

プロセス代数CSPによる シーケンス図設計の詳細化と検証

2010/01/21

○ 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)

目次

- ▶ 背景・目的
- ▶ 提案手法の説明
- ▶ 検証の例
- ▶ まとめ・今後の課題

背景・目的

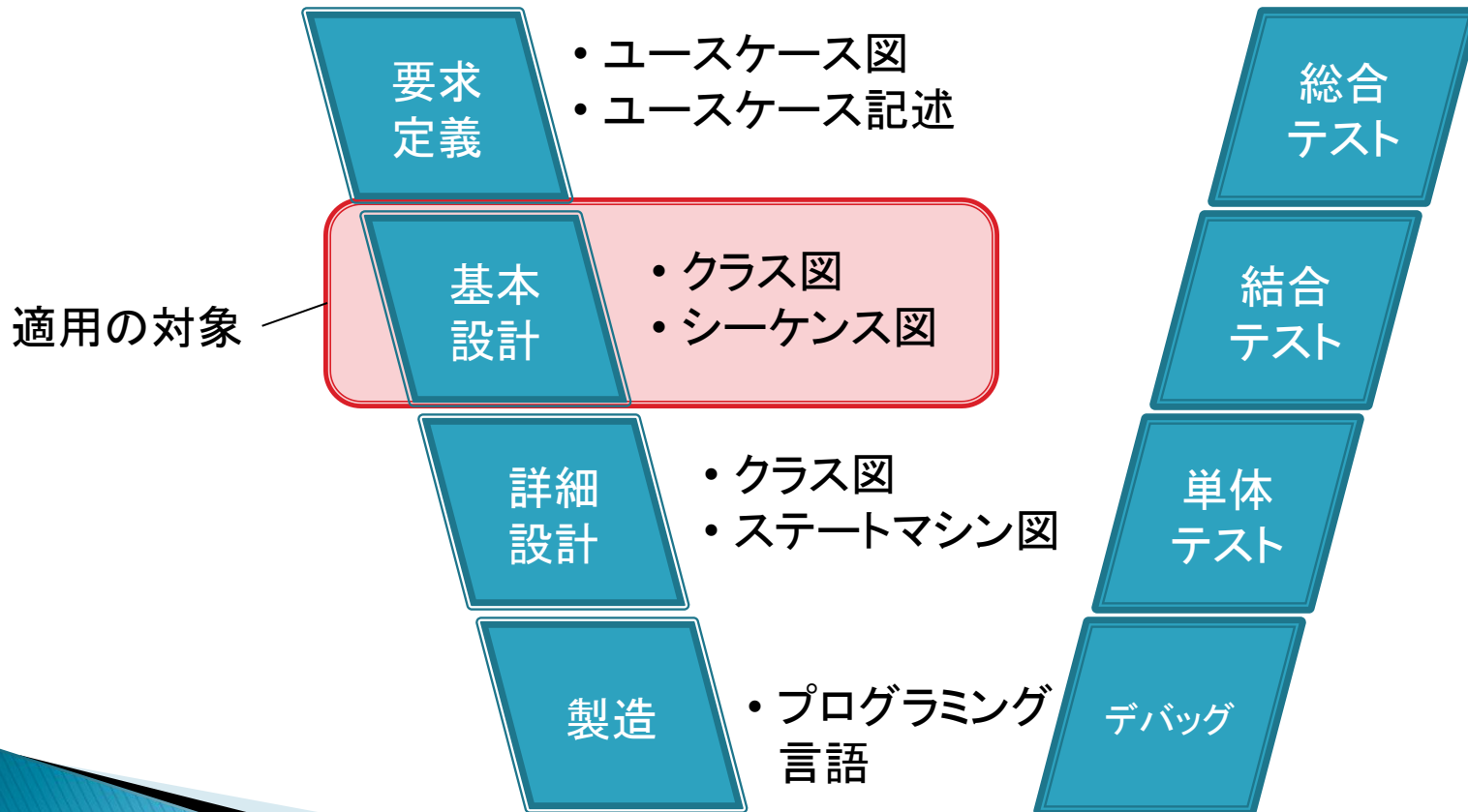
- ▶ ソフトウェアの複雑化、多様化
- ▶ ソフトウェアの信頼性はますます求められている
- ▶ 限られた納期・工数



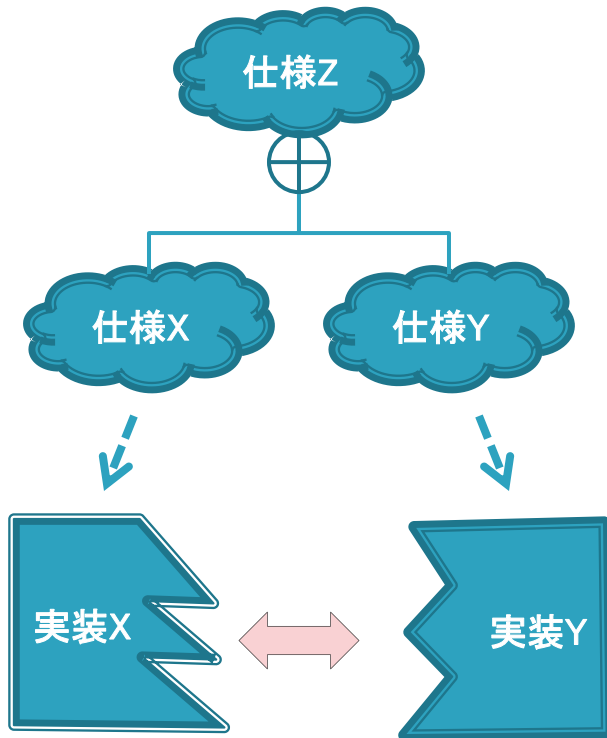
- ▶ 問題点の早期発見により、バグの発見漏れや手戻り工数の増大を抑えることが重要に

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



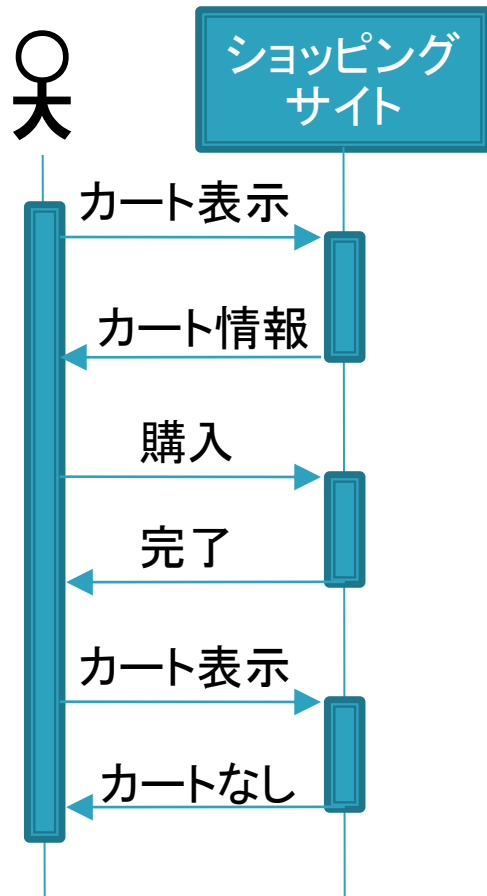
モジュール間の不整合が結合テストで発覚すると、手戻り工数が多い

- ▶ 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。

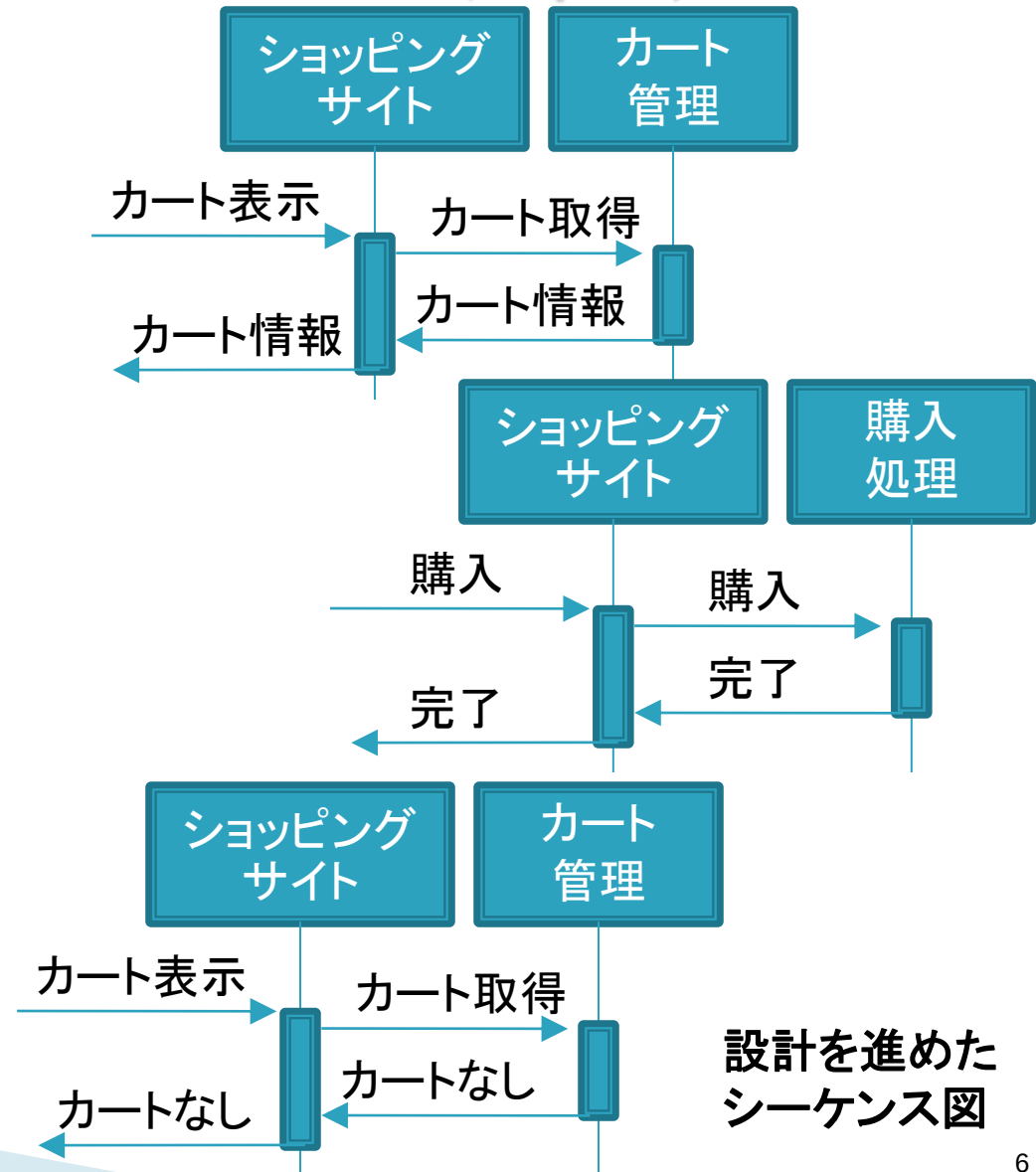
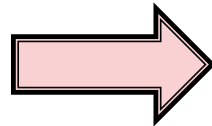


設計工程初期によく使われるシーケンス図を検証することはできないか？

シーケンス図の誤りの例 (1 / 2)

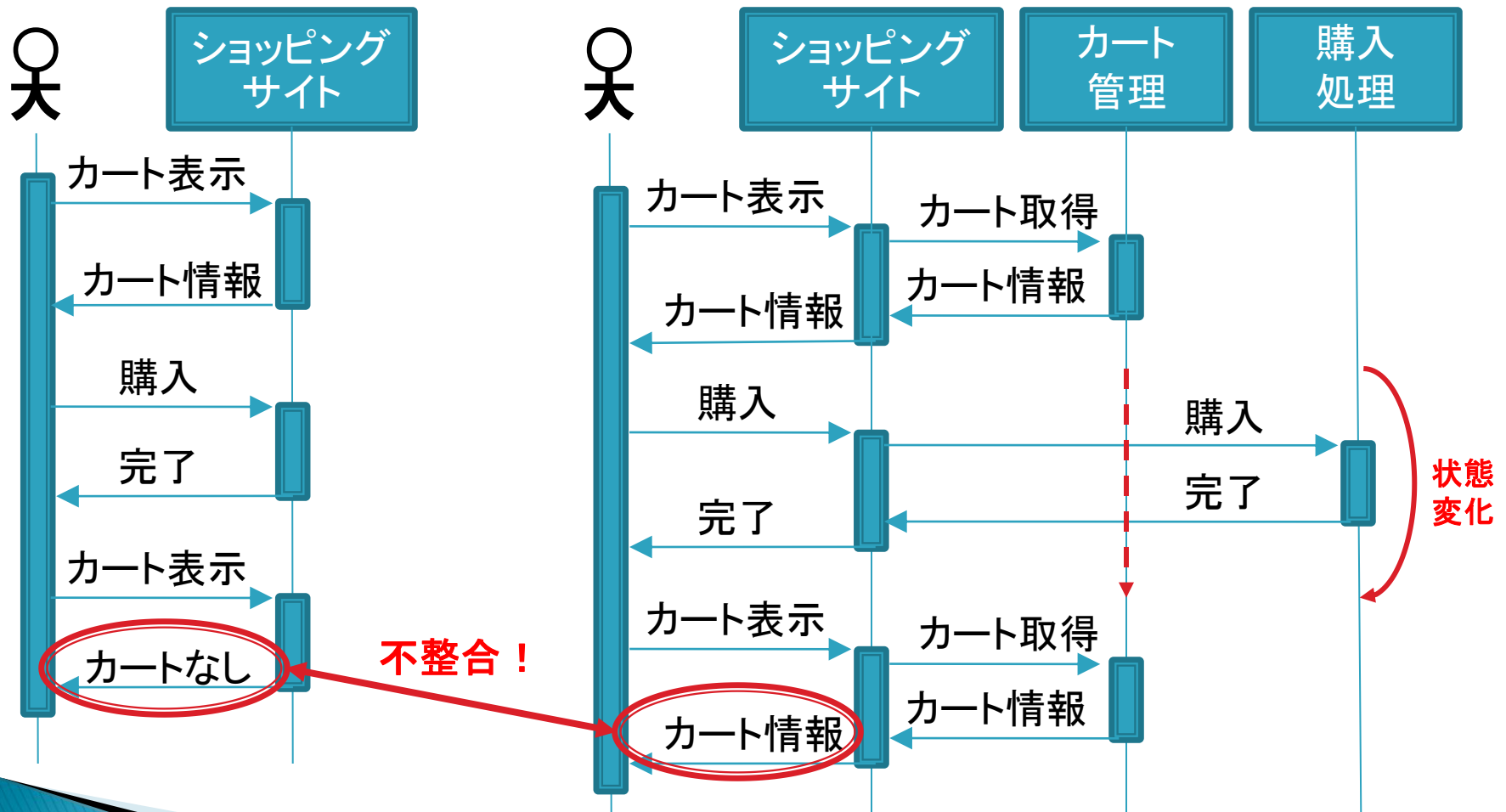


実行可能であるべき
シーケンス図

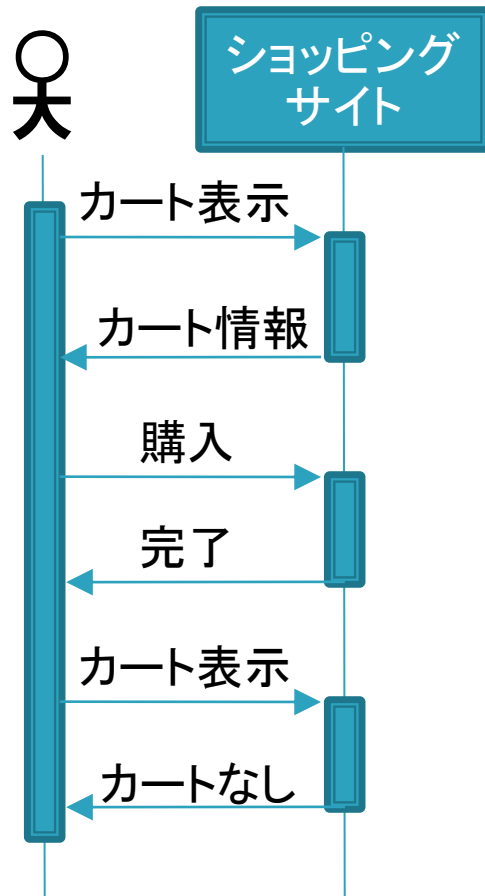


設計を進めた
シーケンス図

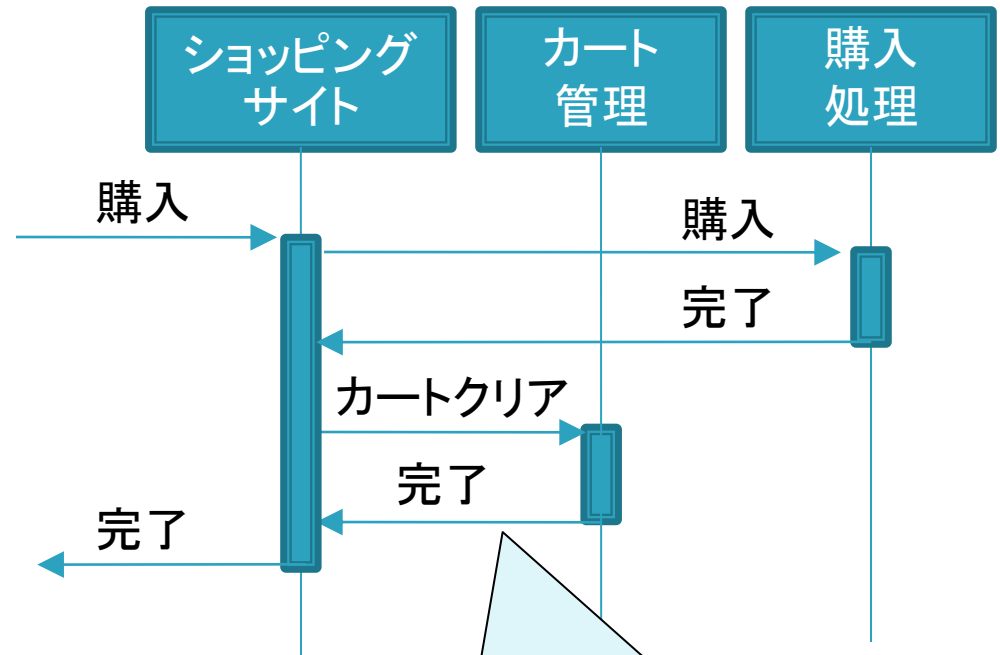
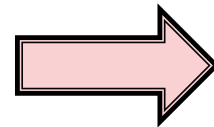
シーケンス図の誤りの例 (2/2)



正しい設計



実行可能であるべき
シーケンス図

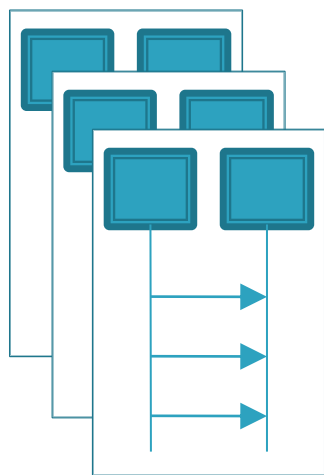


購入処理後に「カート表示」の結果が変わるので、カート管理コンポーネントの処理を呼び出す必要がある

設計を進めた
シーケンス図

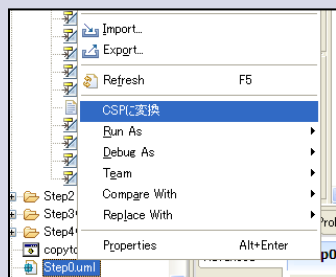
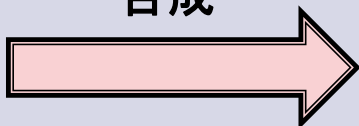
提案手法の概要

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。

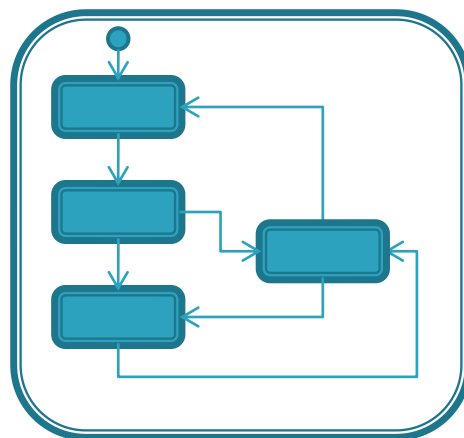


シーケンス図

合成



合成方法の
提案とツール化
(SD2CSP)

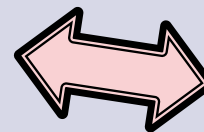


状態遷移 (CSPモデル)

検証

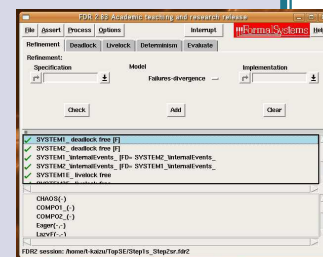


検証



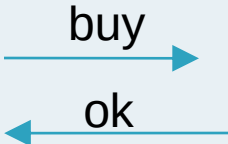
詳細化
前の状態遷移

制約



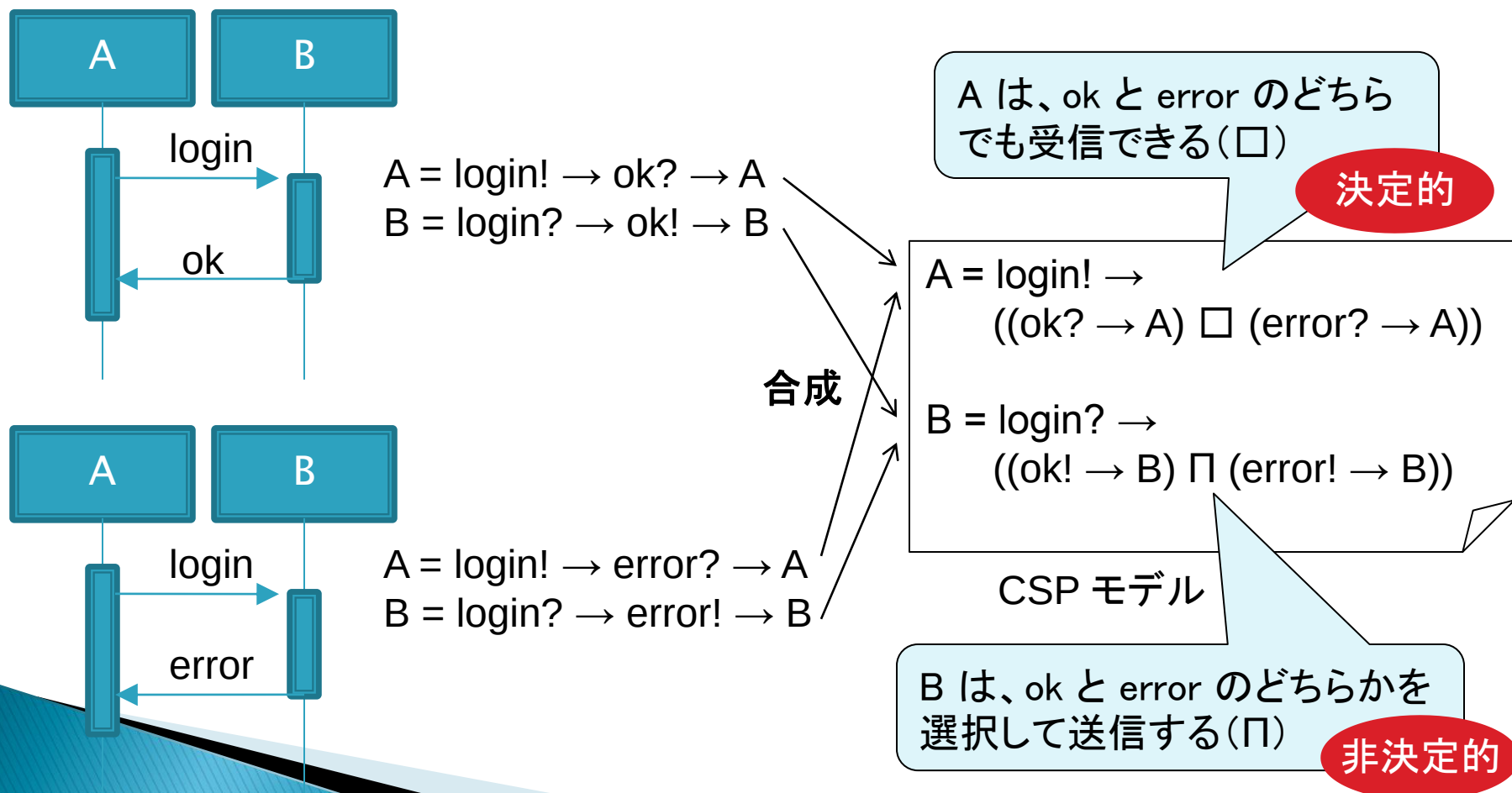
既存ツール利用
(FDR2)

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_(MarketSite.default_) = ...
状態不変式 	プロセスの状態 COMPO_(MarketSite.haveCart) = ...
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_.buy.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

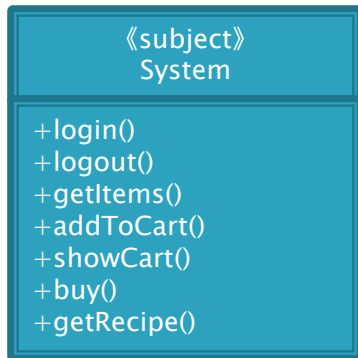


シーケンス図合成演算子・の導入

以下の性質を満たすような新しい演算子を定義した。

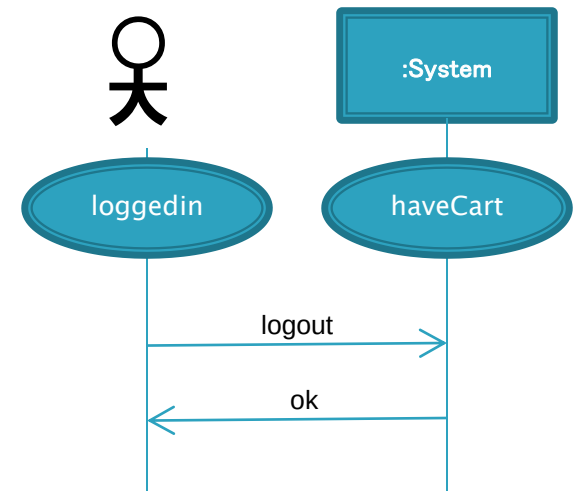
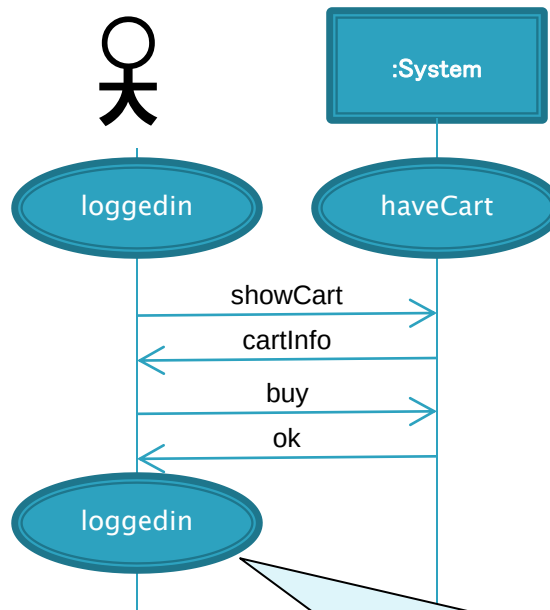
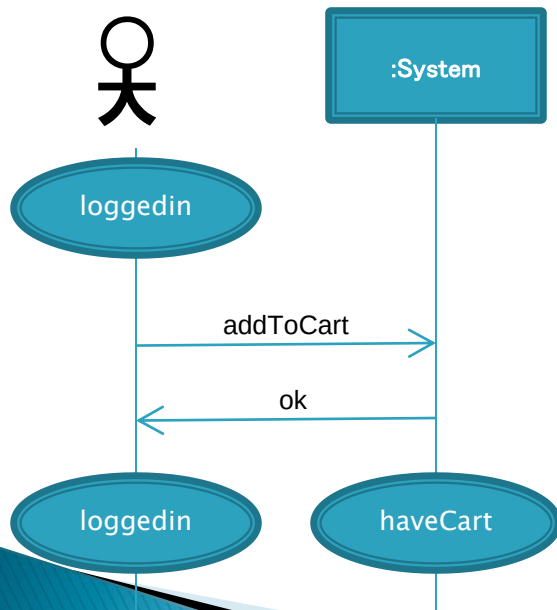
- ▶ 合成する各シーケンス図で同じイベントが発生する場合、遷移先は1つの状態となる
 - $(\alpha \rightarrow P) \circ (\alpha \rightarrow Q) =_F (\alpha \rightarrow (P \circ Q))$
- ▶ メッセージに複数の選択肢がある場合、送信側は内部選択、受信側は外部選択となる
 - $(a! \rightarrow P) \circ (b! \rightarrow Q) =_F (a! \rightarrow P) \sqcap (b! \rightarrow Q)$
 - $(a? \rightarrow P)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

検証の例 (1 / 5) - 詳細化前のシーケンス図



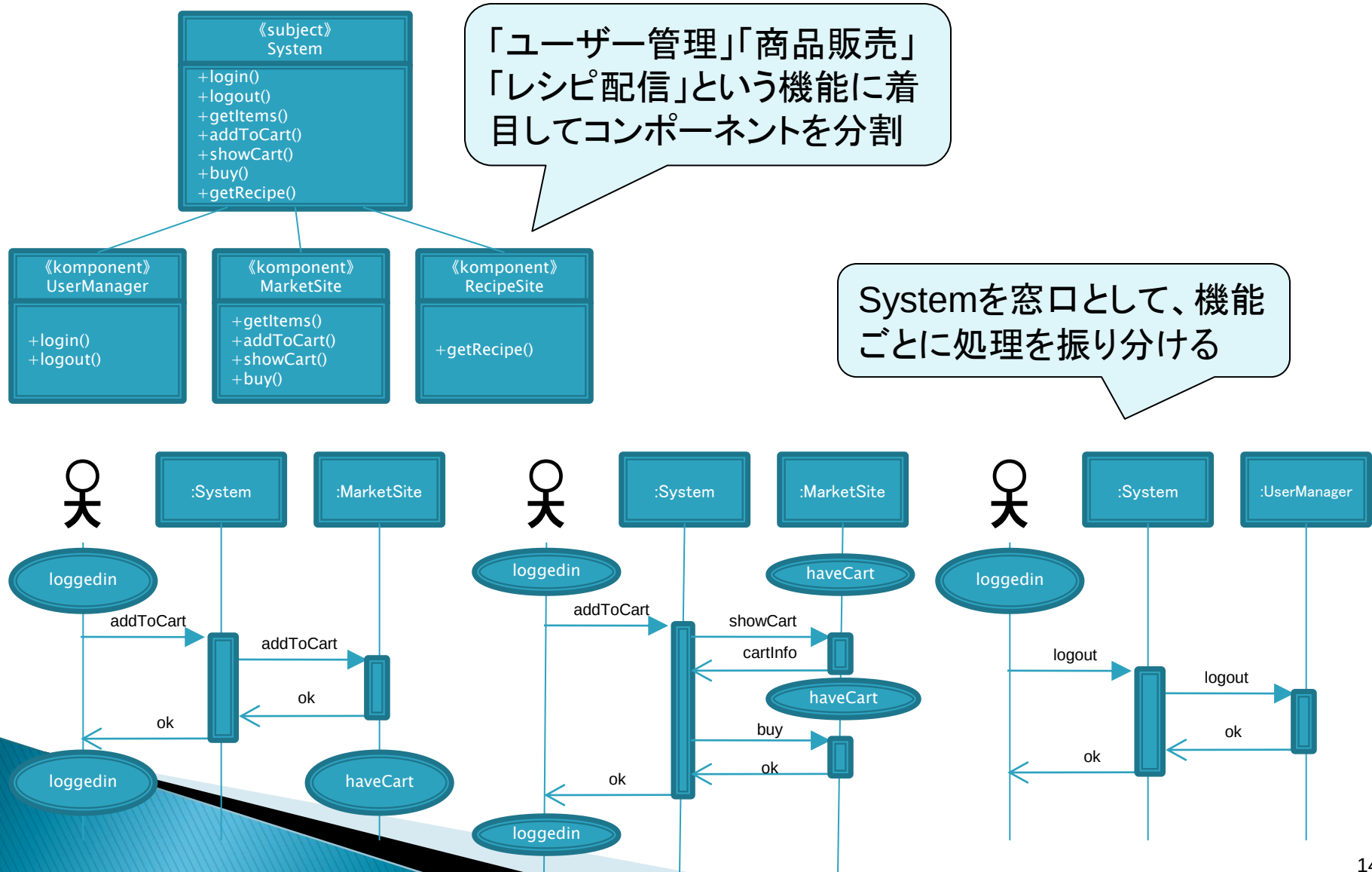
単一の「System」コンポーネントで
全機能を提供

ユースケースごとに、ユースケース
記述相当のシーケンス図を用意

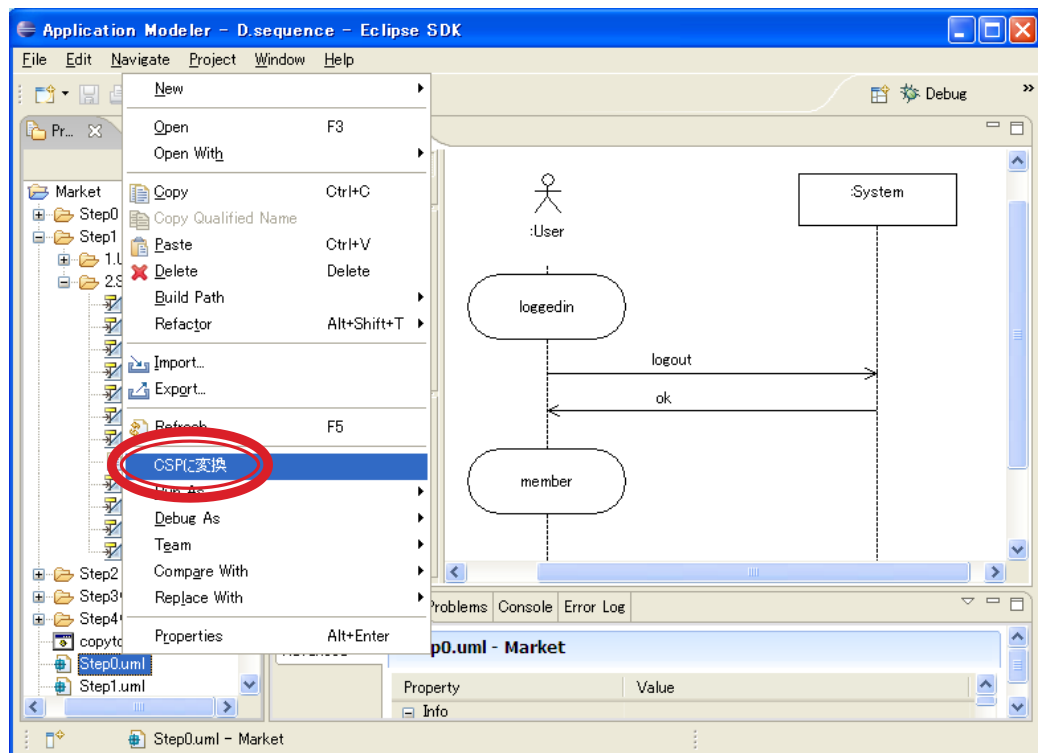


事前条件・事後条件を状態として記述

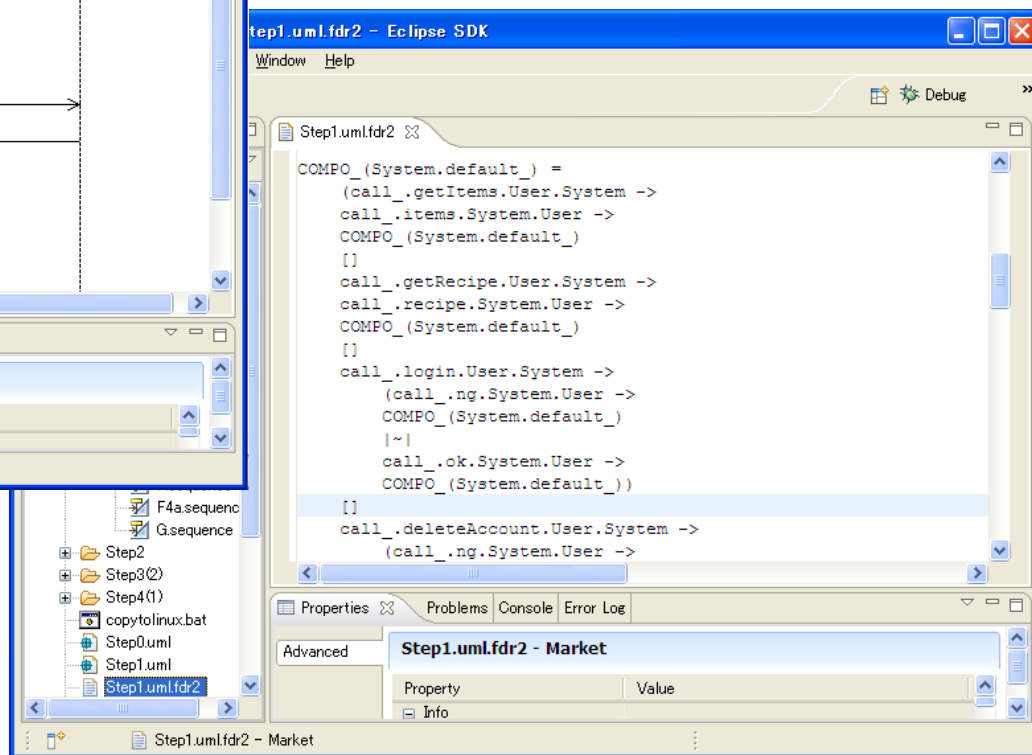
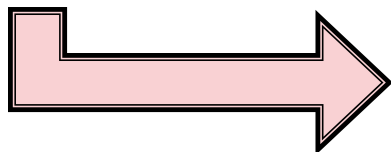
検証の例 (2 / 5) - 詳細化後のシーケンス図



検証の例 (3 / 5) – CSPモデルの生成

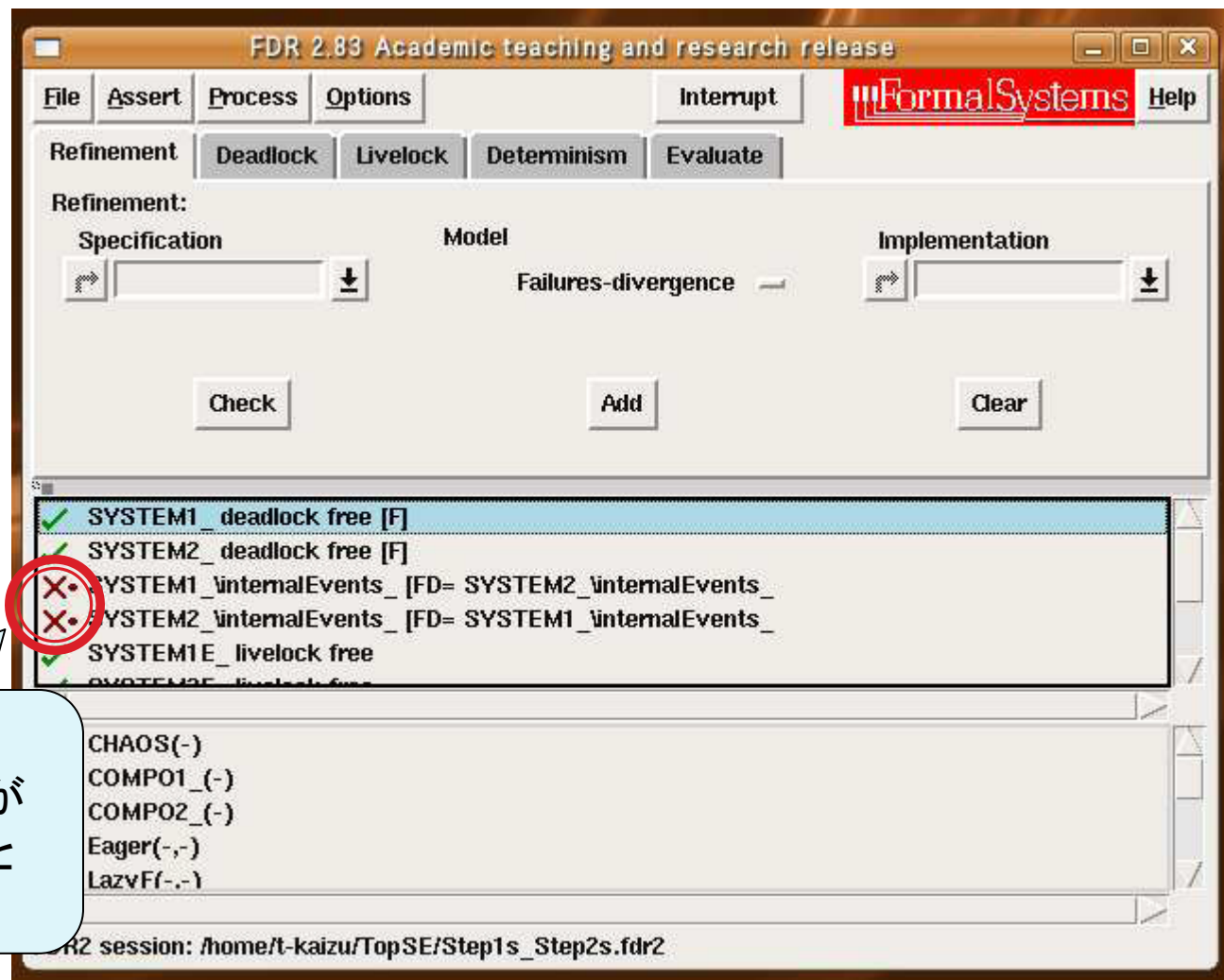


Eclipse プラグインとして実装した変換ツール
(SD2CSP)



生成されるCSPモデル

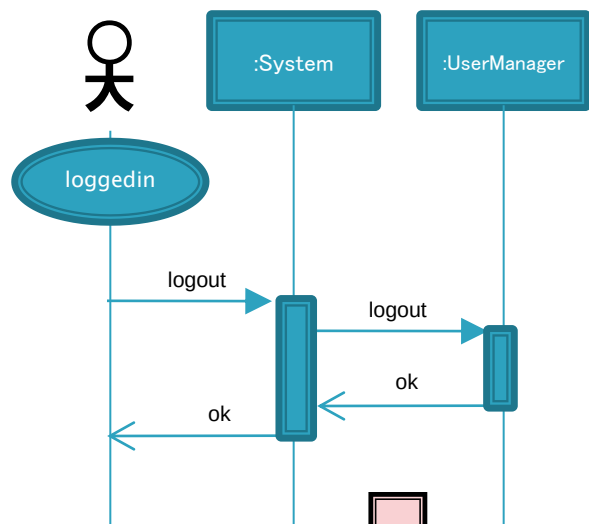
検証の例 (4 / 5) – 検証



検証失敗

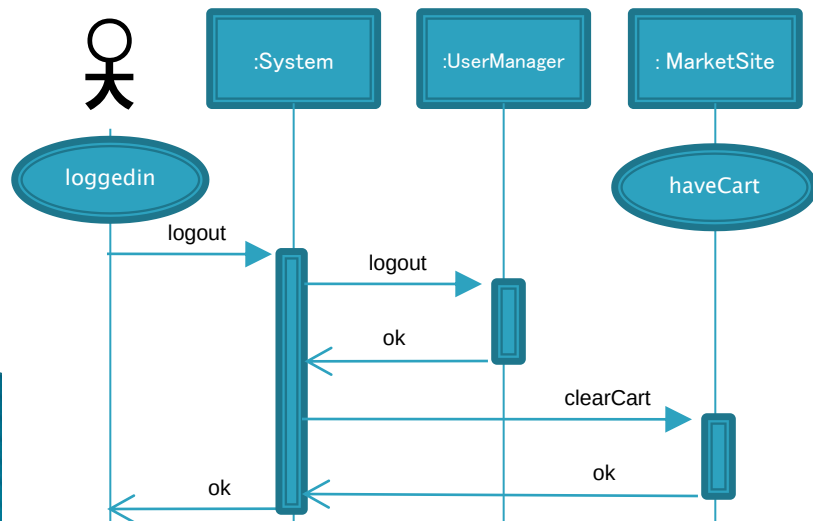
詳細化後のシーケンスが
詳細化前のシーケンスと
整合していない

検証の例 (5 / 5) – 問題点の修正

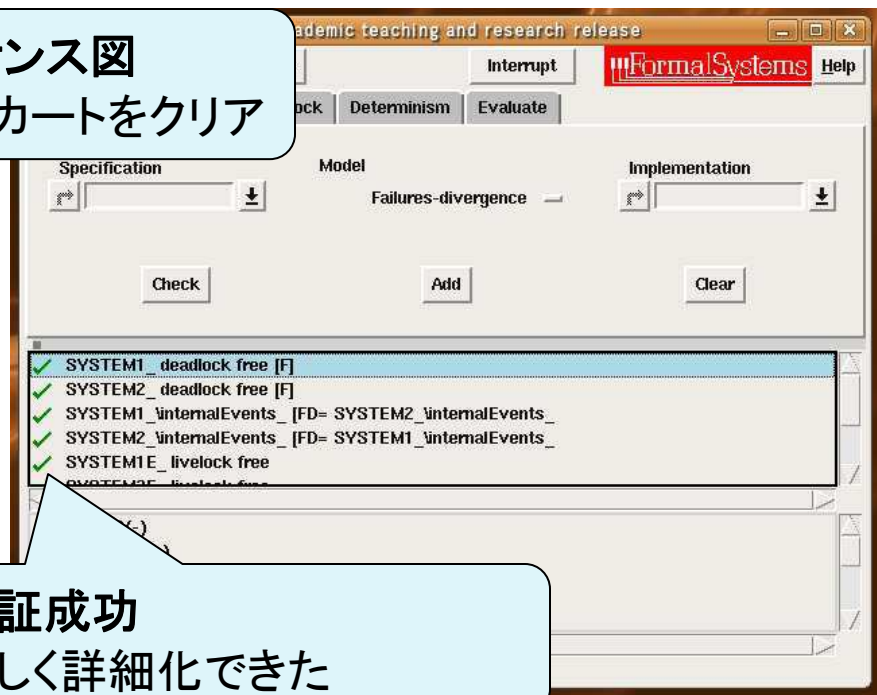


誤りのあったシーケンス図
ログアウト時にカートの情報
を更新しない

修正したシーケンス図
ログアウト時にカートをクリア

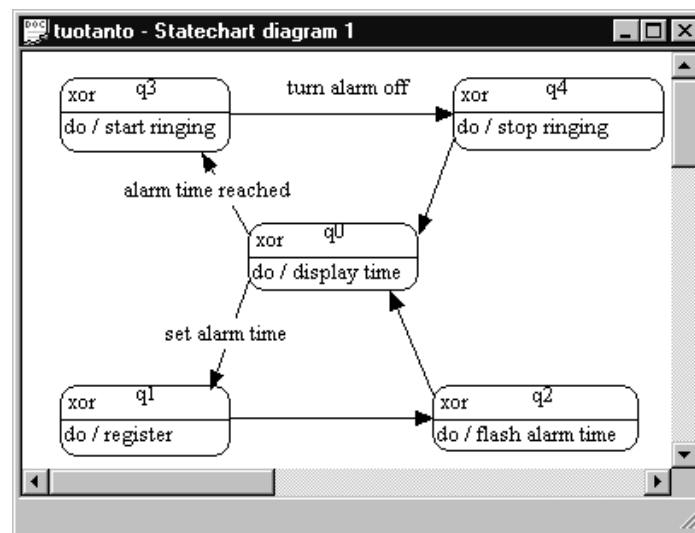
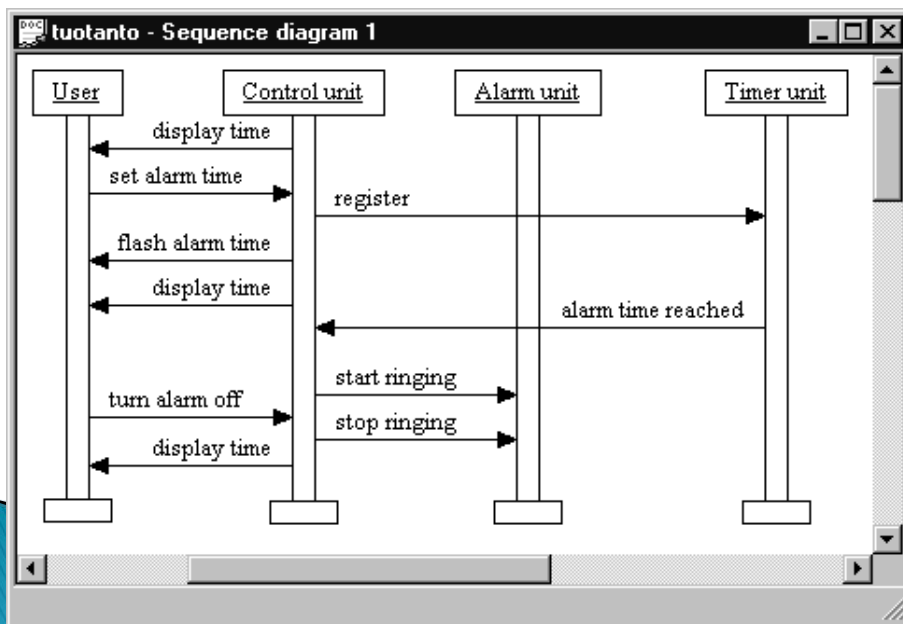


検証成功
正しく詳細化できた



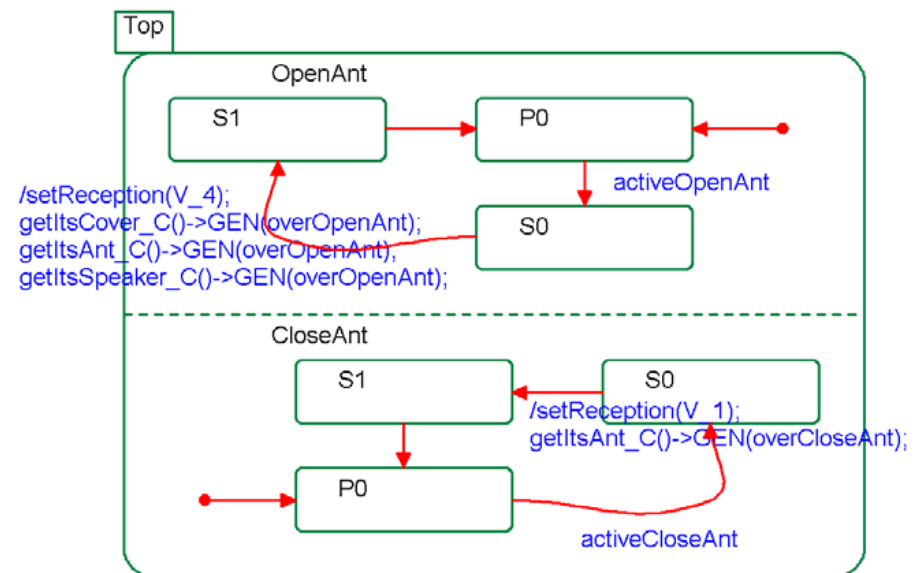
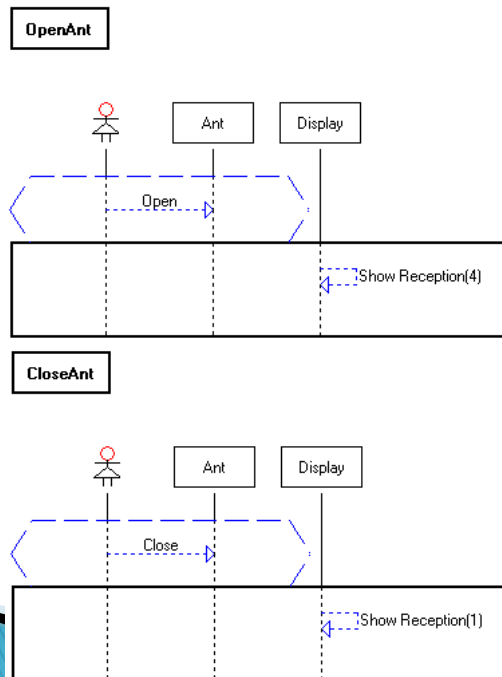
関連研究 (1 / 3) MAS

- ▶ 送信イベントを状態に、受信イベントを遷移の条件に対応させる。
- ▶ 非決定性が含まれない範囲で、状態数を最小化する。
 - → 本研究で想定しているような非決定性を含むモデルは実現不可能



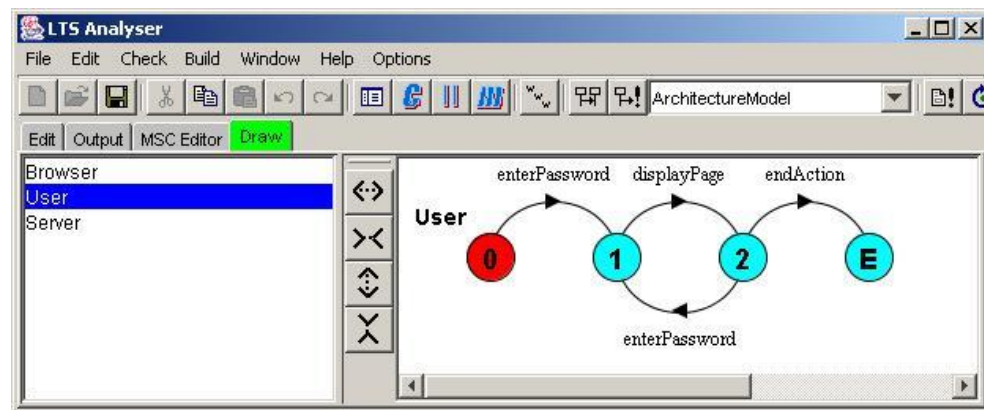
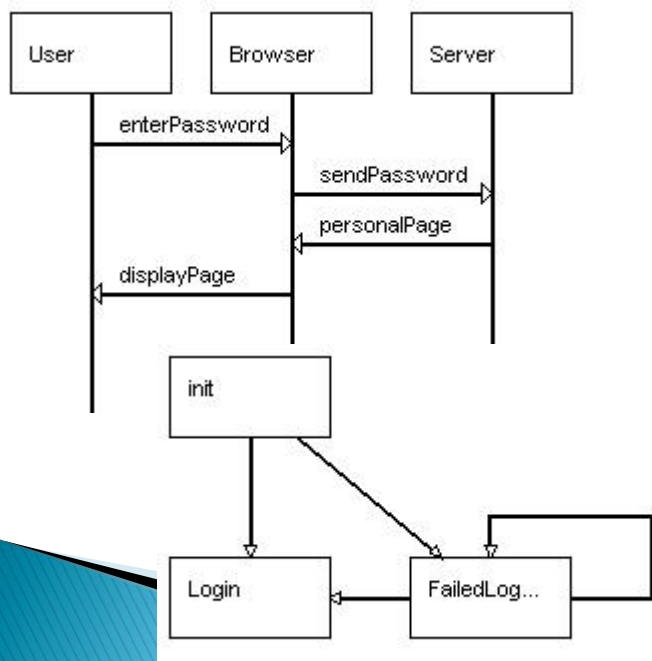
関連研究 (2 / 3) Live Sequence Chart

- ▶ 条件に一致する場合に、必ず特定のシナリオが発生することを記述。
 - → 正確な条件を洗い出す工程をシナリオ作成に含む
- ▶ シナリオを元に、動作するコードを生成可能。



関連研究 (3 / 3) 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



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

- ▶ 合成演算子。の性質をさらに明らかにする
 - より一般的な変換規則の証明など
- ▶ 合成アルゴリズムの改善
 - 状態不変式が記述されていない場合でも、途中の状態からのシーケンス図を適切に合成
- ▶ シーケンス図記述の省力化
 - 異常系などで記述が省略されている場合に、似た箇所の記述をもとにシナリオを補完する方法を検討
- ▶ 反例解析の支援
 - モデル検査の結果をシーケンス図にフィードバック