CSPを用いたシーケンス図検証ツールの適用事例

2010/11/20

○ 海津智宏（北陸先端大）
磯部祥尚（産総研）
鈴木正人（北陸先端大）

http://www.topse.jp/

目次

- 背景・目的
- 提案手法の説明
- デモ
- 関連研究・まとめ

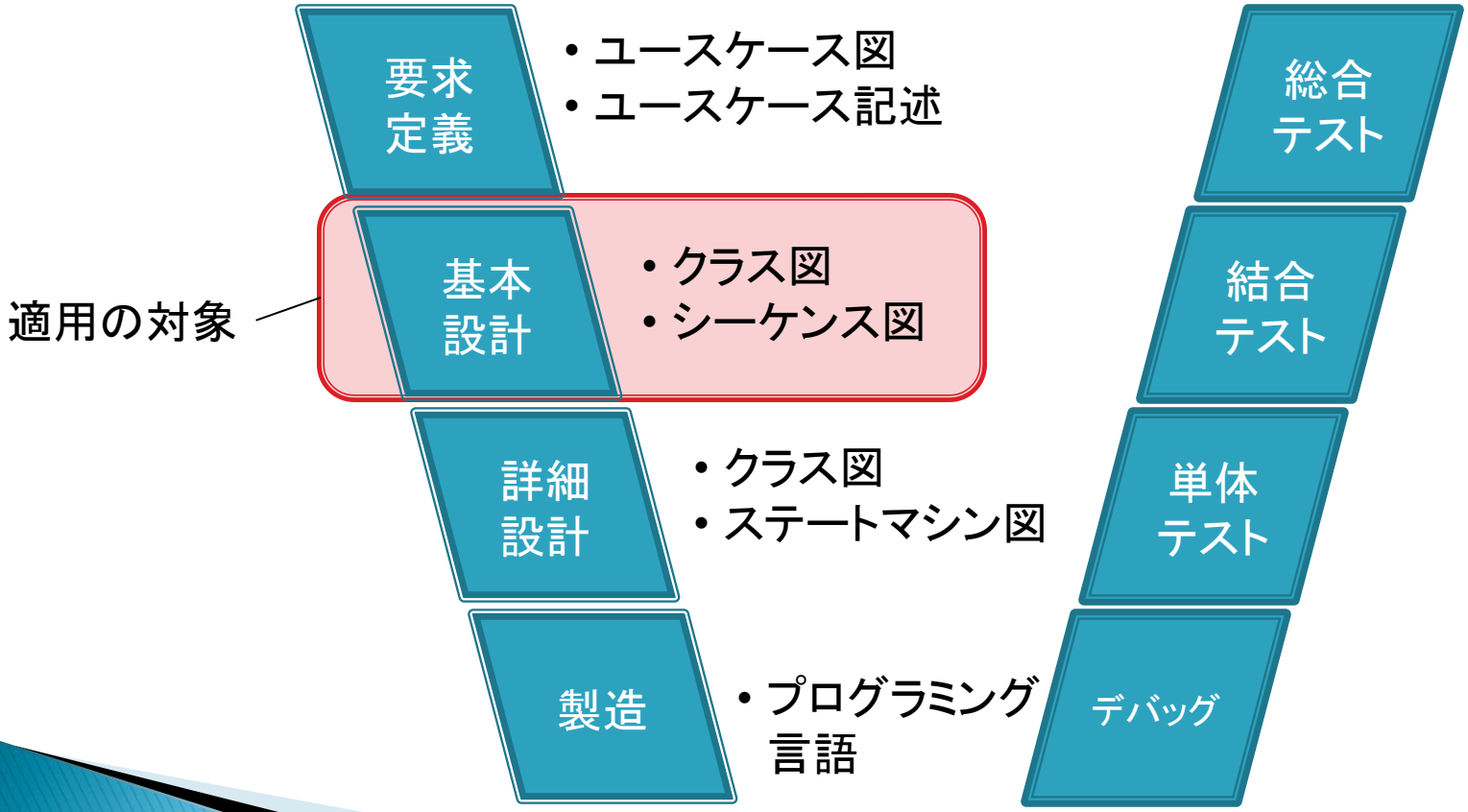
背景・目的

- ソフトウェアの複雑化、多様化
- ソフトウェアの信頼性はますます求められている
- 限られた納期・工数

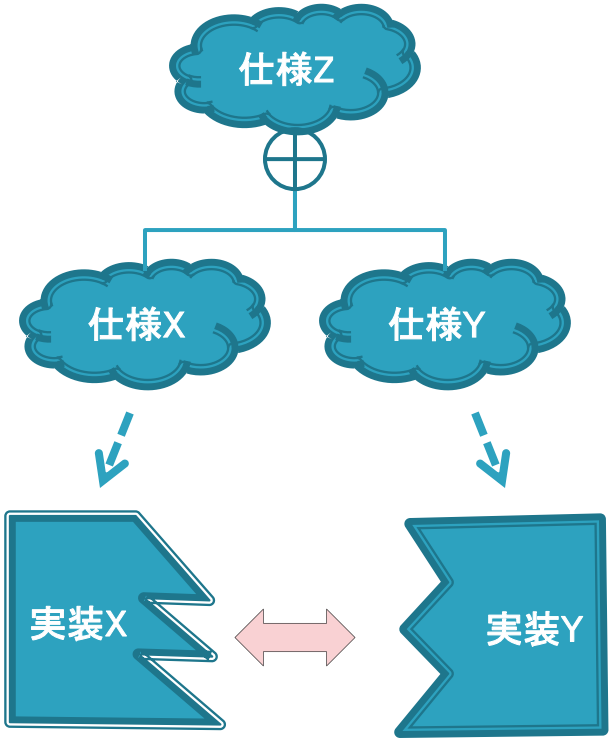
- 問題点の早期発見により、バグの発見漏れや手戻り工数の増大を抑えることが重要に

モジュール化を前提とした開発方法に形式手法を適用することで、品質の向上・手戻り工数の削減を目指す

対象とする開発プロセス



基本設計工程の検証のアプローチ

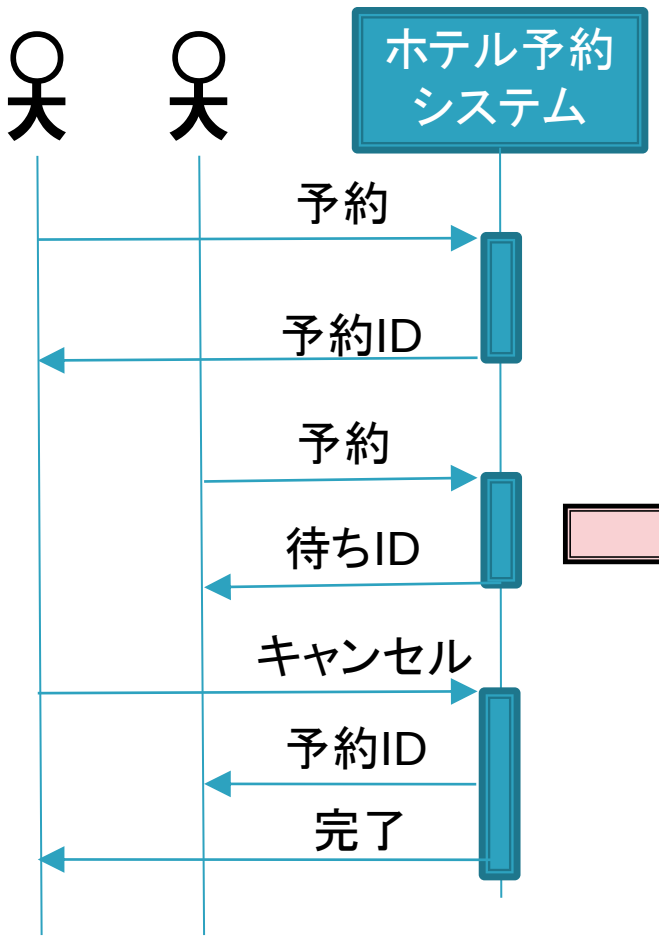


モジュール間の不整合が結合テストで発覚すると、手戻り工数が大きい

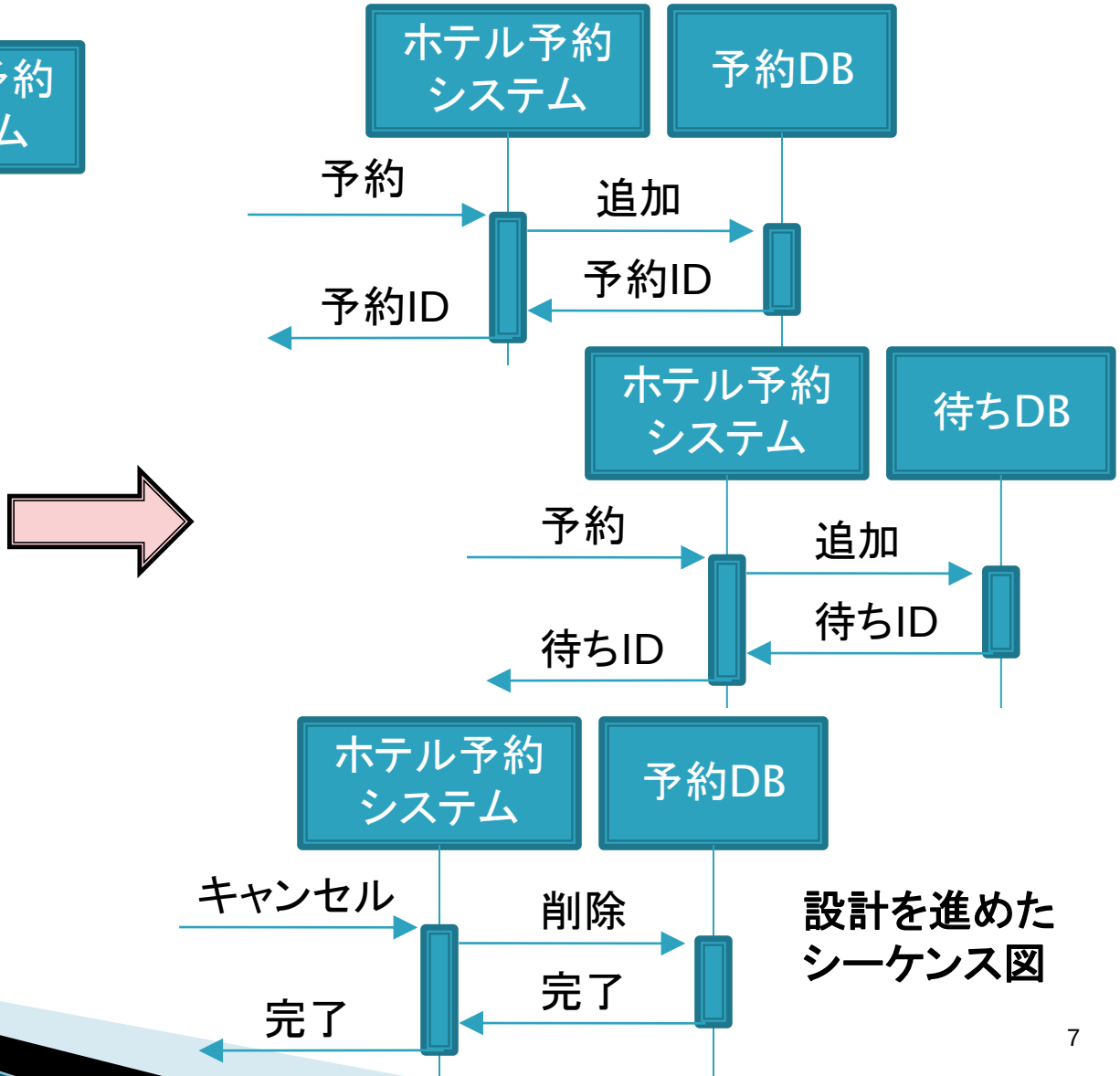
- 品質向上、手戻り工数の削減のためには、検証等によるバグの早期発見が効果的。
- しかし、形式的な仕様記述のモデルを直接記述するのはコストが大きい。
- 現実の開発現場では、準形式的な記法の中ではUMLのシーケンス図の利用率が高い。

設計工程初期によく使われるシーケンス図を検証することはできないか？

シーケンス図の誤りの例 (1 / 2)

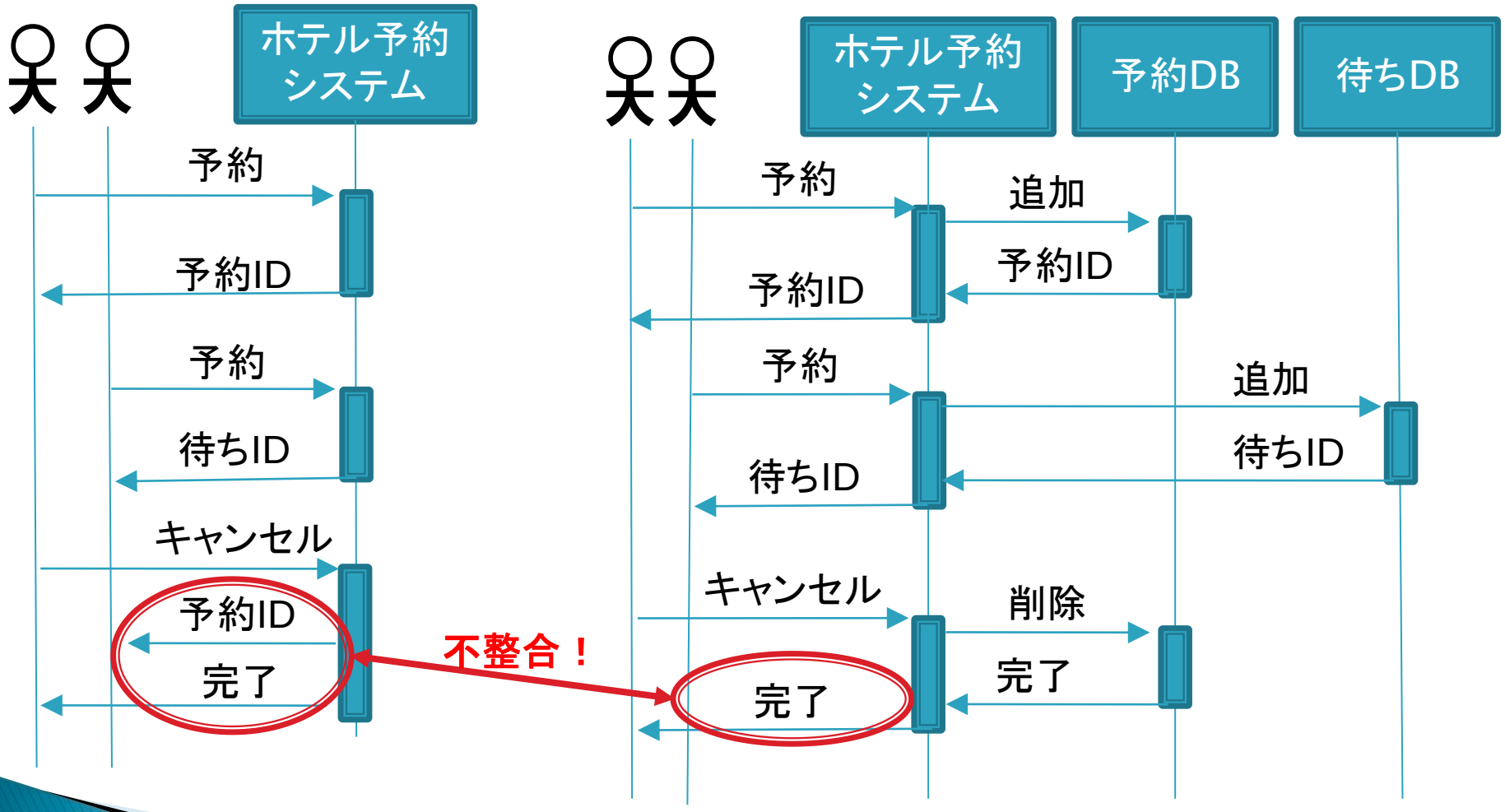


実行可能であるべき
シーケンス図



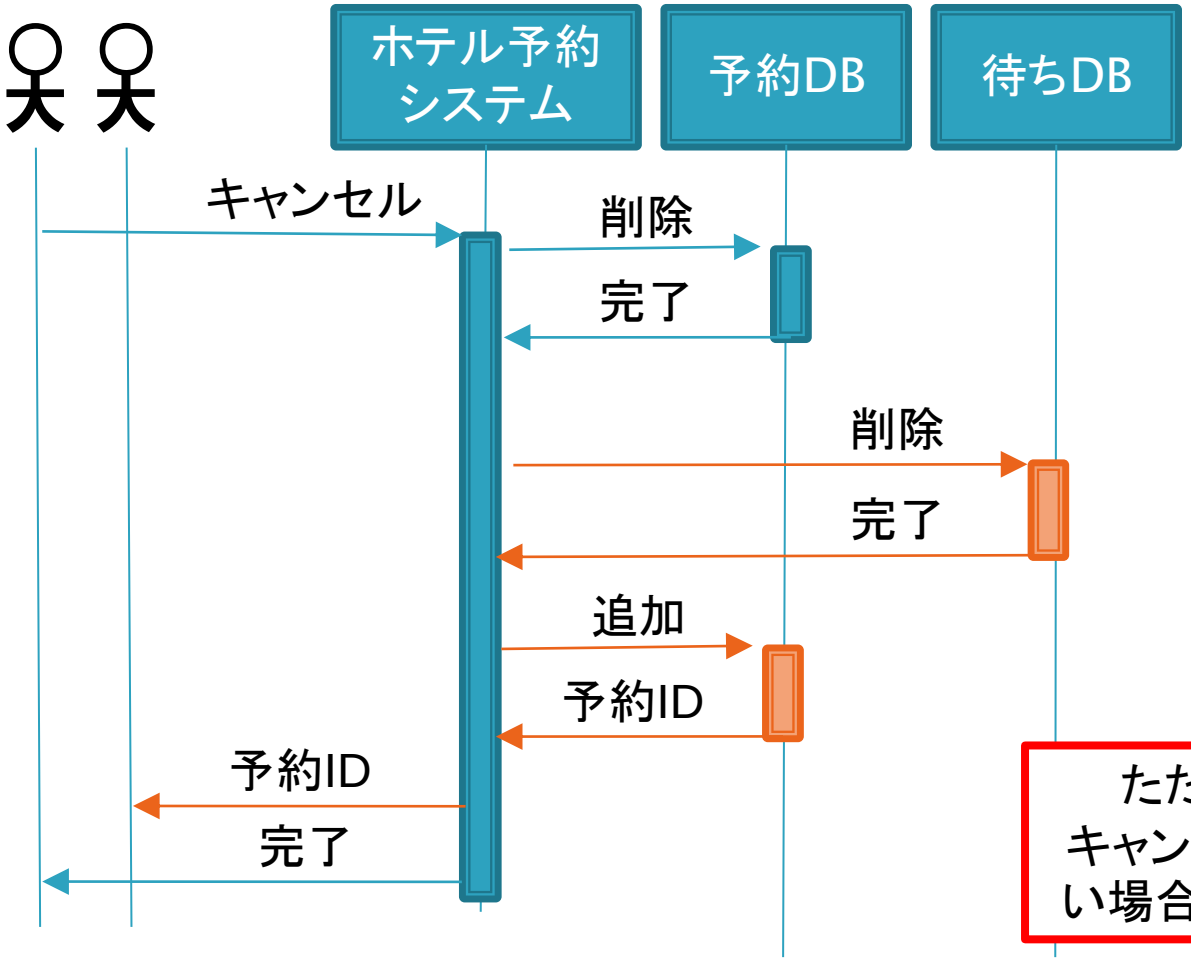
設計を進めた
シーケンス図

シーケンス図の誤りの例 (2/2)



修正した設計 (1/2)

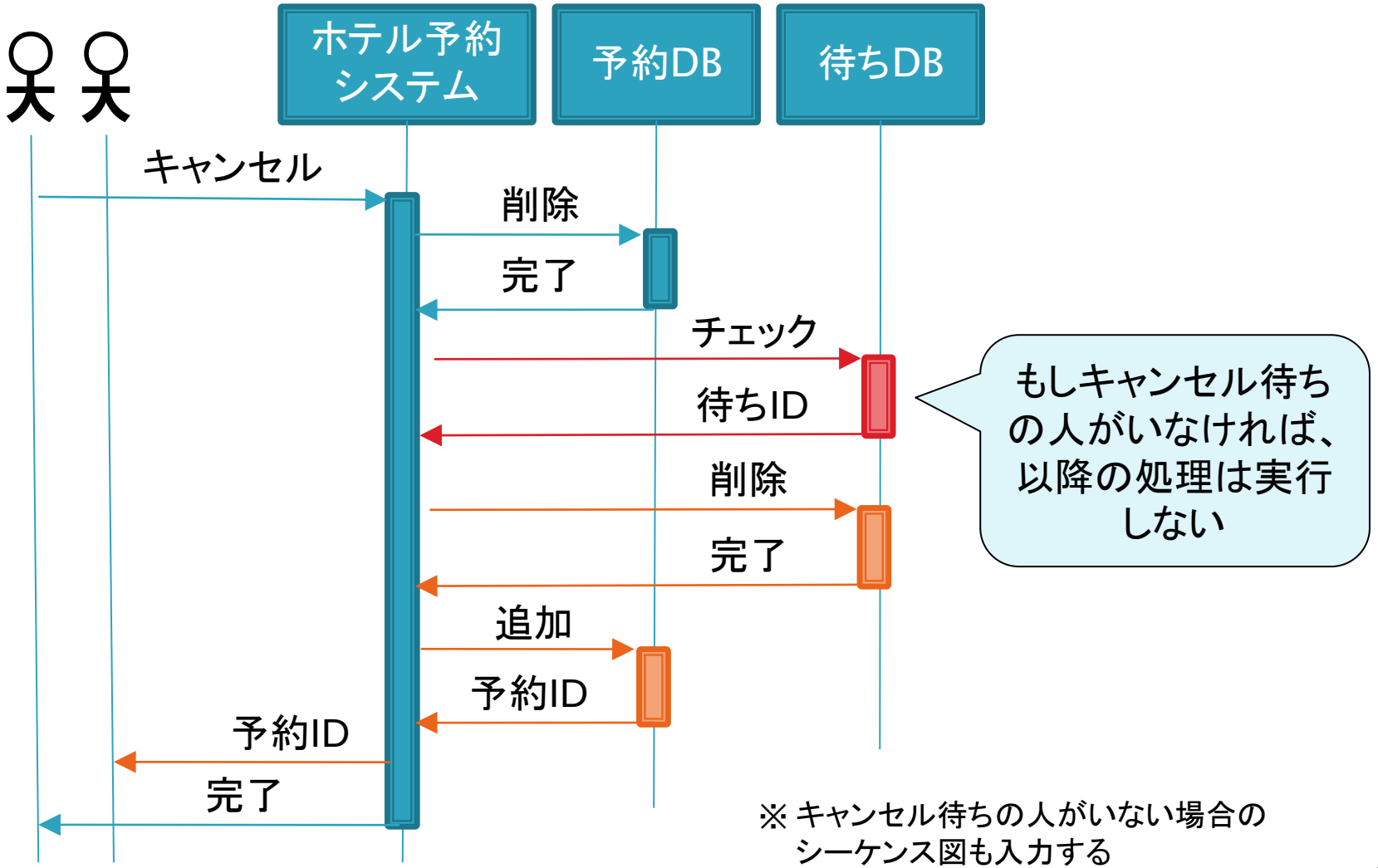
予約キャンセル時に、キャンセル待ちユーザを予約に追加し、該当ユーザに予約IDを通知するように修正



ただし、この設計では
キャンセル待ちの人がいない場合が考慮されていない

修正した設計 (2/2)

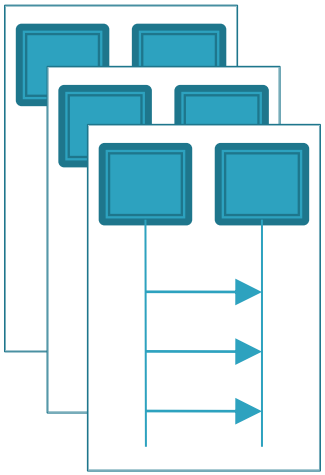
待ちDBの情報の削除前に、キャンセル待ちの有無をチェックするよう修正



※ キャンセル待ちの人がいない場合のシーケンス図も入力する

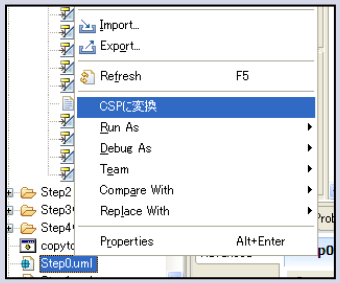
提案手法の概要

検証用モデルとしては CSP を採用。複数のシーケンス図を元に状態遷移モデルを生成する。モデル検査ツールにより詳細化関係を検証可能。

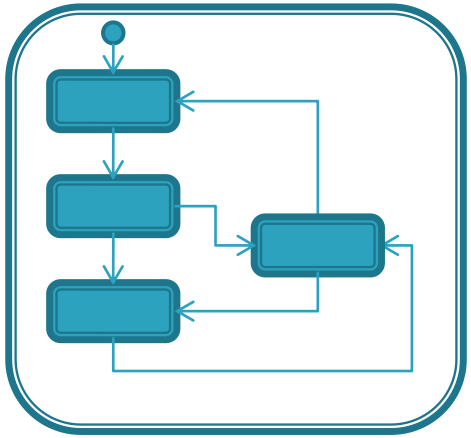


シーケンス図

合成



合成方法の提案とツール化
(SD2CSP)



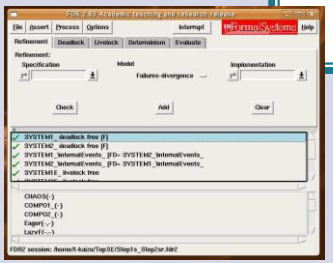
状態遷移（CSPモデル）

検証

検証

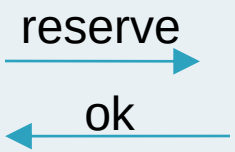
詳細化前の状態遷移

制約



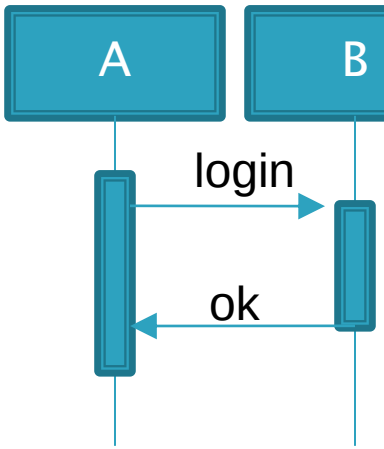
既存ツール利用
（FDR2）

シーケンス図とCSPとの対応

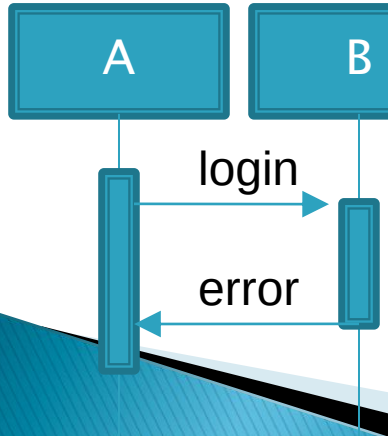
シーケンス図	CSP
オブジェクト 	**プロセス** COMPO_(WDB.default_) = …
状態不変式 	**プロセスの状態** COMPO_(WDB.waiting) = …
活性区間 	**プロセスの動作** （活性区間単位で複数の動作を合成）
メッセージ送受信 	**イベント** call_.reserve.From.To -> call_.ok.To.From -> …

シーケンス図の合成

CSP の「内部選択」という概念を活用し、メッセージの選択における送信と受信の違いを区別して合成する



A = login! → ok? → A
B = login? → ok! → B



A = login! → error? → A
B = login? → error! → B

合成

A は、ok と error のどちらでも受信できる（□）

決定的

A = login! →
((ok? → A) □ (error? → A))

B = login? →
((ok! → B) ⊓ (error! → B))

CSP モデル

B は、ok と error のどちらかを選択して送信する（⊓）

非決定的

※「!」は送信、「?」は受信を表す

シーケンス図合成演算子・の導入

以下の性質を満たすような新しい演算子を定義した。

- 合成する各シーケンス図で同じイベントが発生する場合、遷移先は1つの状態となる
  - $(\alpha \rightarrow P) \cdot (\alpha \rightarrow Q) =_F (\alpha \rightarrow (P \cdot Q))$
- メッセージに複数の選択肢がある場合、送信側は内部選択、受信側は外部選択となる
  - $(a! \rightarrow P) \cdot (b! \rightarrow Q) =_F (a! \rightarrow P) \sqcap (b! \rightarrow Q)$
  - $(a? \rightarrow P) \cdot (b? \rightarrow Q) =_F (a? \rightarrow P) \,\square\, (b? \rightarrow Q)$
  - $(a! \rightarrow P) \cdot (b? \rightarrow Q) =_F (a! \rightarrow P) \triangleright (b? \rightarrow Q)$
- 詳細化関係を保存する
  - P’ が P を詳細化したもので、Q’ が Q を詳細化したものであれば、P’・Q’ も P・Q を詳細化したものとなっている

※「$P \triangleright Q$」は「$(P \,\square\, Q) \sqcap Q$」の略記

デモ（ホテル予約システム）

予約

チェックイン

キャンセル

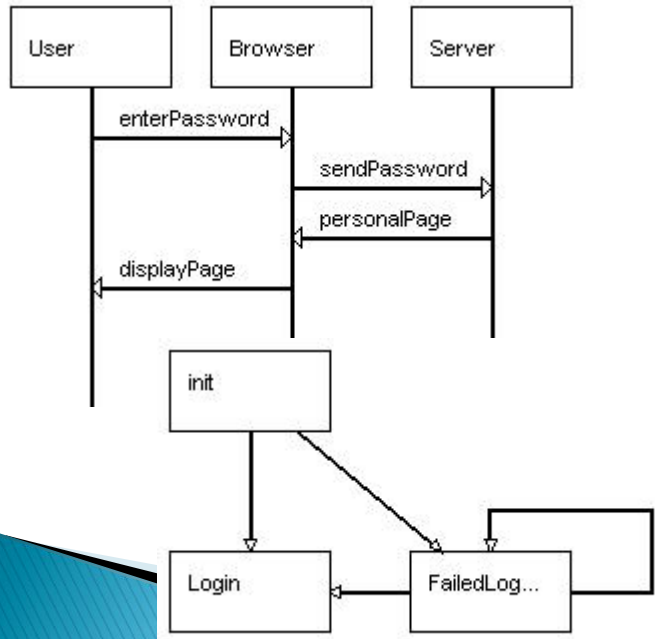
キャンセル待ちのキャンセル

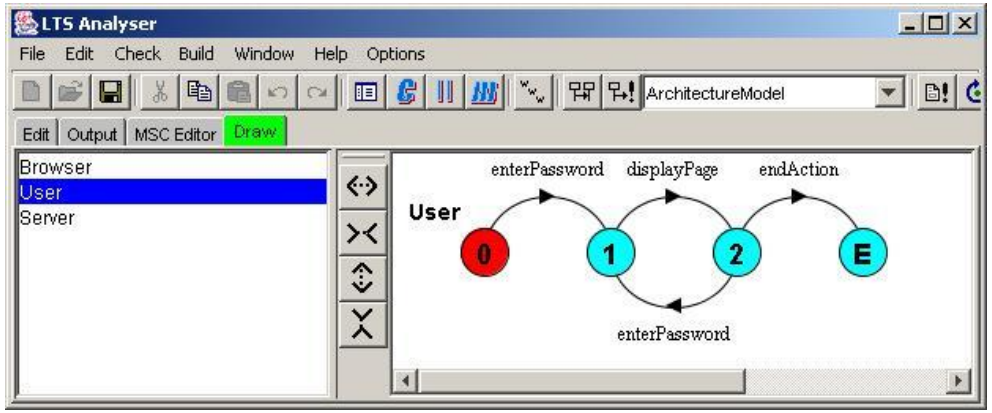
ホテル予約システム

- 利用者はホテルの部屋を予約できる
- ホテルが満室の場合、キャンセル待ちとなる
  - デモのシーケンス図では、部屋数 1 部屋、キャンセル待ち最大 1 人に簡略化してモデル化
- 予約のキャンセル時、キャンセル待ちの人がいればその人が予約中となる
- 内部の実装では、予約は “RDB”, キャンセル待ちは “WDB” で管理。“Controller” は情報を持たず、RDB および WDB に処理を振り分ける

関連研究: LTSA-MSC

- シーケンスチャート間の順序関係を記述したHMSC というモデルを入力。HMSCに従ってLTSAモデルを生成できる。
  - → シーケンスチャートと並行してHMSCの作成が必要





※図は http://www.doc.ic.ac.uk/ltsa/msc/tutorial/ より

まとめ

- シーケンス図からCSPプロセスを合成し、検証する手法を提案した。
  - 合成を表す新しい演算子を導入した。
    - シーケンス図合成方法を形式的に議論できるようになった。
    - 詳細化関係を保存するので、部分的な詳細化を検証できる。
- 本提案手法を実現するツール SD2CSP を開発した。
  - Eclipse プラグインとして実装することで、Eclipse上でシーケンス図を記述・変換可能とした。
  - サンプルプロジェクトに適用し、詳細化の間違いを発見できることを確認した。
- 今後の課題として、例外系など定型的なシーケンスの自動補完、反例解析の支援などを検討している。