ワークフローのモジュール化による ネットワーク運用管理自動化手法

大阪市立大学工学部　電気情報工学科
情報通信領域
瓦井 太雄

研究背景

- 近年，様々な分野において自動化が注目されている

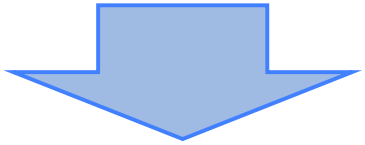
  Ex）工場での生産ライン，サービスの受付対応

- ICT，IT インフラにおいてフロースルーの全自動化が最終目標

  1. サービスオーダー
  2. プロビジョニング
  3. オペレーション
  4. アカウンティング

研究背景

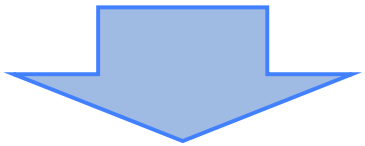
- システム運用管理(オペレーション) は自動化が進んでいない
  - ✓ 情報インフラ環境の複雑化
  - ✓ エラーの多様化に伴い制御構築の新規実装にかかるコストが増加
  - ✓ ケースバイケースな状況の把握と適切な処理の選択を必要とする複雑作業



複雑化するネットワークに追従できる自動化システムが求められる

研究目的

- 処理時間の短縮，二次的エラーの発生リスク低減
- 自動化ワークフローの**資源を再利用**できる実装フレームワークを確立



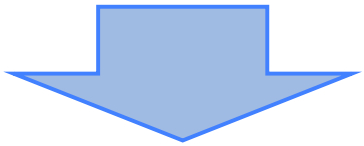
- ワークフローを構成する処理機能を分解し，それらを論理関係でつなぐ手法を提案

提案手法

1. 処理機能の**モジュール化**（再利用可能）

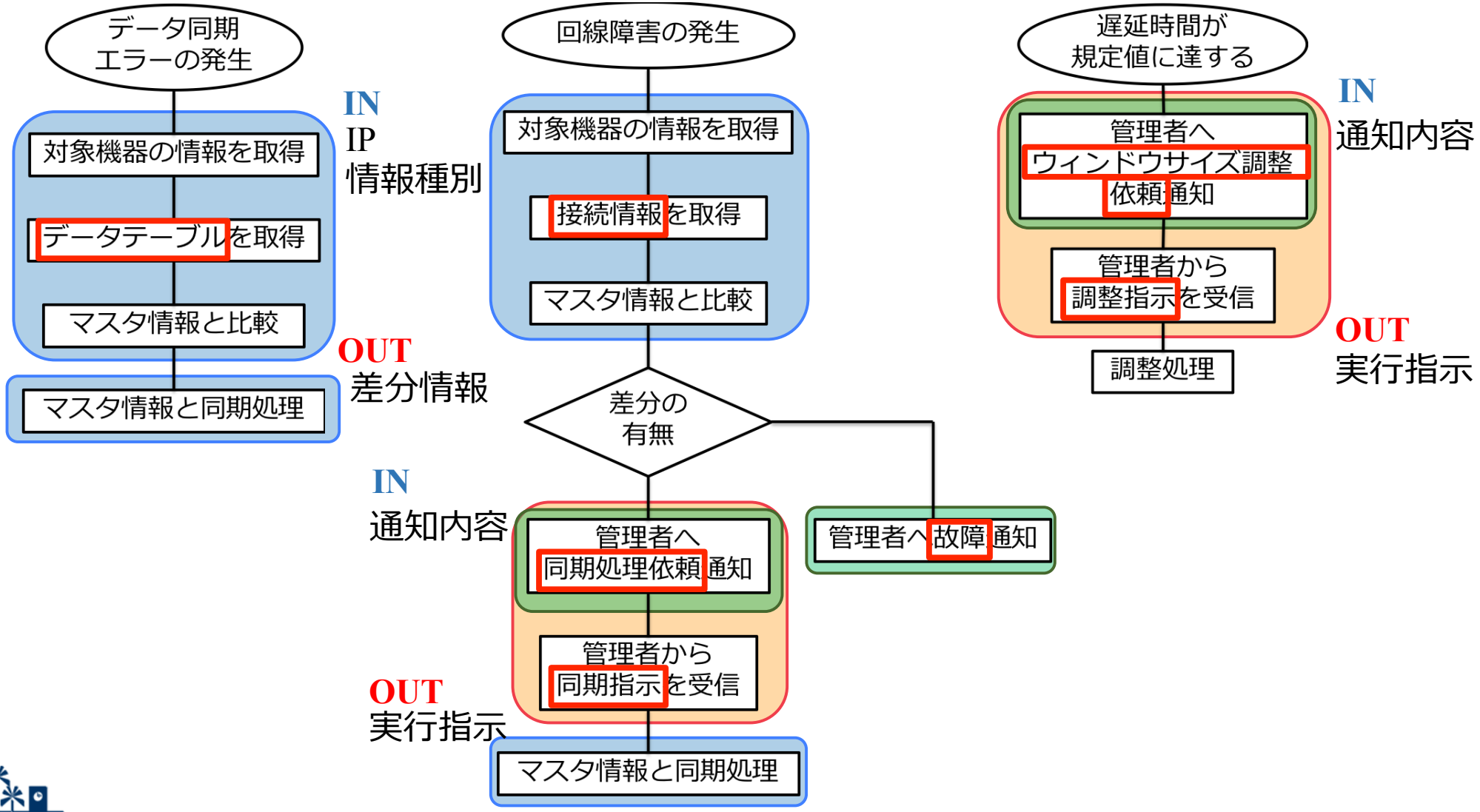
2. モジュールの API 化（処理の汎用化，安全性保持）

3. WMS を使用したワークフロー実装（制御ロジックに注力可能）



さらに複雑化するネットワークに追従できる自動化を実現する

ワークフローのモジュール化による再利用例

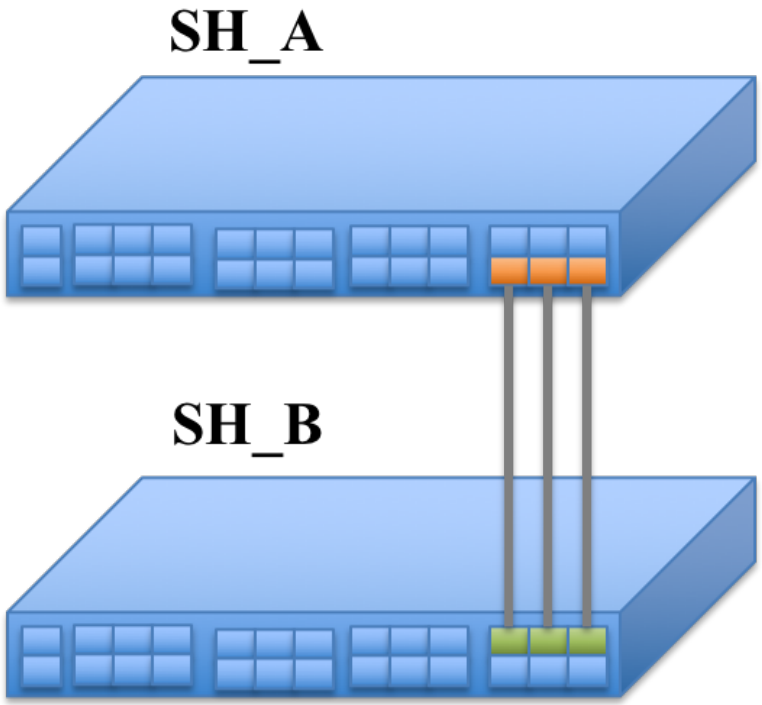


実装・検証

- 現場では自動化の進んでいない事例の一つであるリンクアグリゲーションの回線障害を実検証に取り上げ，アラート受信から原因判定までの処理を実装した
- 提案手法による実装と人の手入力の場合で処理時間，二次的エラーのリスク，提案手法の有用性を検討する

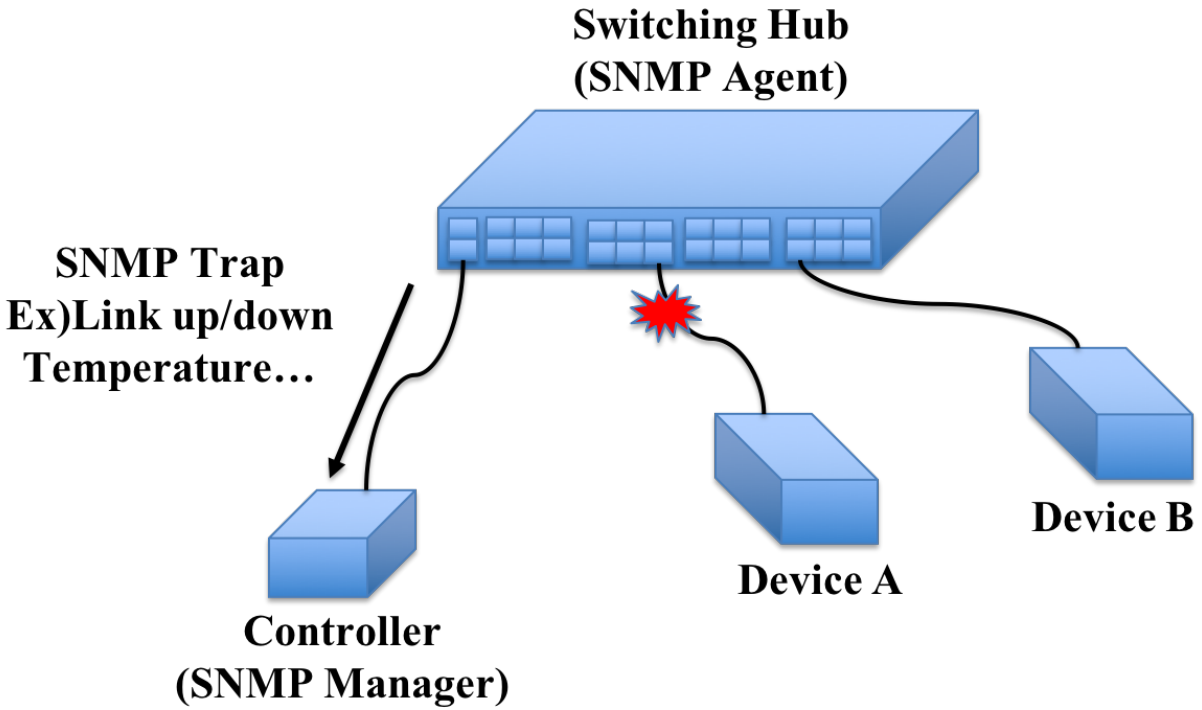
Link Aggregation(LA)

Ex)
ポート 1 つの通信帯域が 100Mbps のとき
SH_A のと SH_B の通信帯域は
100 [Mbps] × 3 [本] = 300 [Mbps]

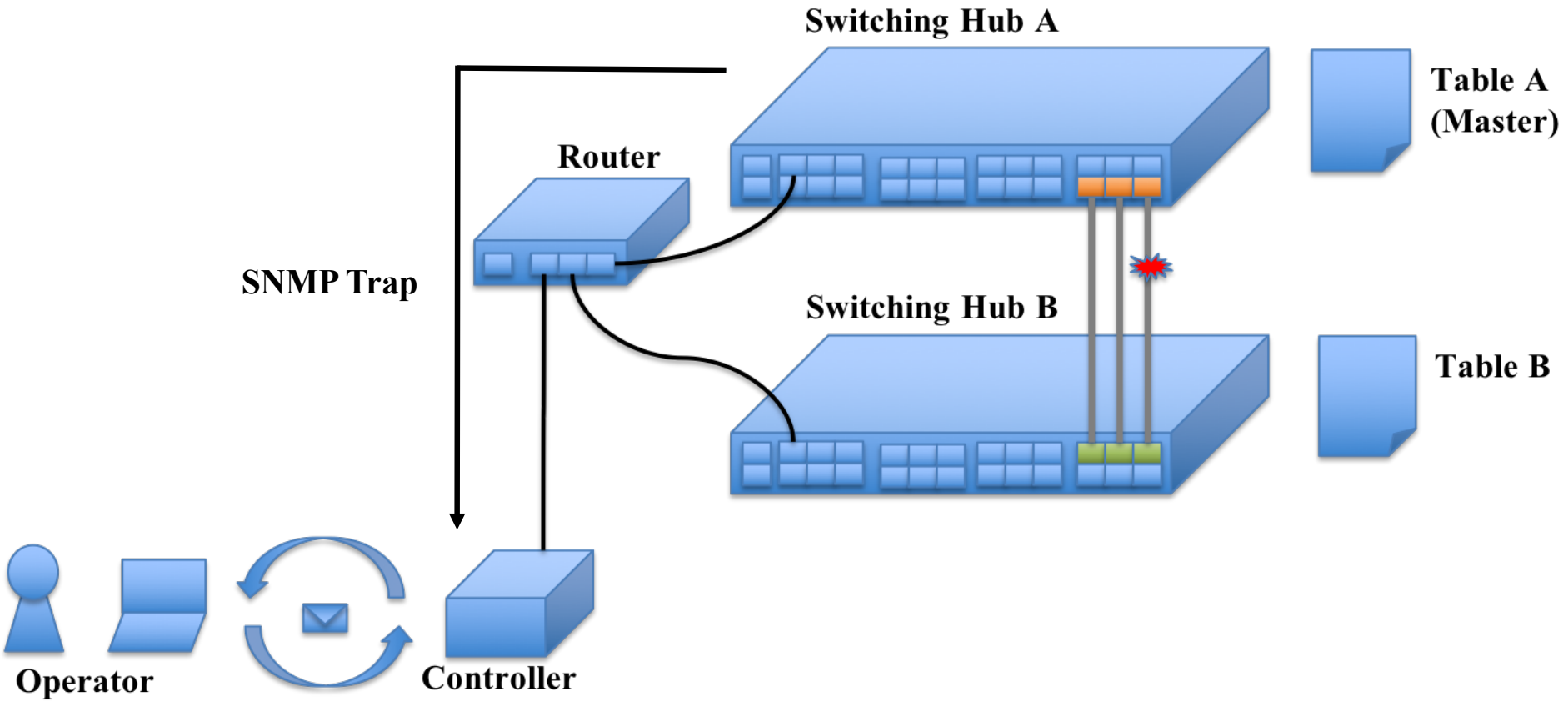


Simple Network Management Protocol(SNMP)

- ルータやスイッチングハブなどの機器の状態を監視するプロトコル
- マネージャからの要求に対しエージェントが情報を返したり，エージェントが自主的に状態の変化を通知する



実験環境


Switching Hub A
Table A
(Master)
Router
SNMP Trap
Switching Hub B
Table B
Operator
Controller

機器詳細(1)

- 各役割の機器の詳細表一覧

機器	製品型番	製造メーカー
ルータ	LSW5-GT-8NS	Buffalo
スイッチングハブ A	Catalyst 2960G	Cisco
スイッチングハブ B	Catalyst 2960	Cisco

機器詳細(2)

- コントローラマシンの詳細

CPU	Intel Pentium G4400 3.3GHz
メモリ	16GB
CentOS version	CentOS Linuxrelease7.4.1708
StackStorm version	st2 2.6.0
Python version	Python 2.7.5

人（手入力）による処理手順

- 監視対象に対して，LAG 情報を確認，比較

  ①ssh user@ip_addr　②(Password 入力)　③show ethernet summary

- 相違があった場合スイッチングハブ B に対して

  ④enable　⑤(特権モード Password 入力)　⑥configure terminal

- 本来と異なるポートを対象 LAG から外す処理

  ⑦interface range gigabitethernet 0/(対象ポート番号)

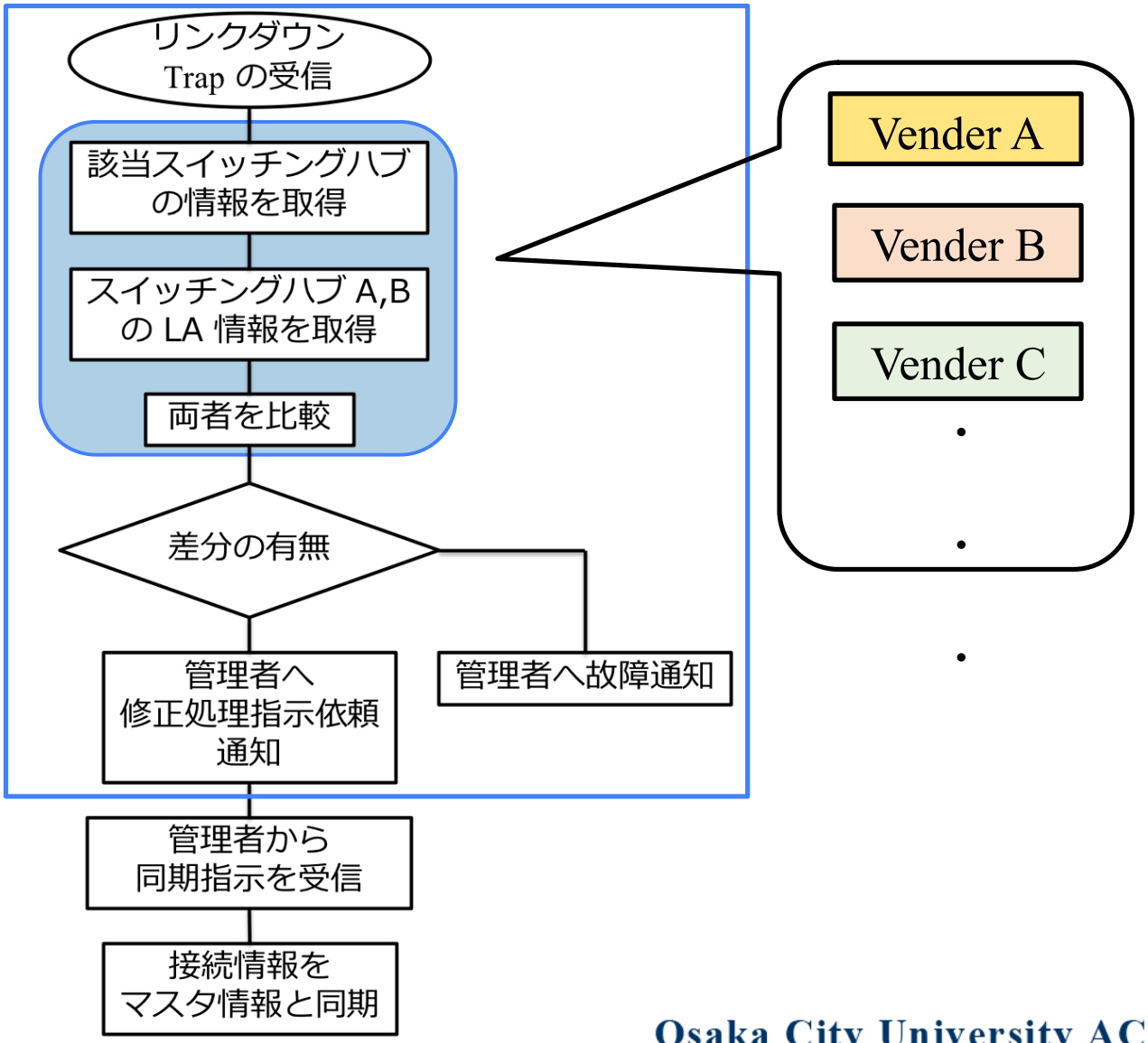
  ⑧no channel-group (対象 LA 番号)

- 足りないポートを対象 LAG に登録する処理

  ⑨interface range gigabitethernet 0/(対象ポート番号)

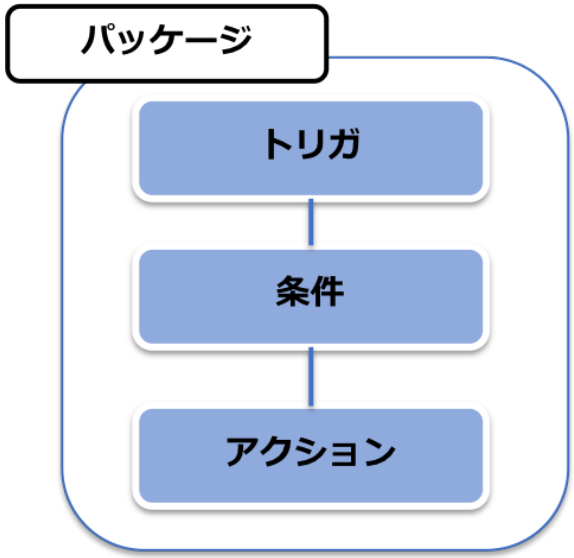
  ⑩channel-group (LA 番号) mode active

回線障害復旧ワークフローとベンダー依存

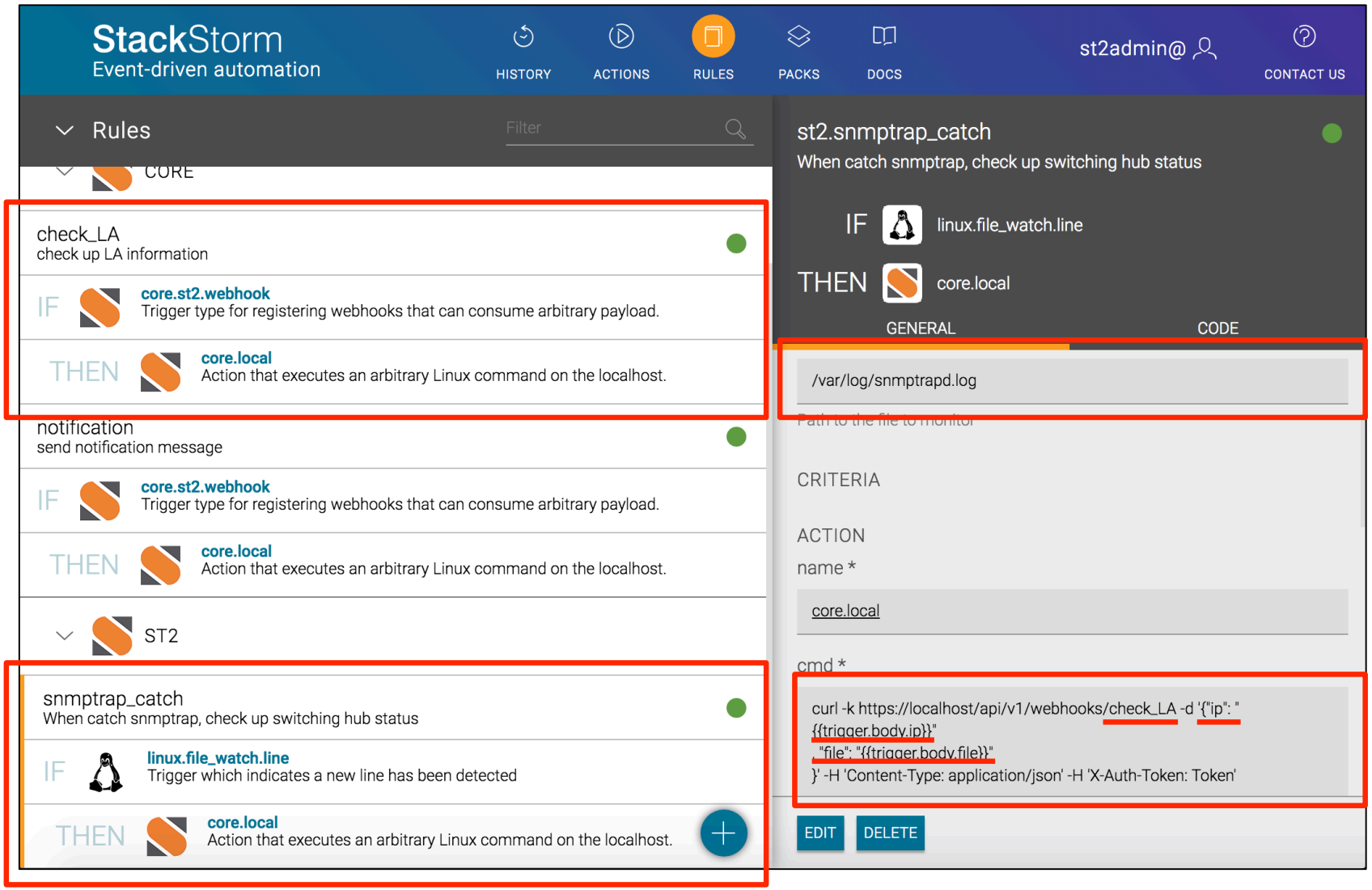


StackStorm（st2）

- ユーザや他の機器，プログラムが実行した操作（イベント）に応じてワークフローを実行する，イベント駆動型のワークフロー自動化プラットフォーム
- REST API や Python，Javascript でトリガやアクションの実装可能



st2 の webUI


StackStorm
Event-driven automation
HISTORY
ACTIONS
RULES
PACKS
DOCS
st2admin@
CONTACT US
Rules
Filter
CORE
check_LA
check up LA information
IF
core.st2.webhook
Trigger type for registering webhooks that can consume arbitrary payload.
THEN
core.local
Action that executes an arbitrary Linux command on the localhost.
notification
send notification message
IF
core.st2.webhook
Trigger type for registering webhooks that can consume arbitrary payload.
THEN
core.local
Action that executes an arbitrary Linux command on the localhost.
ST2
snmptrap_catch
When catch snmptrap, check up switching hub status
IF
linux.file_watch.line
Trigger which indicates a new line has been detected
THEN
core.local
Action that executes an arbitrary Linux command on the localhost.
st2.snmptrap_catch
When catch snmptrap, check up switching hub status
IF
linux.file_watch.line
THEN
core.local
GENERAL
CODE
/var/log/snmptrapd.log
CRITERIA
ACTION
name *
core.local
cmd *
curl -k https://localhost/api/v1/webhooks/check_LA -d '{"ip": "
{{trigger.body.ip}}"
, "file": "{{trigger.body.file}}"
}' -H 'Content-Type: application/json' -H 'X-Auth-Token: Token'
EDIT
DELETE

結果

- ワークフローに要する処理時間はおよそ 12 分の 1に短縮された
- 人の数値入力ミスの発生率は $8\%_{[1]}$ とのデータもあり，オペレーションステップの削減が二次的エラーの発生リスク低減に繋がると考えられる
- 提案手法による処理機能のモジュール化により，資源の再利用が可能な形で実装した

回目	st2[sec]	人（tab 補完使用）[sec]
1	9.45	219.09
2	8.05	124.12
3	8.91	97.58
4	8.23	89.45
5	7.86	86.17
6	8.50	87.83
7	7.73	103.67
8	8.25	89.30
9	8.83	95.22
10	10.05	76.93
平均	8.586	106.936

[1]吉村治正, 小久保温, 澁谷泰秀, and 渡部諭, “社会調査の入力ミスの発生率について,” 青森大学付属総合研究所 紀要, vol. 15, no. 1, pp. 1–5, 2014.

まとめ

- 回線障害復旧を取り上げ，モジュール化した処理の連結による
  ワークフローの自動化を実装した
- 処理時間と二次的エラーの発生リスク低減を検討した

- 今後の課題
  - ✓ 提案手法によるモジュールの再利用性の確認
  - ✓ 処理モジュールの単位決定をアルゴリズム化
    （トップダウン設計，STS 分割，TR 分割）