トップエスイー修了制作

SOA時代における通信プログラムのモデル検査手法の確立

茂呂　範

開発における問題点

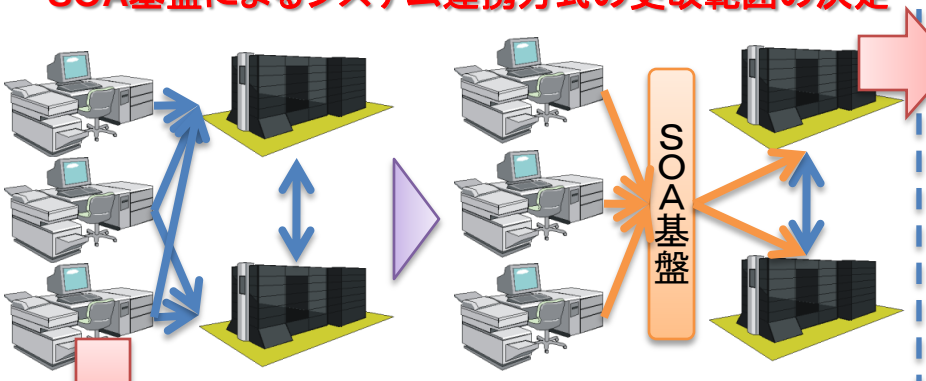
SOAにてシステム連携を行う場合、現行システムの通信手順が順守されること、また新規連携システム追加時に現行連携システムへの影響を与えないことが求められる。複数種の品質の異なる通信プロトコル間での連携にて、現行の通信品質を担保するために、上流工程から設計の正しさを検証できる手法が求められる。

手法・ツールの提案による解決

現行システムの運用設計より、障害時のシステム間のデータ不整合発生状況を、目標とすべき現行通信の品質モデルと定義する。新規に追加される各種通信プロトコルと通信手順を異常系を含めてモデル化し、それらを組み合わせた場合においても現行通信の品質モデルを担保できるかを、SPINおよびJPFを用いて検証する。

連携方式が現行通信品質を担保するか判定する作業フロー

SOA基盤によるシステム連携方式の更改範囲の決定



現行システムの調査

障害時の通信保障とデータ不整合の調査

・電文ロストは発生するか？
・電文多重送信に対する対処は？

通信プロトコル
連携手順
異常系処理設計
障害時復旧手順

通信保障の品質モデルとして定義

項番	連携システム	連携種別	障害種別	連携元検知	連携先検知	連携元状態	連携先状態
1	連携元：在庫管理	オンライン連携	正常系	なし	なし	更新	更新
2	連携先：ホスト		連携元送信経路異常	送信エラー	なし	取消	未処理
3	GW：SNA-TCP変換		連携先送信経路異常	異常応答受信	なし	取消	未処理
4			連携先受信経路異常	異常応答受信	応答エラー	取消	取消
5			連携元受信経路異常	受信エラー	なし	取消	更新
6			受付前中継異常	送信エラー	なし	取消	未処理
7			送信前中継異常	タイムアウト	なし	取消	未処理
8			受信前中継異常	タイムアウト	応答エラー	取消	取消
9			応答前中継異常	タイムアウト	なし	取消	更新
10			応答後中継異常	なし	なし	更新	更新
11		ディレード連携	正常系	なし	なし	更新	更新
12			連携元送信経路異常	送信エラー	なし	取消	未処理
13			連携先送信経路異常	なし	なし	更新	未処理
14			受付前中継異常	送信エラー	なし	取消	未処理
15			送信前中継異常	なし	なし	更新	未処理
16			送信後中継異常	なし	なし	更新	更新

新規連携方式の設計

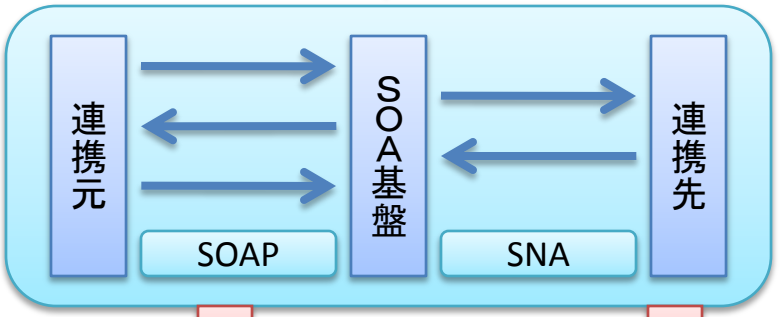
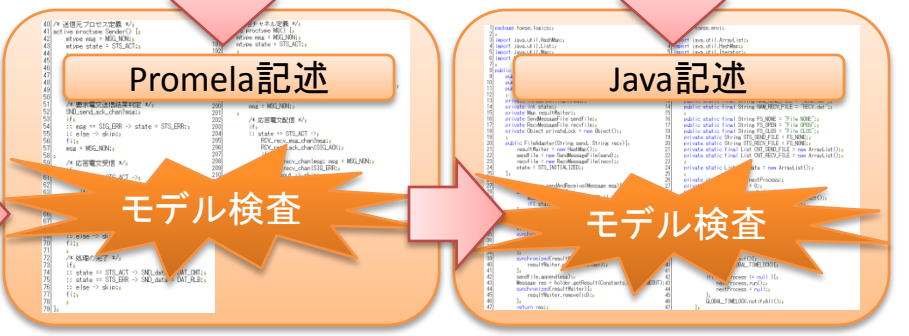
通信プロトコルの異常系を含めたモデル化

項番	通信プロトコル	タイミング	連携元	通信路	連携先	発生事象
1	SOAP	送信前	ダウン	—	未受信	なし
2	BASE: HTTP		HTTP/コネクションエラー TCP/SYNACK-TO	切断	未受信	連携元：送信時エラー
3	BASE: TCP/IP		HTTP/コネクションエラー TCP/SYNACK-TO	—	ダウン	連携元：送信時エラー
4	FROM: Linux 2.6	REQ送信中	ダウン	—	TCP/TO, HTTP/EOF	連携先：受信時タイムアウト後に電文エラー
5	TO: Linux 2.6		通信エラー TCP/ACK-TO	切断	TCP/KeepAlive-TO HTTP/EOFエラー	
6			通信エラー TCP/ACK-TO	—	ダウン	
7		REQ送信後	ダウン	—	HTTP/送信エラー TCP/ACK-TO	
8			HTTP/EOFエラー TCP/KeepAlive-TO	切断	HTTP/送信エラー TCP/ACK-TO	
9			HTTP/EOFエラー TCP/KeepAlive-TO	—	ダウン	
10		RES受信中	ダウン	—	HTTP/送信エラー TCP/ACK-TO	連携先：結果送信中エラー
11			HTTP/EOFエラー TCP/KeepAlive-TO	切断	HTTP/送信エラー TCP/ACK-TO	
12			HTTP/EOFエラー TCP/KeepAlive-TO	—	ダウン	

プロトコル仕様
実装依存部
識者ノウハウ

電文フロー設計
トランザクション設計
異常系処理設計
通信保障処理設計

連携方式を設計モデルとして定義



SPINとJPFを用いたモデル検査