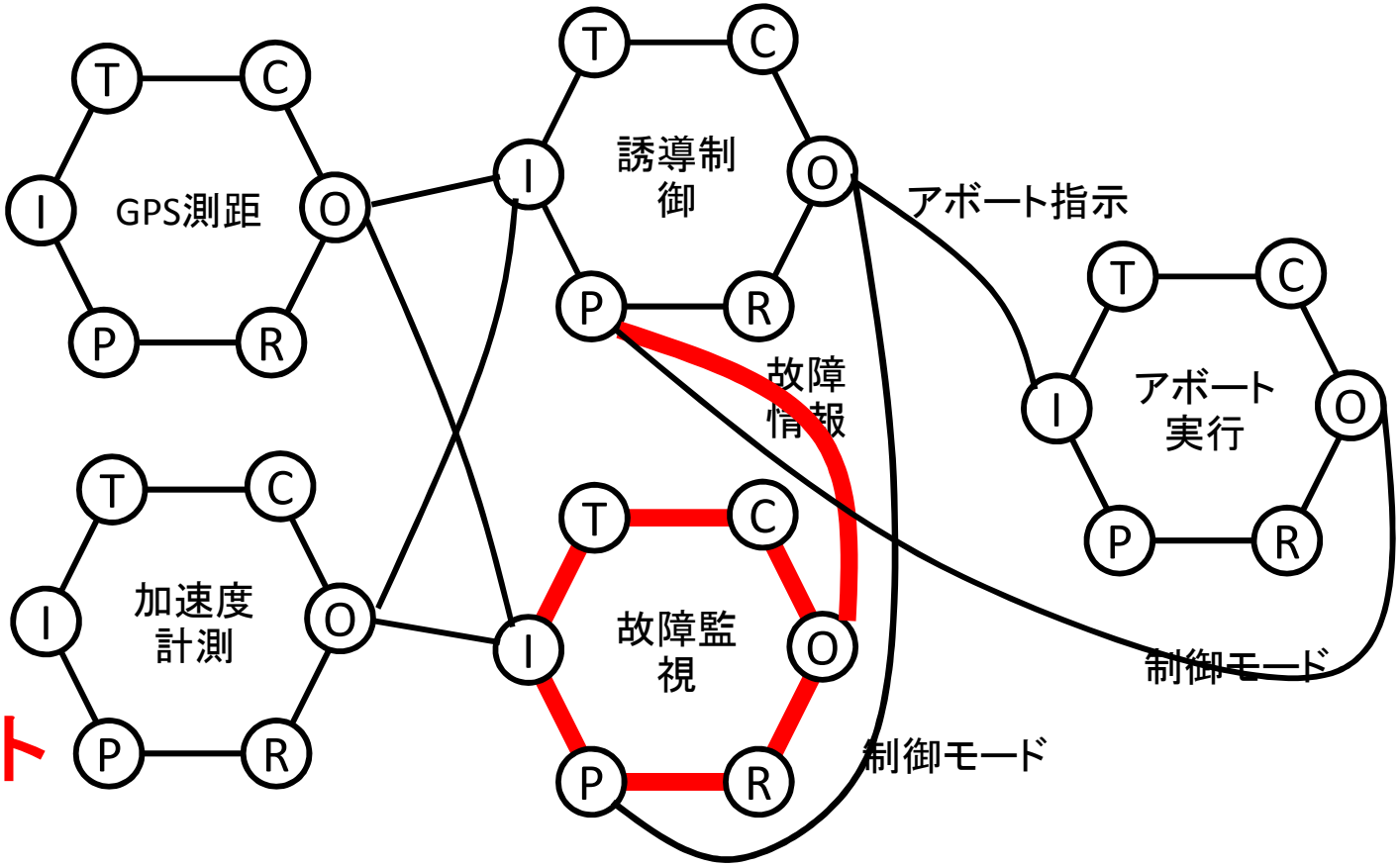
→異常な制御



成功要因からリスクを見出す

故障監視が先に
動作:

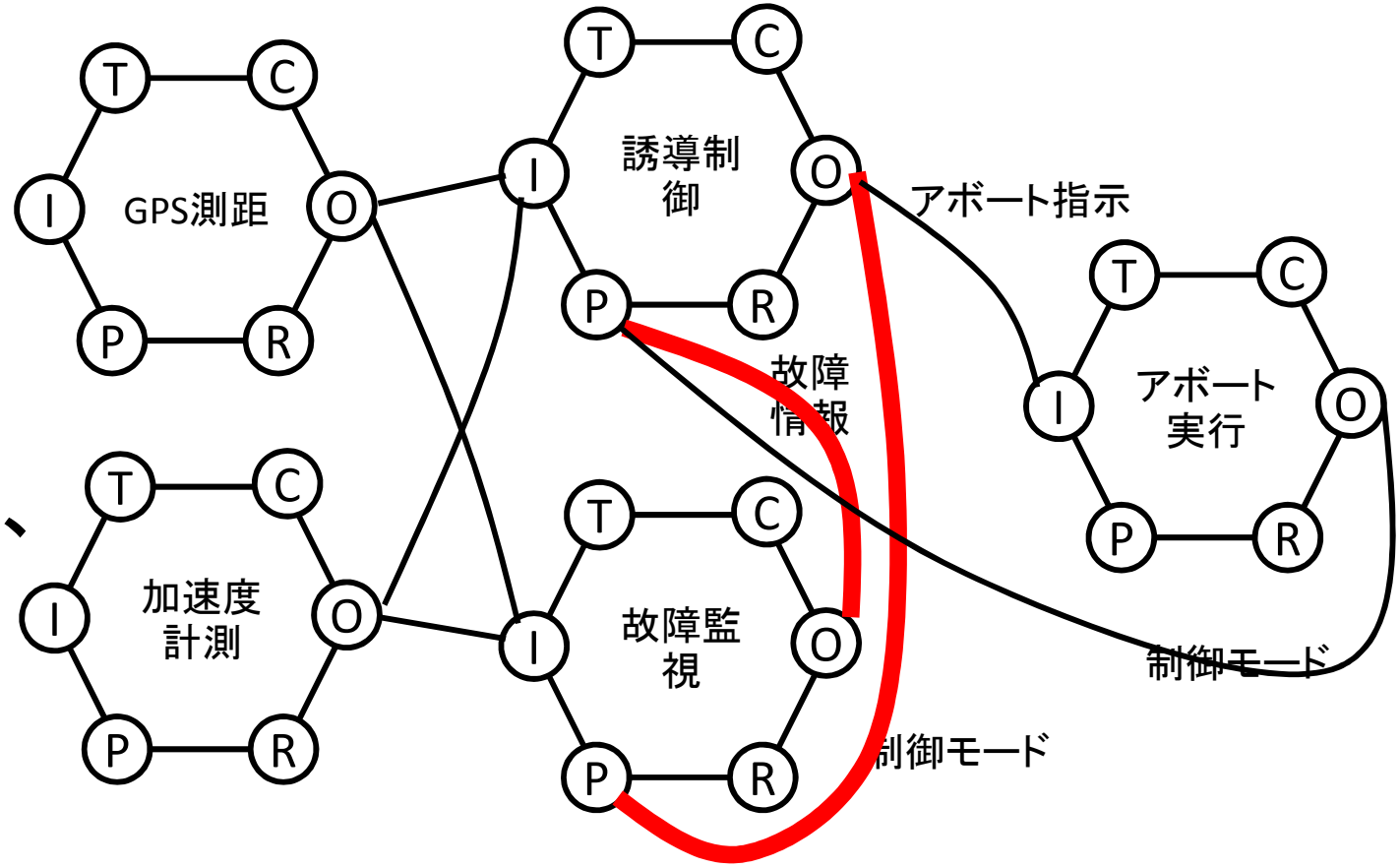
制御モードの
遷移直後に
遷移を知らずに
故障監視して
しまう
→不意なアボート



成功要因からリスクを見出す

リスク要因

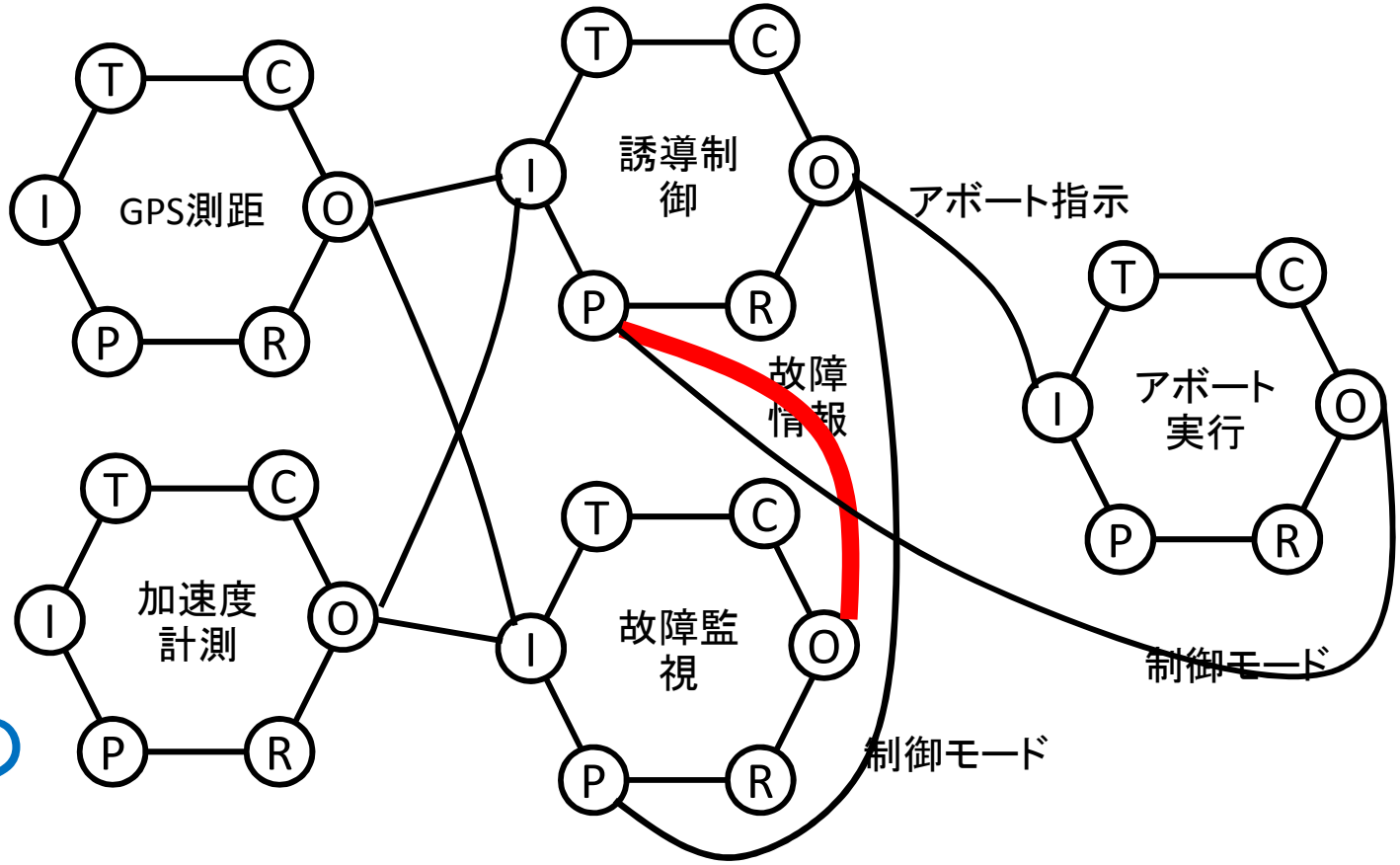
誘導制御と
故障監視が
互いに依存
するため、
処理順序により、
異常な制御
又は
不意なアボート
のリスクが有る。



成功要因からリスクを見出す

安全設計

故障監視を
先に実施する。
但し、
モード遷移直後
の監視結果は
使用しない。
相互依存関係の
強みを強化。



成功要因からリスクを見出す

従来の安全工学:
失敗要因が無い
ことを証明

レジリエンス工学:
成功要因を強固
にする設計を創出