# 構造化日本語仕様としての
# VDM仕様

## エクスプレス予約での
## 悲惨な体験モデル

佐原伸

(株)CSK　ITソリューション社

# 発端

- モバイルSuica（JR東）でエクスプレス予約（JR東海）を使っていた
- ロンロンViewカードが廃止になりアトレクラブViewカードに変更して下さいとの連絡があった
- JR東海サポートセンターに電話した
  - モバイルSuicaの登録クレジットカードを変更すれば、2日後にはエクスプレス予約が使えますとのこと
  - モバイルSuicaでクレジットカードを変更し、2日後に使おうとしたが、エクスプレス予約できなかった
    - 「予約できなかったんですが」「カードを変更したら、エクスプレス予約を新規に契約して下さい」「新規にすると会費がかかるのでは？」「はい」「カード会社の都合で変更するのに、それはおかしいでしょう？」「では、無料にします」
    - 「カード変更前に予約したe特急券に引換えようとしたらできないのですが？」「エクスプレス予約会員証で引換できるようになったので、それで引換えて下さい。暗証番号はクレジットカードのものを使って下さい」
- 東京駅での問答
  - 「会員証で引換えできないですねー。おかしいな。クレジットカードでやってみましょう。駄目ですねー。」「古いクレジットカードでは？」「あ、できましたね。はい、切符です。」

# 用語

## 問題領域の用語

- JR東海、JR東
- モバイルSuica
- エクスプレス予約
- エクスプレス予約会員証、EX-ICカード、エクスプレスカード
- クレジットカード
  - ロンロンViewカード、アトレクラブViewカード
- e特急券

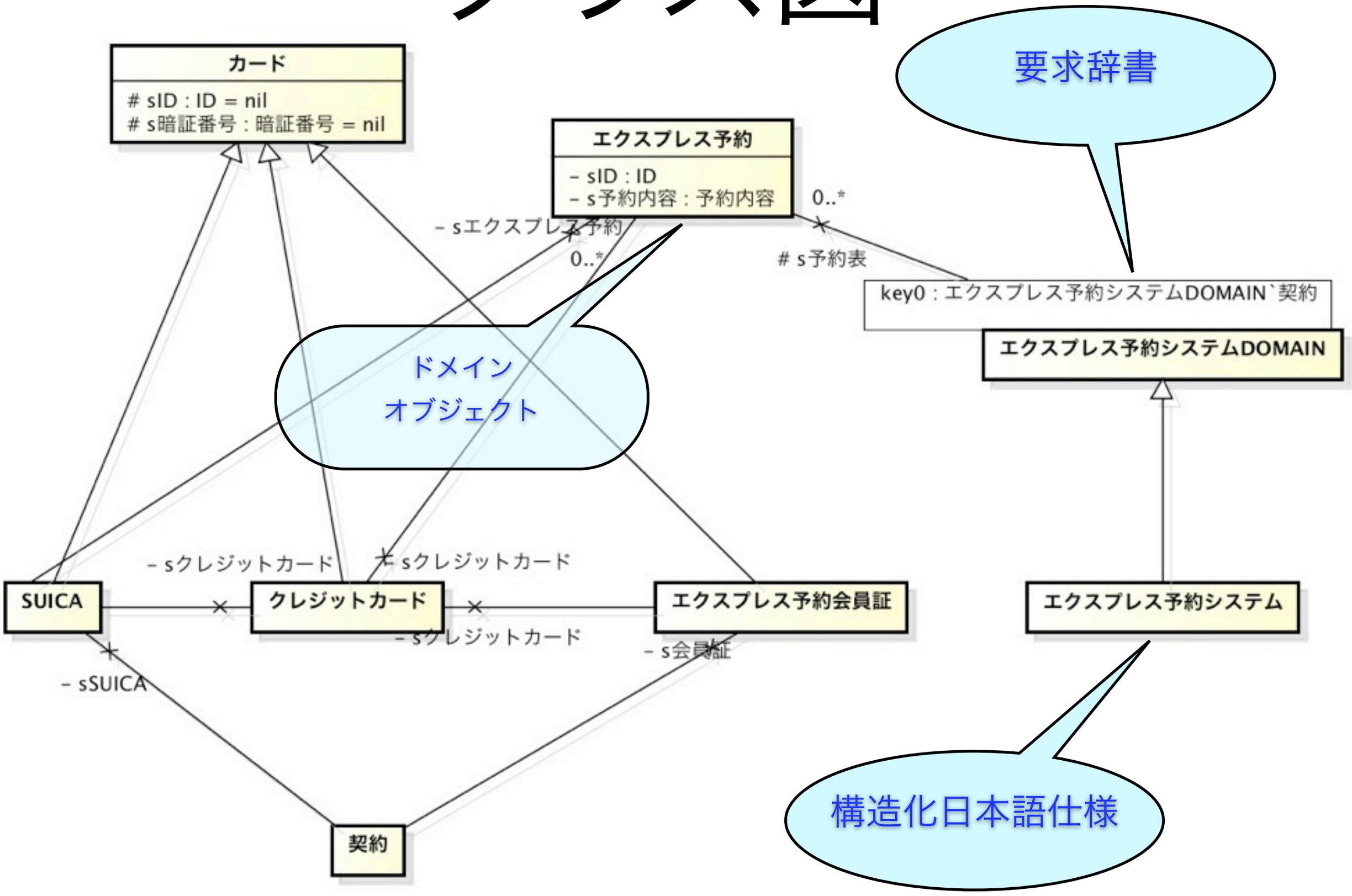
## モデル化対象用語

- モバイルSuica
- エクスプレス予約
- エクスプレス予約会員証
- クレジットカード
- e特急券

# モデル化の範囲

- 予約する
- e特急券を得る
- クレジットカードを切り替える

# クラス図


カード
# sID : ID = nil
# s暗証番号 : 暗証番号 = nil
エクスプレス予約
- sID : ID
- s予約内容 : 予約内容
0..*
- sエクスプレス予約
0..*
# s予約表
要求辞書
key0 : エクスプレス予約システムDOMAIN`契約
エクスプレス予約システムDOMAIN
ドメイン
オブジェクト
- sクレジットカード
- sクレジットカード
SUICA
クレジットカード
エクスプレス予約会員証
エクスプレス予約システム
- sクレジットカード
- s会員証
- sSUICA
契約
構造化日本語仕様

# エクスプレス予約システム

```
class エクスプレス予約システム is subclass of エクスプレス予約システムDOMAIN

...
public 予約する : 契約 * ID * クレジットカード * エクスプレス予約`予約内容 ==> ()
予約する(a契約, anID, aクレジットカード, a予約内容) == (
        def wエクスプレス予約 = new エクスプレス予約(anID, aクレジットカード, a予約内容) in
        if 予約がある契約である(a契約, s予約表) then
                予約表を更新する(a契約, wエクスプレス予約)
        else
                予約表に追加する(a契約, wエクスプレス予約));

public e特急券を得る : ID * クレジットカード ==> エクスプレス予約`予約内容
e特急券を得る(anID, aクレジットカード) == (
        def w予約 = 予約を得る(anID, aクレジットカード) in return w予約.予約内容を得る())
pre
        let w予約 = 予約を得る(anID, aクレジットカード) in
        aクレジットカード = w予約.クレジットカードを得る();

public e特急券を得る : ID * エクスプレス予約会員証 ==> エクスプレス予約`予約内容
e特急券を得る(anID, a会員証) == (
        def w予約 = 予約を得る(anID, a会員証) in return w予約.予約内容を得る())
pre
        let w予約 = 予約を得る(anID, a会員証) in
        a会員証.クレジットカードを得る() = w予約.クレジットカードを得る();
```

# エクスプレス予約システムDOMAIN

```
class エクスプレス予約システムDOMAIN is subclass of 共通定義

types
public 予約表 = map 契約 to set of エクスプレス予約;

instance variables
protected s予約表 : 予約表 := {|->};

operations
public 予約を得る : ID * エクスプレス予約会員証 ==> エクスプレス予約
予約を得る(anID, aエクスプレス予約会員証) == (
        let     w予約 in set dunion rng s予約表 be st
                w予約.クレジットカードを得る() = aエクスプレス予約会員証.クレジットカードを得る()  and  w予約.IDを得る() = anID in
        return w予約)
pre   exists w予約 in set dunion rng s予約表 &
                w予約.クレジットカードを得る() = aエクスプレス予約会員証.クレジットカードを得る()  and w予約.IDを得る() = anID;

...
public 予約表を更新する : 契約 * エクスプレス予約 ==> ()
予約表を更新する(a契約, aエクスプレス予約) ==
        s予約表 := s予約表 ++ {a契約 |-> 予約集合に追加する(s予約表(a契約), {aエクスプレス予約})}
post
        s予約表 = s予約表~ ++ {a契約 |-> 予約集合に追加する(s予約表~(a契約), {aエクスプレス予約})};

public 予約表に追加する : 契約 * エクスプレス予約 ==> ()
予約表に追加する(a契約, aエクスプレス予約) ==
        s予約表 := s予約表 munion {a契約 |-> {aエクスプレス予約}}
pre
        not 予約がある契約である(a契約, s予約表)
post
        s予約表 = s予約表~ munion {a契約 |-> {aエクスプレス予約}};
```

# 回帰テストケース

意図的に事前条件エラーを発生させ <RuntimeError>を起こさせた例

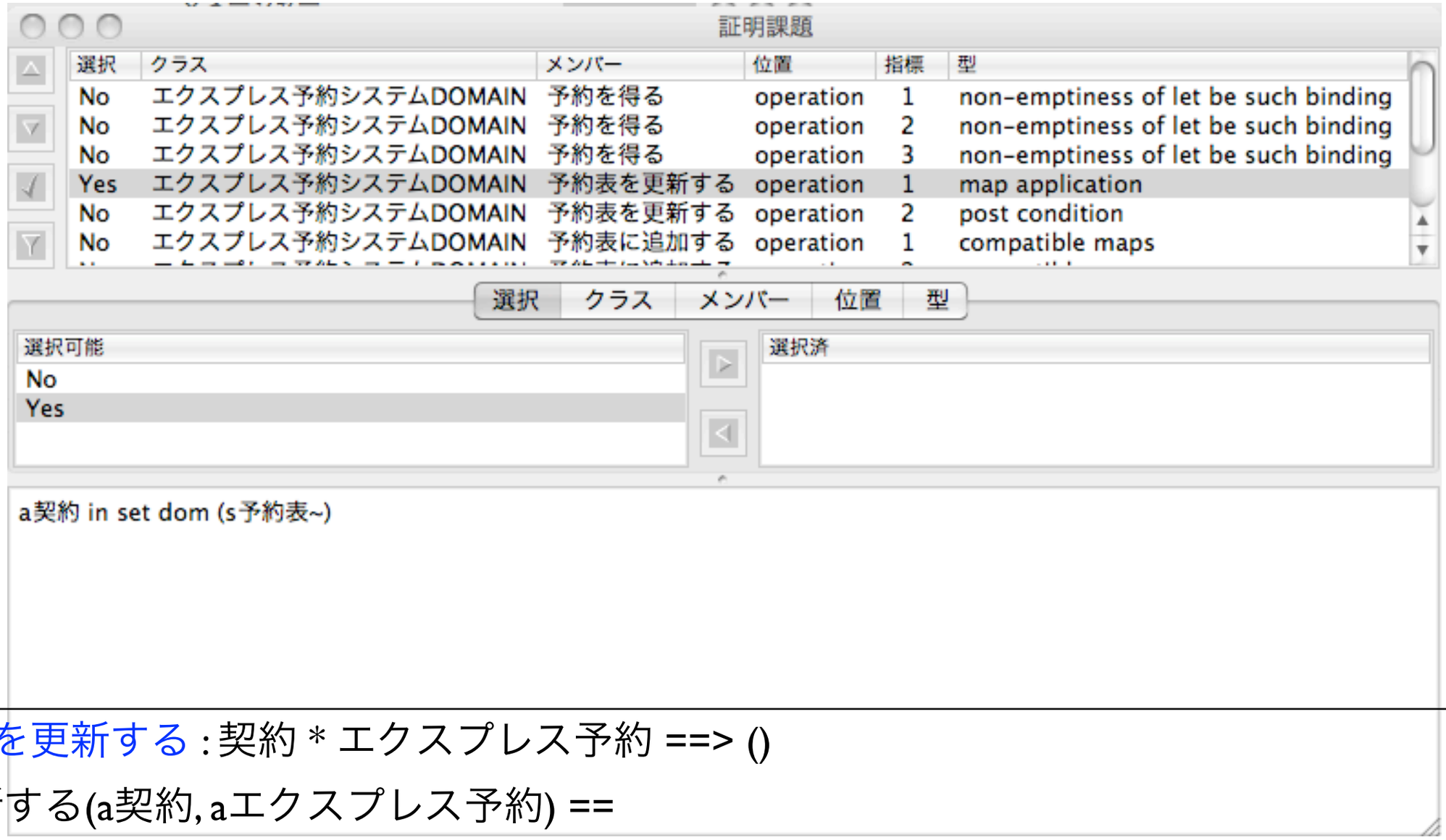
```
def
    wクレジットカード =  new クレジットカード(mk_token(<クレジットID01>), mk_token(<クレジットPW01>));
    wSUICA = new SUICA(mk_token(<SUICAID01>), wクレジットカード);
    w会員証 = new エクスプレス予約会員証(mk_token(<会員証ID01>), mk_token(<会員証PW01>), wクレジットカード);
    w契約 = new 契約(wSUICA, w会員証);
    wシステム = new エクスプレス予約システム({|->})
    w変更後のクレジットカード  = ...     in

...
wシステム.予約する(w契約, mk_token(<予約ID01>), wクレジットカード,  mk_token(<最初の予約内容>));

...
trap <RuntimeError> with
        print("\ttest01 テストの意図通り、エクスプレス予約会員証でe特急券を得ることに失敗\n")
in (
        wシステム.クレジットカードを切り替える(w契約, w変更後のクレジットカード);
        def w3予約内容 = wシステム.e特急券を得る(mk_token(<予約ID01>), w会員証)
        in
                assertTrue("\ttest01 テスト意図に反し、エクスプレス予約会員証でe特急券を得ることに失敗\n",
                w3予約内容 = mk_token(<最初の予約内容>)
        )
);
```

日本語記述の間違い例

# 証明課題生成で見つかった間違い




```
public 予約表を更新する : 契約 * エクスプレス予約 ==> ()
予約表を更新する(a契約, aエクスプレス予約) ==
    s予約表 := s予約表 ++ {a契約 |-> 予約集合に追加する(s予約表(a契約), {aエクスプレス予約})}
pre
    a契約 in set dom s予約表
post
    s予約表 = s予約表~ ++ {a契約 |-> 予約集合に追加する(s予約表~(a契約), {aエクスプレス予約})};
```

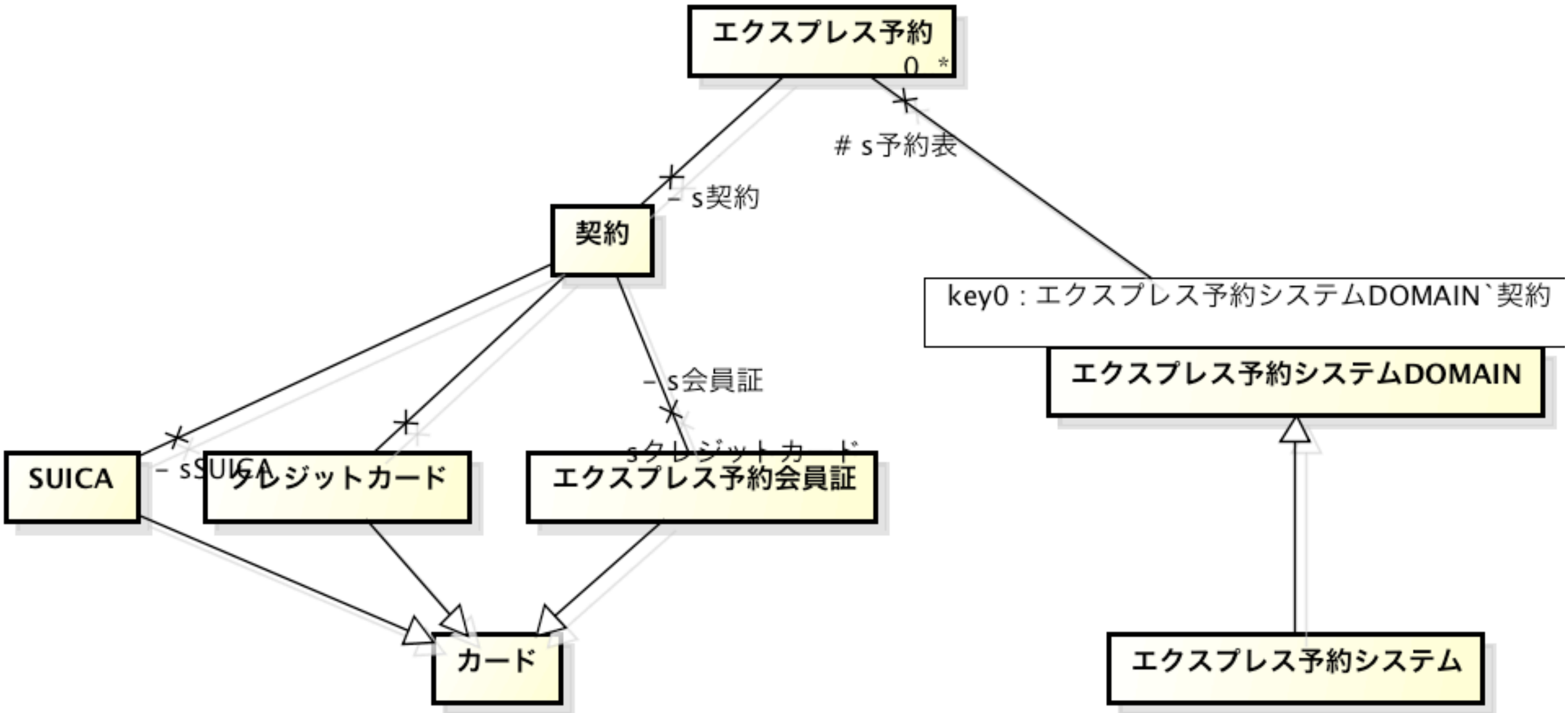
# 問題は何だったか？

- 要求仕様モデルの考慮不足
  - モバイルSUICA のクレジットカードを変更すると、エクスプレス予約を新規に契約せねばならず、エクスプレス予約会員証も新規に作成
    - 変更という概念が無い
  - 個々の予約を表すエクスプレス予約からクレジットカードにリンク
    - クレジットカードの変更に弱い
  - エクスプレス予約会員証からクレジットカードにリンク
    - クレジットカードの変更に弱い

# 本来どうすべきだったか？

- 契約が、クレジットカードやエクスプレス予約会員証とリンクすべき
- 個々のエクスプレス予約は、契約を介してクレジットカード情報に到達すべき

# 修正後のクラス図


エクスプレス予約
0 *
# s予約表
- s契約
契約
key0 : エクスプレス予約システムDOMAIN`契約
エクスプレス予約システムDOMAIN
- s会員証
sクレジットカード
SUICA
- sSUICA
クレジットカード
エクスプレス予約会員証
カード
エクスプレス予約システム

# 統計情報

| | 量 | 備考 |
|---|---|---|
| VDM++行数 | 501行 | 注釈抜き |
| 工数 | 3日 | モデル作成工数1日<br>整形・清書工数2日 |
| 修正工数 | 1日 | モデル修正工数1日 |

筆者が実際に問題を解決した工数より少ない

# 結論

- VDMは構造化日本語仕様として使える
  - 構文・型チェック、証明課題レビューで静的に検証できる
  - 回帰テストにより、動的に妥当性検査ができる
- VDMソース自体が、要求辞書ともなる
- 日本語仕様より、記述と検証の工数が少ない
  - 特に、仕様修正に強い
- 擬似コード形式の日本語仕様に近い形で、かなりの部分を記述できる