OKI OKIソフトウェア

Situator® ご説明資料

2011年9月

OKIソフトウェア

目次

Situator® とは

◆ 株式会社OKIソフトウェアが長年の通信系ソフトウェア開発の中で培い、保有ツール＆ノウハウをパッケージ商品として整備した「擬似対向テストツール」として誕生しました。現在は、「インテリジェントな通信ミドルウェア」として多方面でご利用頂いています。

- 2004年7月：Ver1.0発売開始
- 2006年8月：プロトコルオプション追加
  - HTTP、SNMP、SIP プロトコルに対応
- 2007年10月：Ver2.0を発売開始
  - シナリオエディタ（GUI操作でのシナリオ作成）を搭載し、より簡単な対向装置構築を可能化
  - Windows Vista に対応
- 2009年4月： プロトコルオプション追加
  - FTP、SSL/TLSプロトコルに対応
  - シナリオエディタのプロトコルオプション対応機能を強化
  - Windows Server 2008に対応
- 2010年1月：通信ミドルウェアへコンセプト変更
- 2010年10月：対応プラットホームにLinuxを追加
  - Redhat EL 5.5、CentOS 5.5に対応
  - SIPプロトコルに対応
- 2010年11月：対応OS拡充
  - Windows 7 (32bit/64bit)、Windows Server 2008 R2 (64bit)に対応
- 2011年9月：SIPプロトコル機能追加、対応OS拡充
  - SIPプロトコル機能追加（TCP対応）：Windows版/Linux版
  - 64bit Linux対応：Redhat EL 5.6 (64bit)、CentOS 5.6 (64bit)

通信プログラムのプラットフォームとして (1/4)

短期間・高品質

- **慣れた言語で通信手順を意識せずにアプリケーション開発ができます。日本、海外のATMでの稼働実績があります。**

多彩なプロトコル

- HTTP、FTP、SNMP、SIP、SSL/TLS**の各種オプションを利用することができます。**

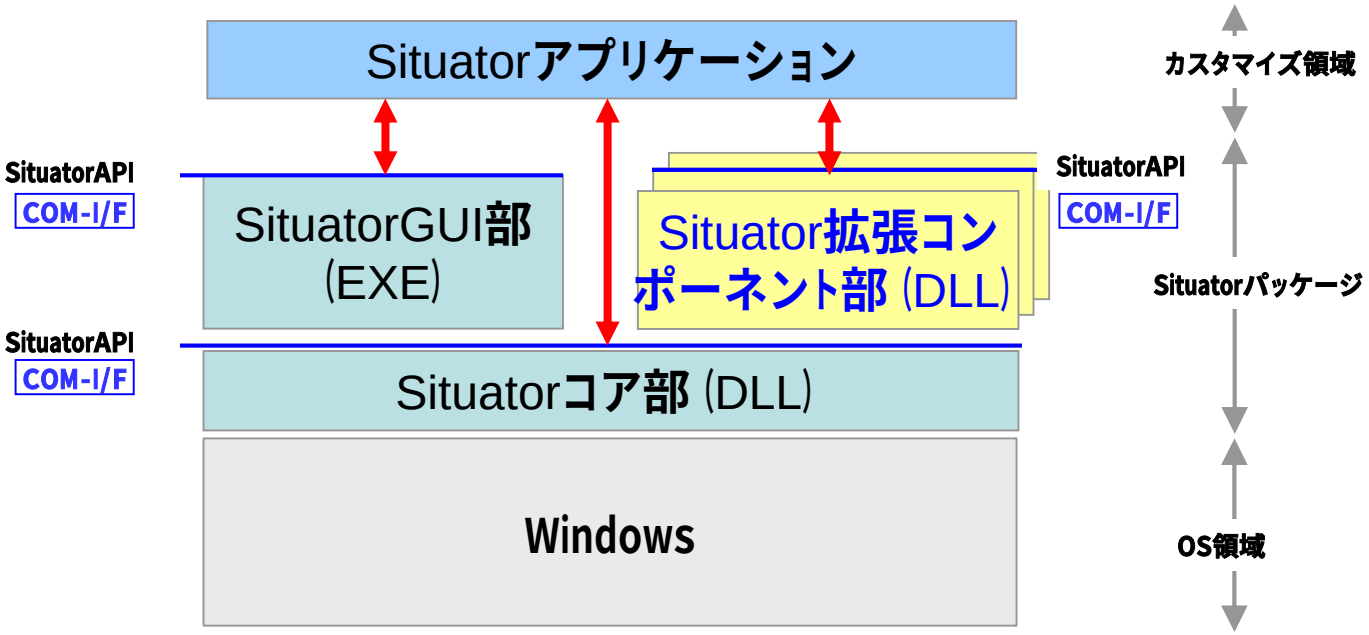
マルチプラットホーム

- Windows**系に加えて**Linux**系でもお使い頂けます。**

通信プログラムのプラットフォームとして (2/4)

◆MicrosoftのCOM (Component Object Model) 技術を中核にして開発

- コンポーネント間のインターフェースはすべてCOMで定義
- 各機能・部品をCOMコンポーネント化し、再利用可能な構造に
- 他のアプリケーションから部品としての利用が簡単に
- COMを使える言語であれば利用可能 (言語を選ばない)
- Windowsで標準のインターフェースであり、各種開発ツールからの利用が容易



通信プログラムのプラットフォームとして (3/4)

◆ アドイン実行

- SituatorにDLLやスクリプト、シナリオを組み込んで実行
- カスタム機能を自由に追加できる

（参考：P19 スクリプト例）

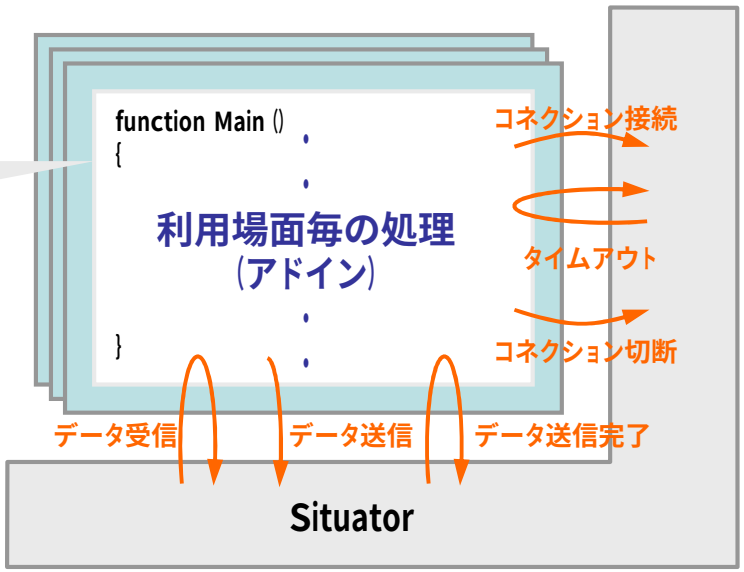
◆ Situator上でアプリケーションを実行する仕組み

- スクリプト、シナリオ実行環境を提供
- ActiveXコントロールのホストに対応
- C/C++のDLLのロード、実行

《プログラミング対応言語》

- VBScript
- JScript
- VisualBasic
- C/C++

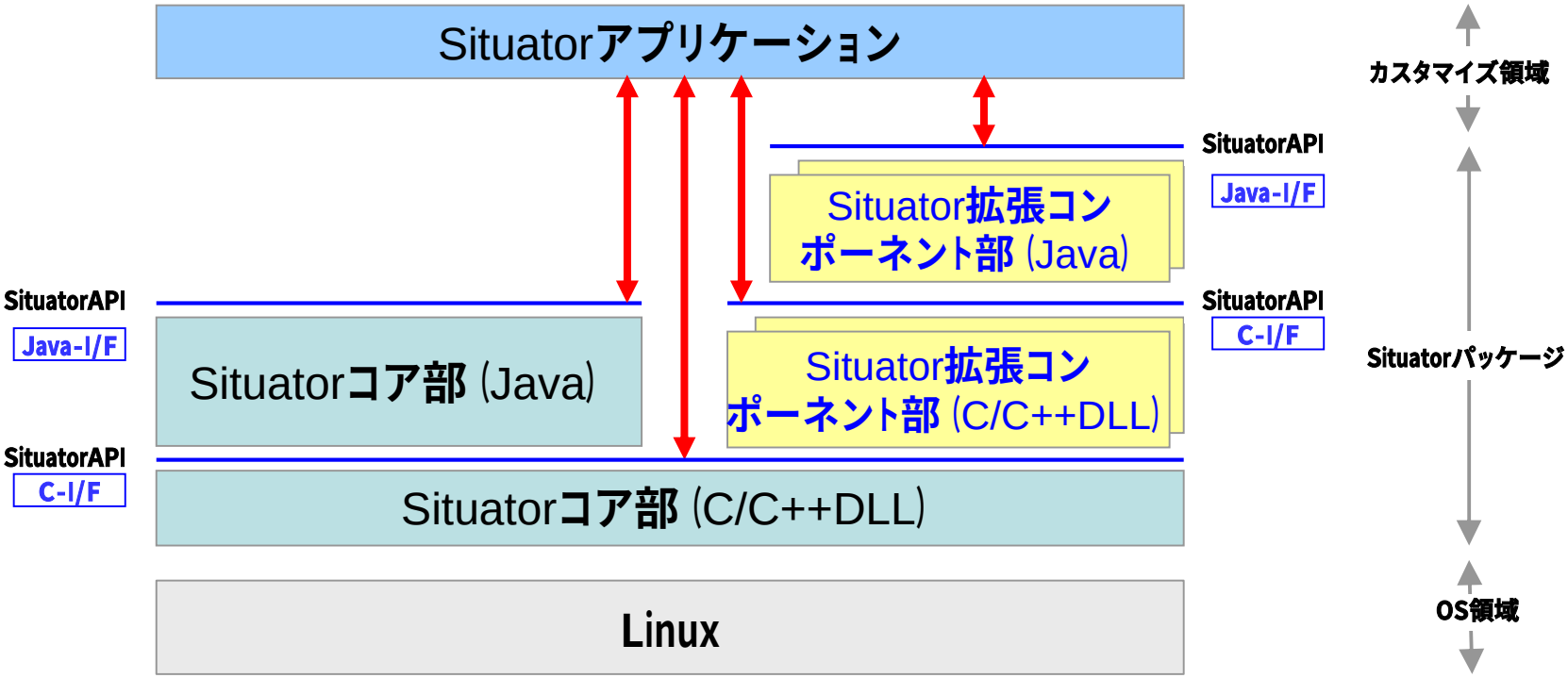
その他、COMを利用可能な言語



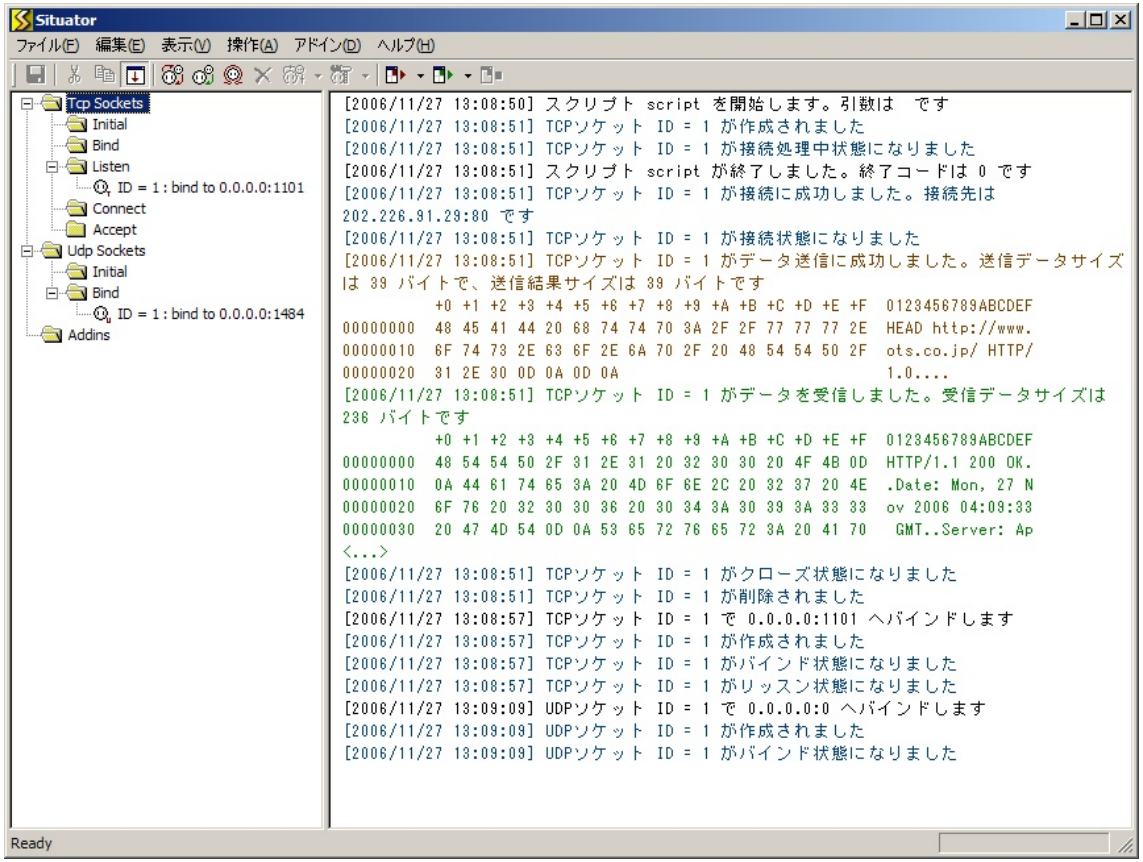
通信プログラムのプラットフォームとして (4/4)

◆対応OSにLinuxを追加

- コンポーネント間のインターフェースはC/C++インタフェースとJavaインタフェースを定義
- アプリケーションは、C/C++,Java,JavaScriptにて開発可能



テストツールとして (1/3)



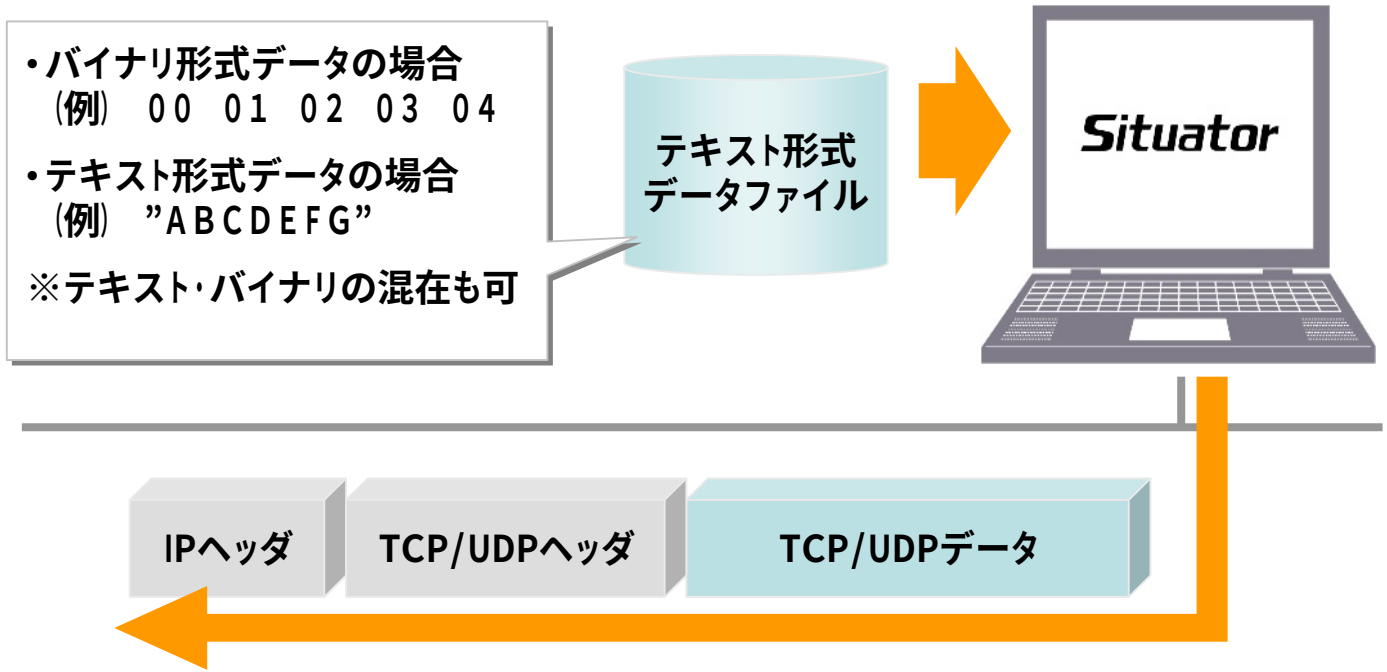
●表示系機能

- 送受信ログ自動収集
- コネクション状態表示
- 受信データの自動表示

●送受信系機能

- ソケットに対する受信データ自動読み出し
- バイトストリームからのメッセージ組み立て
- データファイルや画面入力データの送信
- 複数コネクション
- TCP/UDP、IPv4/v6

テストツールとして (2/3)

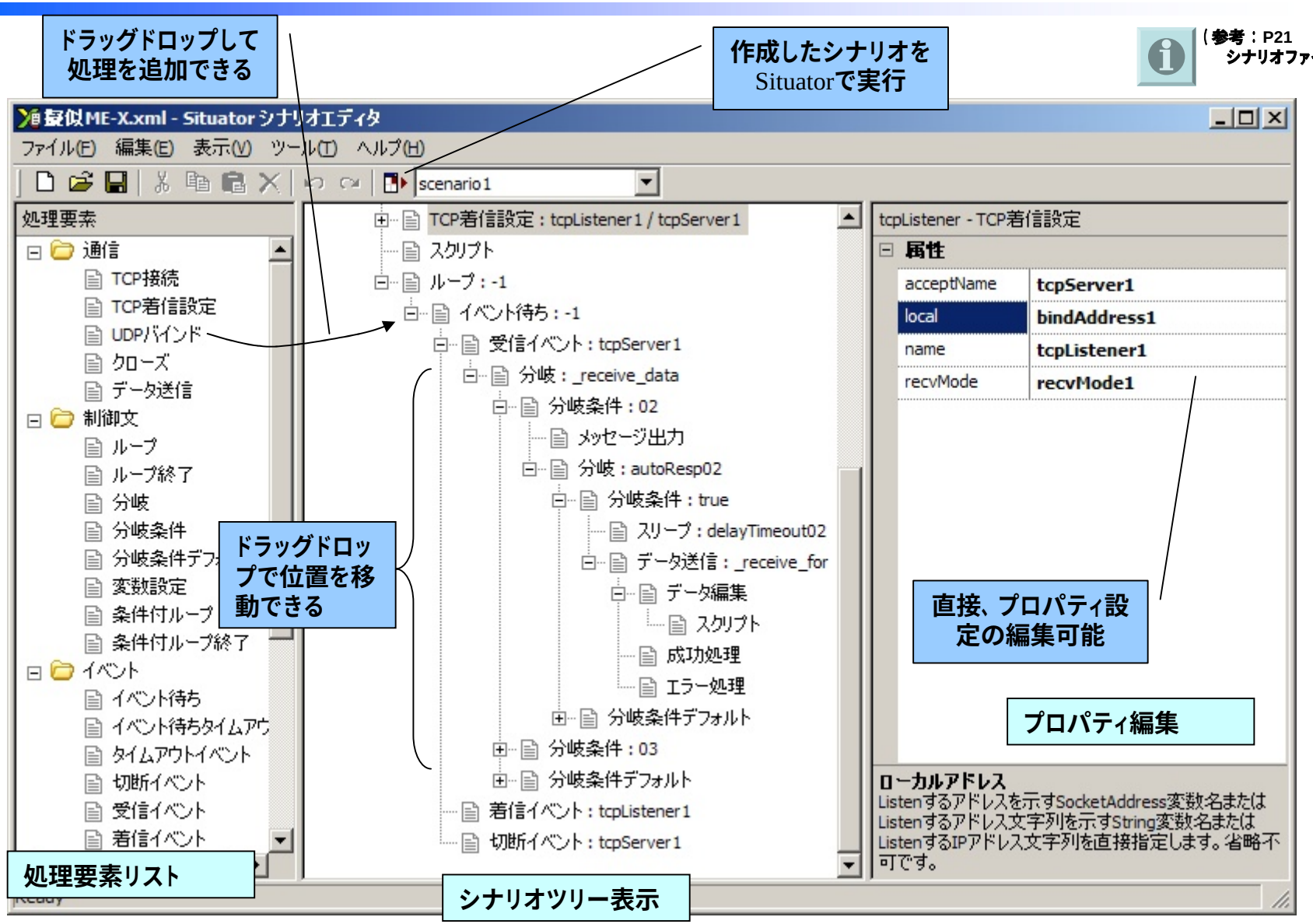


・受信設定ファイル（XML）で電文のサイズ情報を定義して、Situatorに指定しておくと．．．

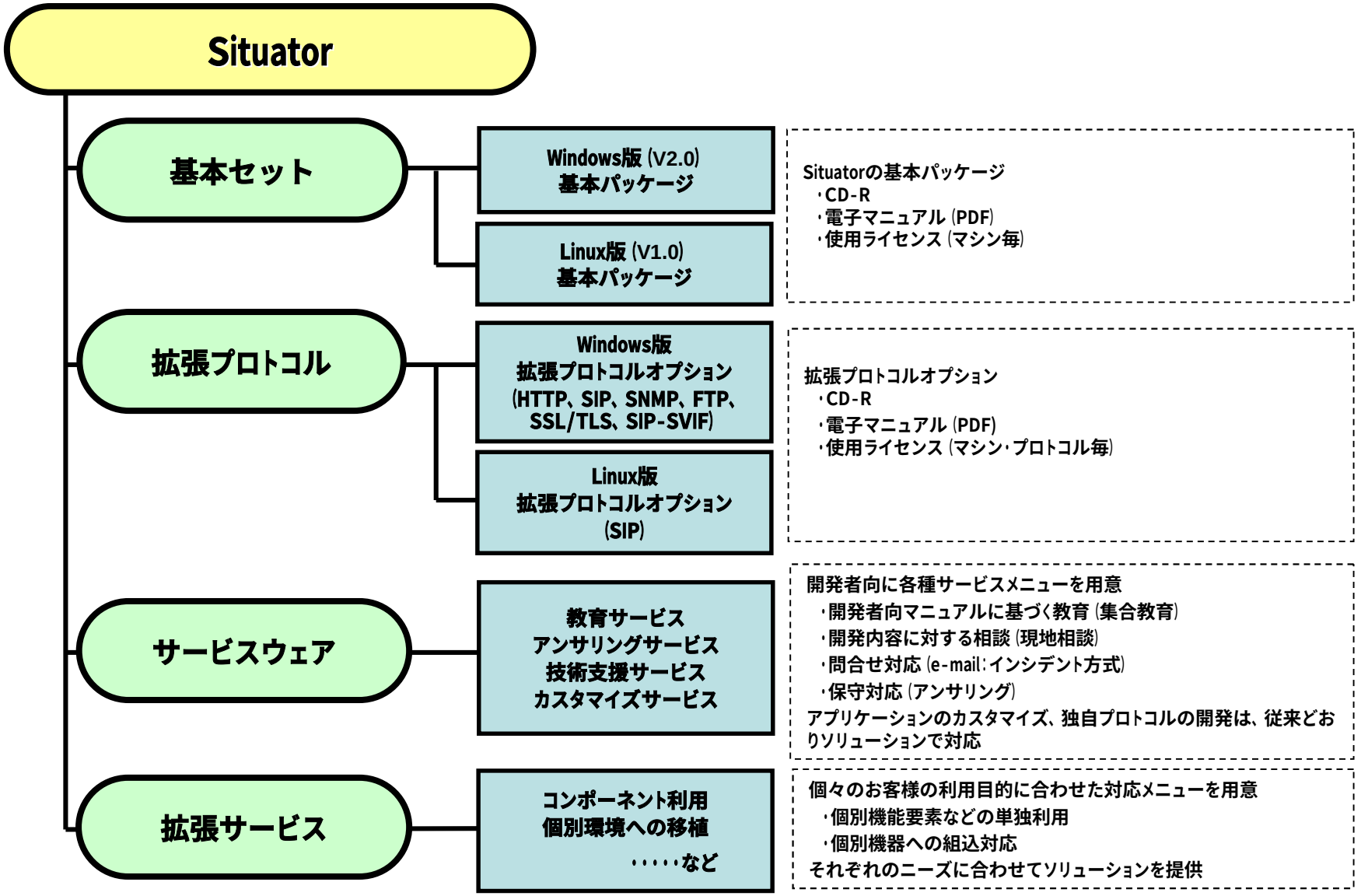
➢TCPストリームデータ受信時に、Situatorが自動的に電文を組み立てて、電文単位で通知してくれる

（参考：P20
受信設定ファイル）

テストツールとして (3/3)



商品体系



活用事例

開発事例（1/2）

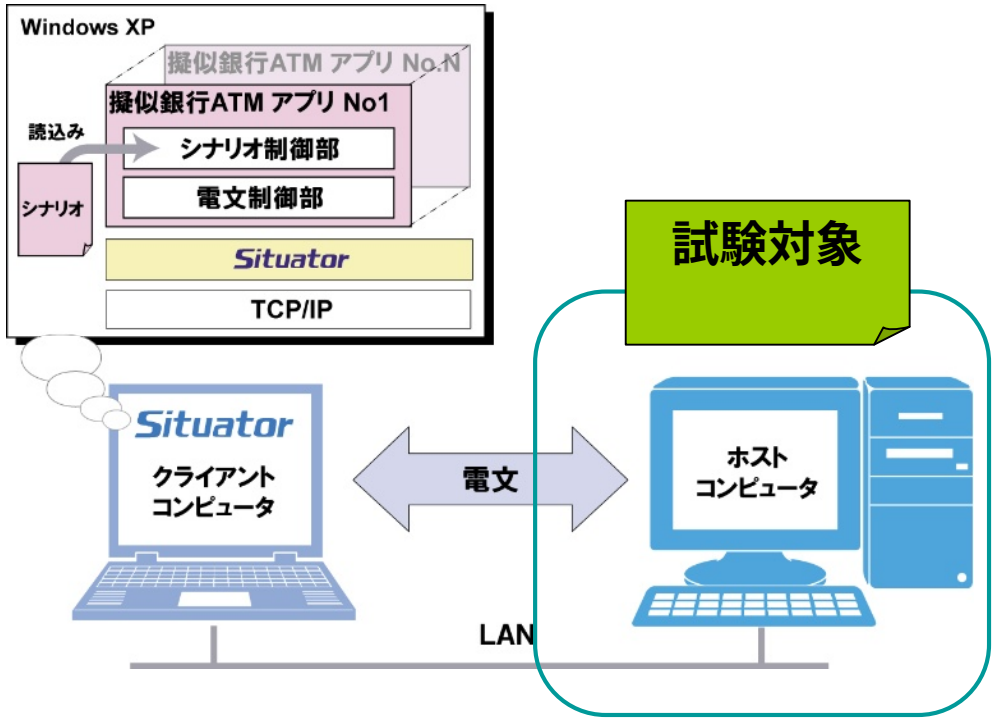
	開発事例	主な機能	用途
★	擬似クライアント	自動発信、複数コネクション制御、 送受信ログ、送受信電文の詳細表示 電文編集送信、画面操作による電文送信 自動応答、シナリオモード	単体、結合、負荷試験における擬似クライアント
	擬似ホスト	自動着信、複数コネクション制御 送受信ログ、送受信電文の詳細表示 画面操作による電文送信、自動応答	単体、結合、負荷試験における擬似ホスト
	擬似SNMPマネージャ	複数SNMPエージェント管理 SNMPエージェント間の送受信状況表示 送受信シナリオ実行	複数のSNMPエージェントの連続試験
★	SOAP-X25ゲートウェイ	回線の制御（HTTP、X25） 送受信ログ、通信モード切替（GW、折り返し） SOAP-X25電文変換	SOAP通信とX25通信のゲートウェイと電文変換
	HTTP文字コード変換	HTTPプロクシ POSTデータの文字コード変換	文字コード変換に対応したHTTPプロクシ
	ATM通信コンポーネント	銀行ホストとの通信制御	ATMプラットフォームにおける銀行ホスト間の通信制御コンポーネント
★	擬似SIP端末	SIP端末として連続発呼動作、着呼待ち動作、 転送動作、一括同時発呼動作	本物のSIP端末の代わりに発呼、着呼、転送、連続発着呼の動作を擬似する
	データ収集ツール	画面操作によりデータ送受信を指示 取得したデータをCSV出力	HTA（HTMLアプリケーション）による運用支援ツール

★:詳細有

開発事例（2/2）

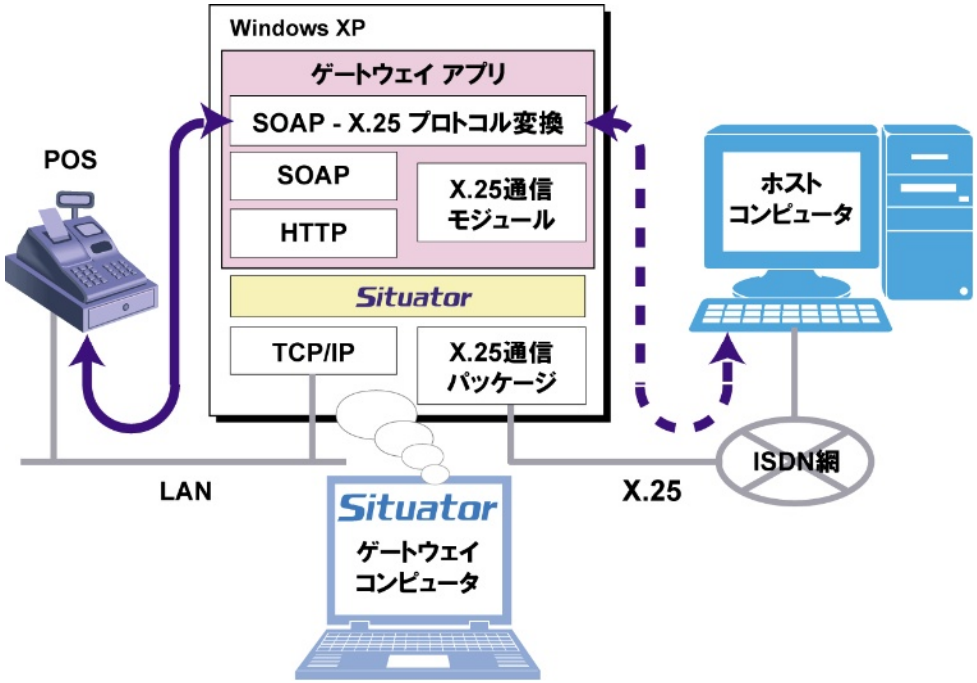
	開発事例	主な機能	用途
★	店舗サーバ	自動応答、DB連携、プリンタ制御	店舗内オーダリングシステムの店舗サーバ
★	ファイル転送モジュール	暗号化 圧縮 チェックサム	システムの保守・運用ツール ・ジャーナル収集 ・ログ収集 ・ファイル配布
★	プロキシサーバ	認証 アクセス先の監視 HTML書き換え リクエスト振り分け	Webアプリケーションサーバ テスト用対応サーバ

擬似ATMクライアント



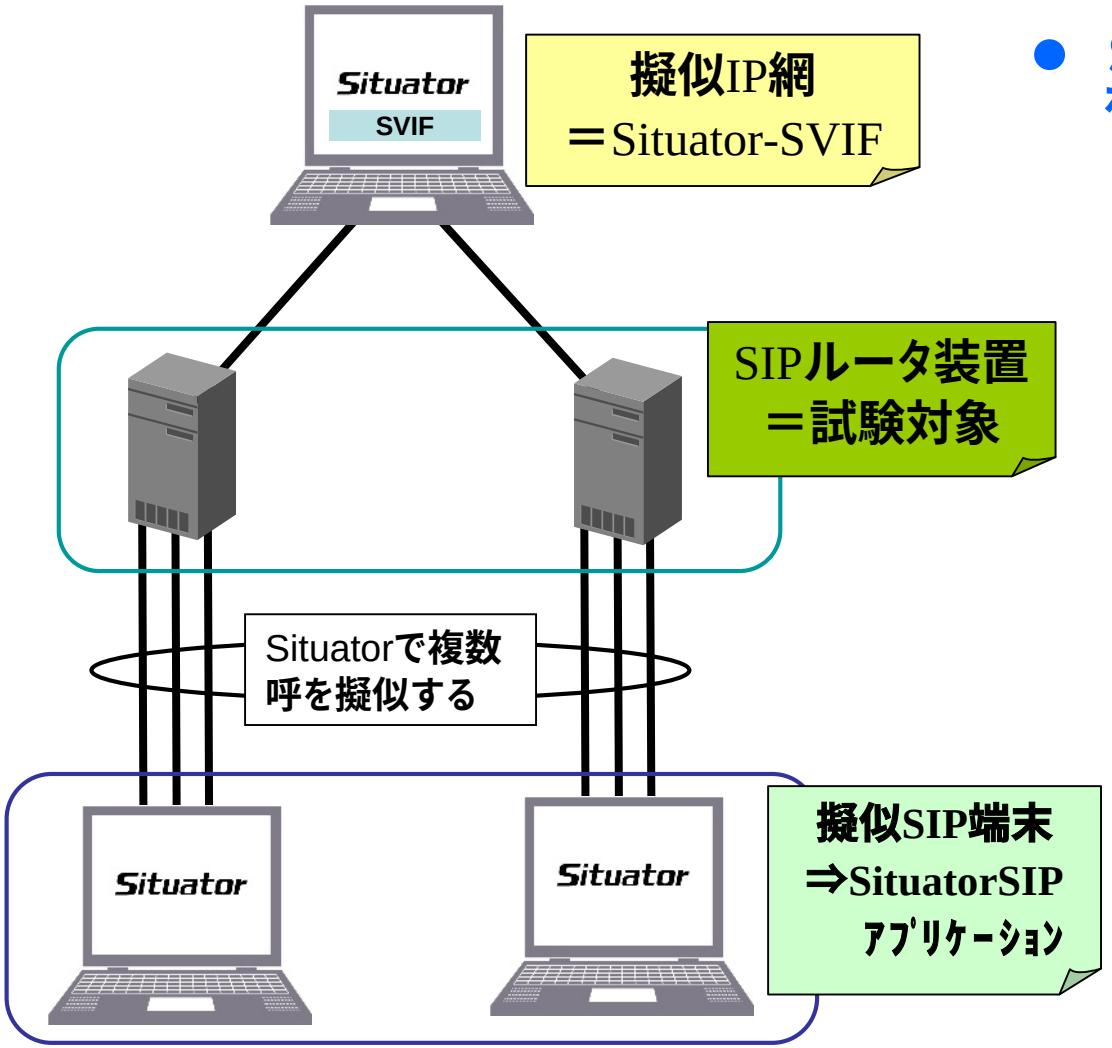
- シナリオ制御のテストツール
  - 1つのパソコン上で擬似的に複数台の銀行ATM動作をシミュレートする
  - 電文を自動的に編集し応答する
    - シーケンス番号、時刻等
  - シナリオの実行により
    - 電文の連続送受信を実施
    - 負荷を自動的かつ継続的に発生
    - 特定の通信シーケンスを簡単に再現

SOAPとX.25のゲートウェイ装置



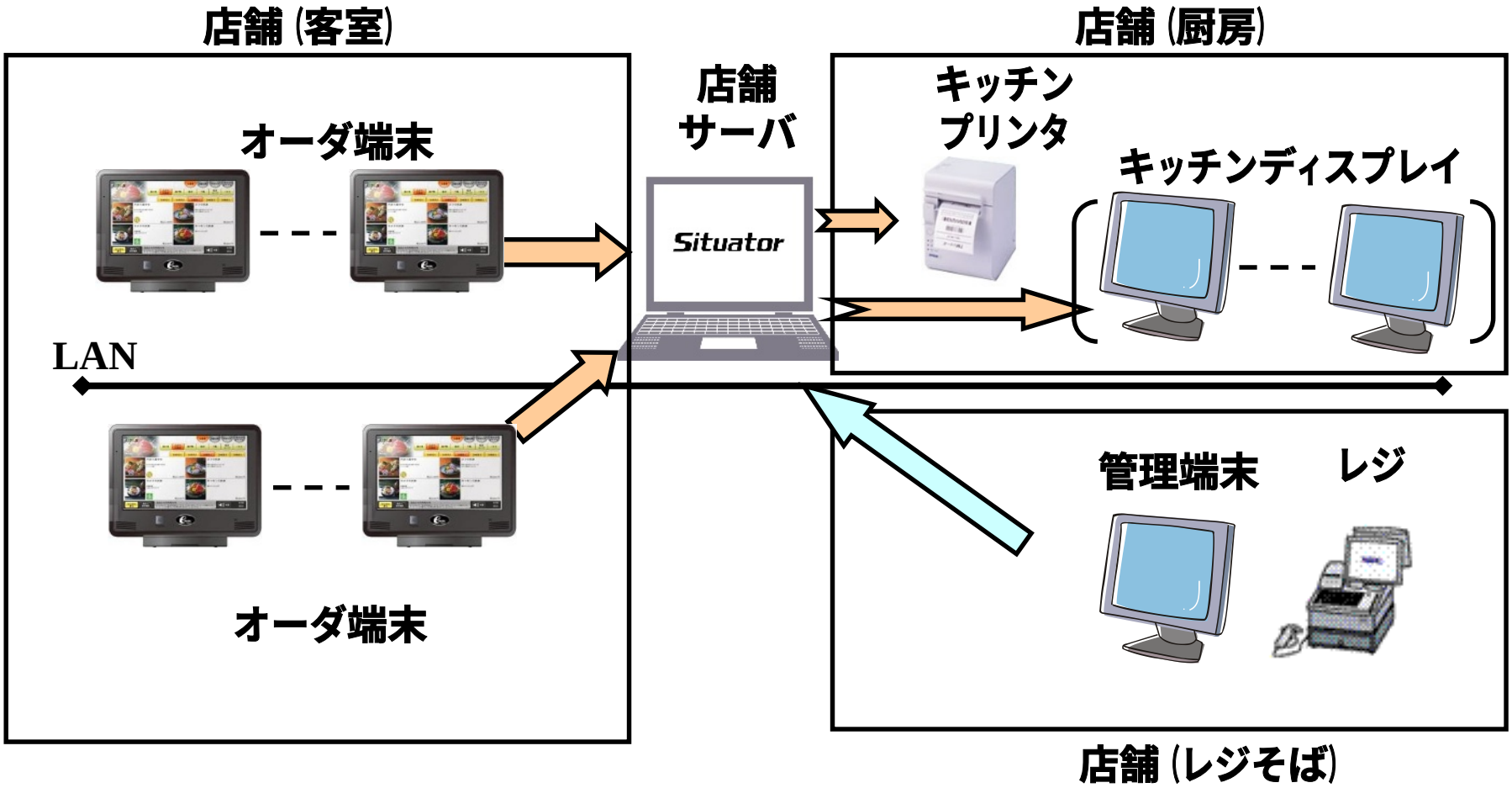
- Situator上のアプリケーションで、SOAPとX25の中継処理を実現
  - SOAPメッセージとホスト向け電文の相互変換
  - SOAPメッセージはMSXMLを使って操作
  - XMLスキーマでSOAPメッセージのパラメータチェック
  - 応答メッセージはXSL Transformationsで生成
  - POSとホスト間の通信文字コードが異なる
    - 電文変換時に文字コードの相互変換

擬似SIP端末／SIP網



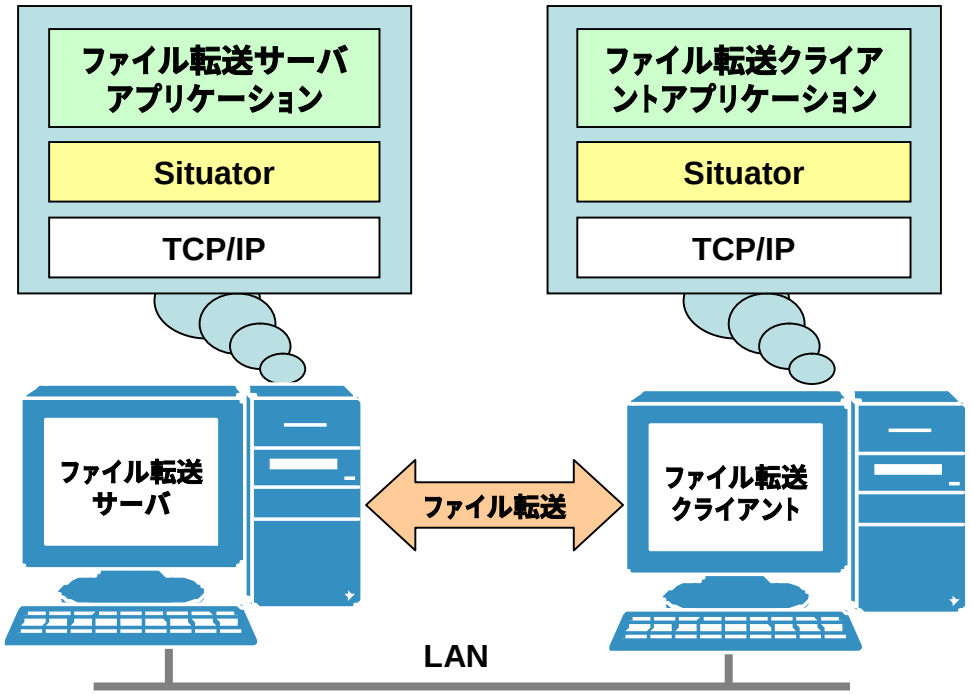
- **SIPルータ装置の擬似試験環境を構築**
  - 発呼、着呼、転送の自動連続動作
  - RTP音声再生（waveファイル再生）
  - RTPVideo再生（回線キャプチャデータの再生）
  - SDP、ヘッダの編集方法を任意に設定可能
  - 複数の擬似SIP端末から一斉に発呼動作開始するための仕組み

店舗サーバ



チェックイン、チェックアウト、オーダ、会計、卓管理、などの
店舗業務を運用する店舗サーバをSituatorアプリケーションで実現

ファイル転送モジュール

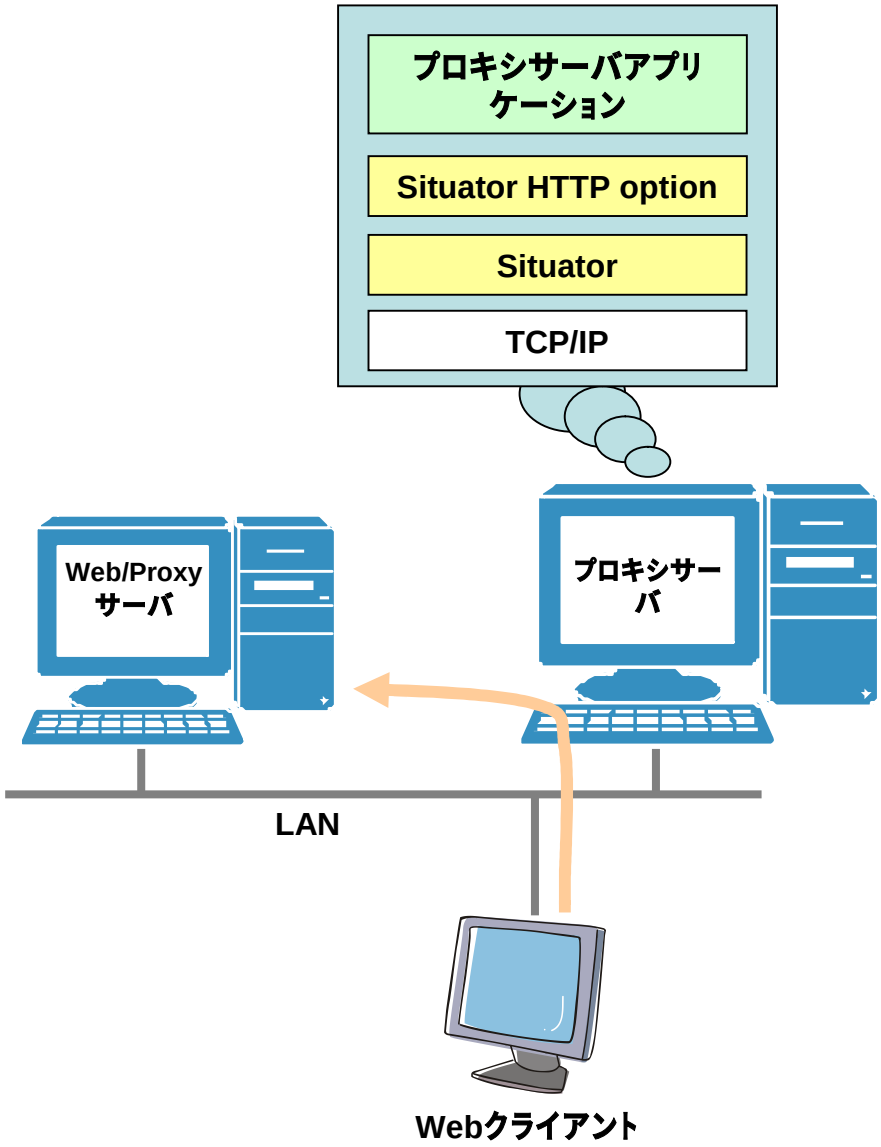


- Situatorコンポーネントを利用して
  ファイル転送機能を実現
  - 付加機能の追加が可能
    - 暗号化
    - 圧縮
    - チェックサム

    など
  - システムの保守･運用ツールとして
    - ジャーナル収集
    - ログ収集
    - ファイル配布

    など

プロキシサーバ



- Situatorコンポーネントを利用してWebプロキシサーバを実現
  - カスタム機能を組み込み可能
    - 認証
    - アクセス先の監視
    - HTML書き換え
    - リクエスト振り分け

    など
  - Webサーバとして振舞うことも可能
    - Webアプリケーションサーバ
    - テスト用対向サーバ

    など

まとめ

◆通信ミドルウェアとして

- マルチプラットホームに組み込んで利用可能
- 各種通信オプションも利用可能

短期間に高品質のアプリケーションを開発可能

◆テストツールとして

- 準備の難しい装置の変わりに擬似装置として活躍
- 発生させ難い異常シーケンス試験、レグレッション試験の環境を容易に構築

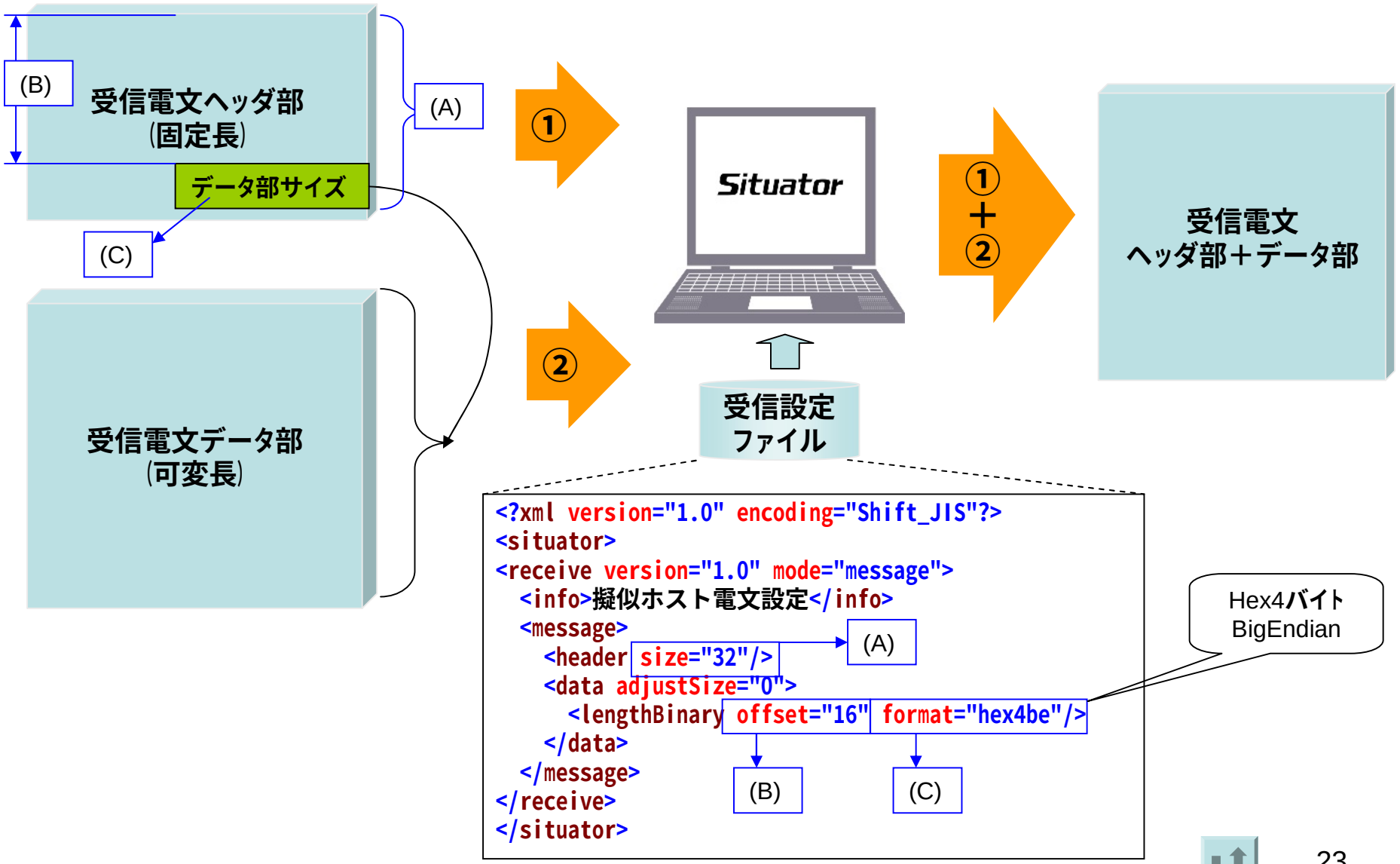
生産性向上、品質確保、保守効率の実現

スクリプト例

(Webサーバに接続してHEADリクエストを送信するJScript)

```
//HEADリクエストを格納するデータオブジェクトを作成
var request = new ActiveXObject("Situator.DataSet");
request.AddString("HEAD http://www.ots.co.jp/ HTTP/1.0¥r¥n¥r¥n");

//TCPソケットオブジェクトを作成
var s = Application.TcpSockets.Create();

//Webサーバに接続
s.Connect("www.ots.co.jp:80");

//HEADリクエストを送信
s.Send(request);

//レスポンス受信は自動で
```

受信設定ファイル



シナリオファイル例

```
<?xml version="1.0" encoding="utf-8" ?>
- <situator>
  - <scenario messageLevel="1" name="scenario1" showdump="true" version="1.0">
      <!-- 送信データ(HTTPリクエスト)を変数var1に設定  -->
    - <var name="var1" src="" type="DataSet">
        <![CDATA[ "GET / HTTP/1.0\r\n\r\n"  ]]>
      </var>
      <!-- TCP接続する  -->
    - <tcpClient local="" name="tcpClient1" recvMode="" remote="www.ots.co.jp:80"
        retryCount="0" retrySleep="0">
        <onsuccess />
      - <onerror>
          <exit code="0" />
        </onerror>
      </tcpClient>
      <!-- HTTPリクエストを送信する  -->
    - <send data="var1" for="tcpClient1" to="">
        <edit />
        <onsuccess />
      - <onerror>
          <exit code="0" />
        </onerror>
      </send>
      <!-- イベント待ちループ  -->
    - <loop count="-1">
      - <waitEvent timeout="-1">
          <!-- データ受信イベント  -->
          <onreceive for="tcpClient1" />
          <!-- 切断イベント  -->
        - <onclose for="tcpClient1">
            <!-- ループ終了  -->
            <exit code="0" />
          </onclose>
        </waitEvent>
      </loop>
    </scenario>
  </situator>
```

HTTPリクエストを変数に設定する

TCP接続を行う

データ送信を行う
変数に設定したHTTPリクエストを送信する

イベント待ち合わせのループを行う

イベント待ち合わせを行う

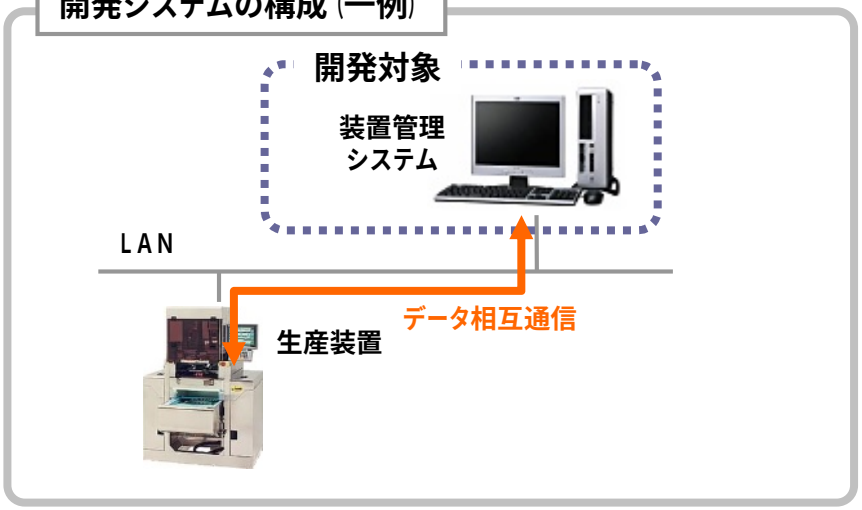
データ受信時に呼び出される処理

切断を受信時に呼び出される処理

シナリオを終了する

擬似対向テストツールを使う場面

開発システムの構成（一例）



ケース①「対向装置を入手できない」

ケース②「対向装置も平行開発」

ケース③「異常系のテストが困難」