2012年度ソフトウェア工学分野の先導的研究支援事業

実用性が高い形式工学手法と支援ツールの研究開発

法政大学・情報科学研究科

劉少英

Email: sliu@hosei.ac.jp

HP: http://cis.k.hosei.ac.jp/~sliu/

目次

1. 研究目的と研究内容

1.1 研究目的

次のソフトウェア開発の問題点を解決し、より実用性が高い形式工学手法を開発します。

既存のソフトウェア開発への形式手法の適用性が低い

従来のソフトウェア開発手法は信頼性を保証することができない、コストも高い

1.2　研究内容

次の三つの技術を提案することによって、実用性が高い形式工学手法と支援ツールを開発しました。

SOFL三段階漸進的な形式仕様記述技術

仕様アニメーション技術

仕様パターンに基づく形式仕様の作成支援技術

2. 研究成果

(1) SOFL三段階漸進的な形式仕様記述技術を提案、支援ツールのプロトタイプを開発しました。

(2) 非形式仕様アニメーションの目標と方法を確立、支援ツールのプロトタイプを開発しました。

(3) 半形式仕様アニメーションの目標と方法を確立、支援ツールのプロトタイプを開発しました。

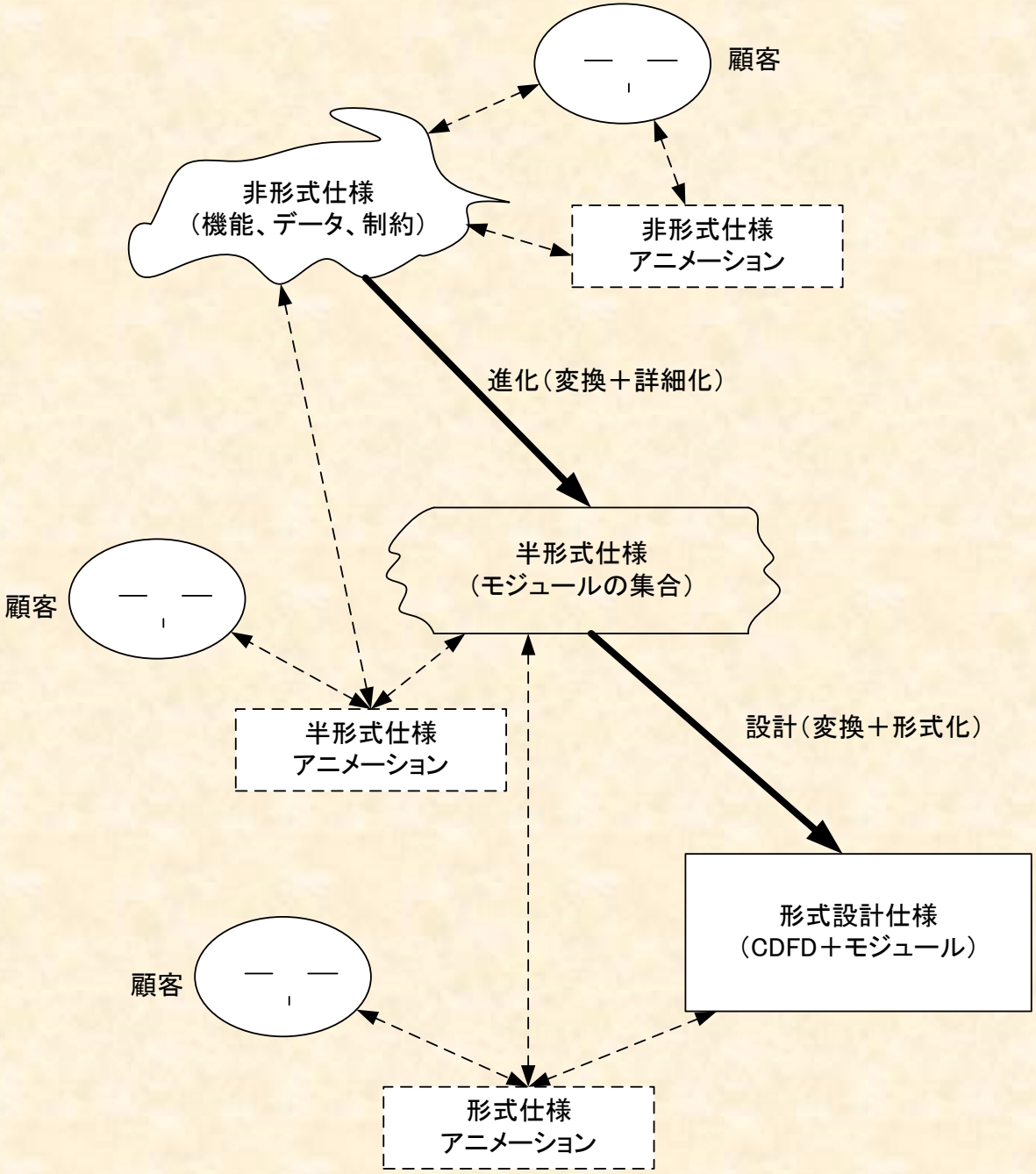
(4) 形式仕様アニメーションの目標と方法を確立、支援ツールのプロトタイプを開発しました。

(5) 形式仕様パターンの定義および仕様パターンシステムを提案しました。

(6) 仕様パターン知識の表現方法、検索と適用アルゴリズム、および支援ツールを開発しました。

2.1 SOFL三段階漸進的な形式仕様記述技の提案および支援ツールのプロトタイプの開発

(1) SOFL三段階漸進的な形式仕様記述技術の提案

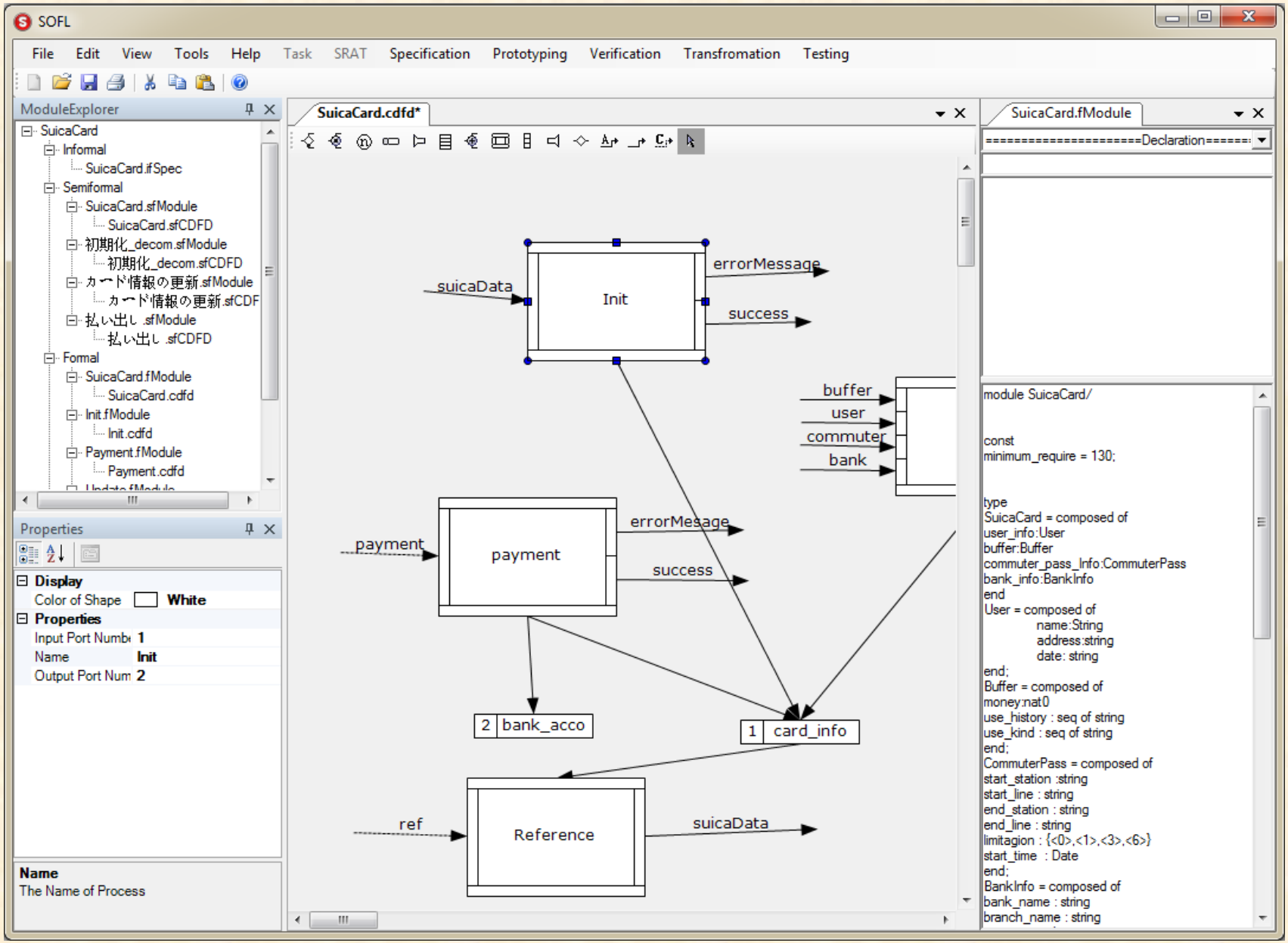

顧客
非形式仕様
（機能、データ、制約）
非形式仕様
アニメーション
進化（変換＋詳細化）
半形式仕様
（モジュールの集合）
顧客
半形式仕様
アニメーション
設計（変換＋形式化）
形式設計仕様
（CDFD＋モジュール）
顧客
形式仕様
アニメーション

(2) 支援ツールのプロトタイプの開発

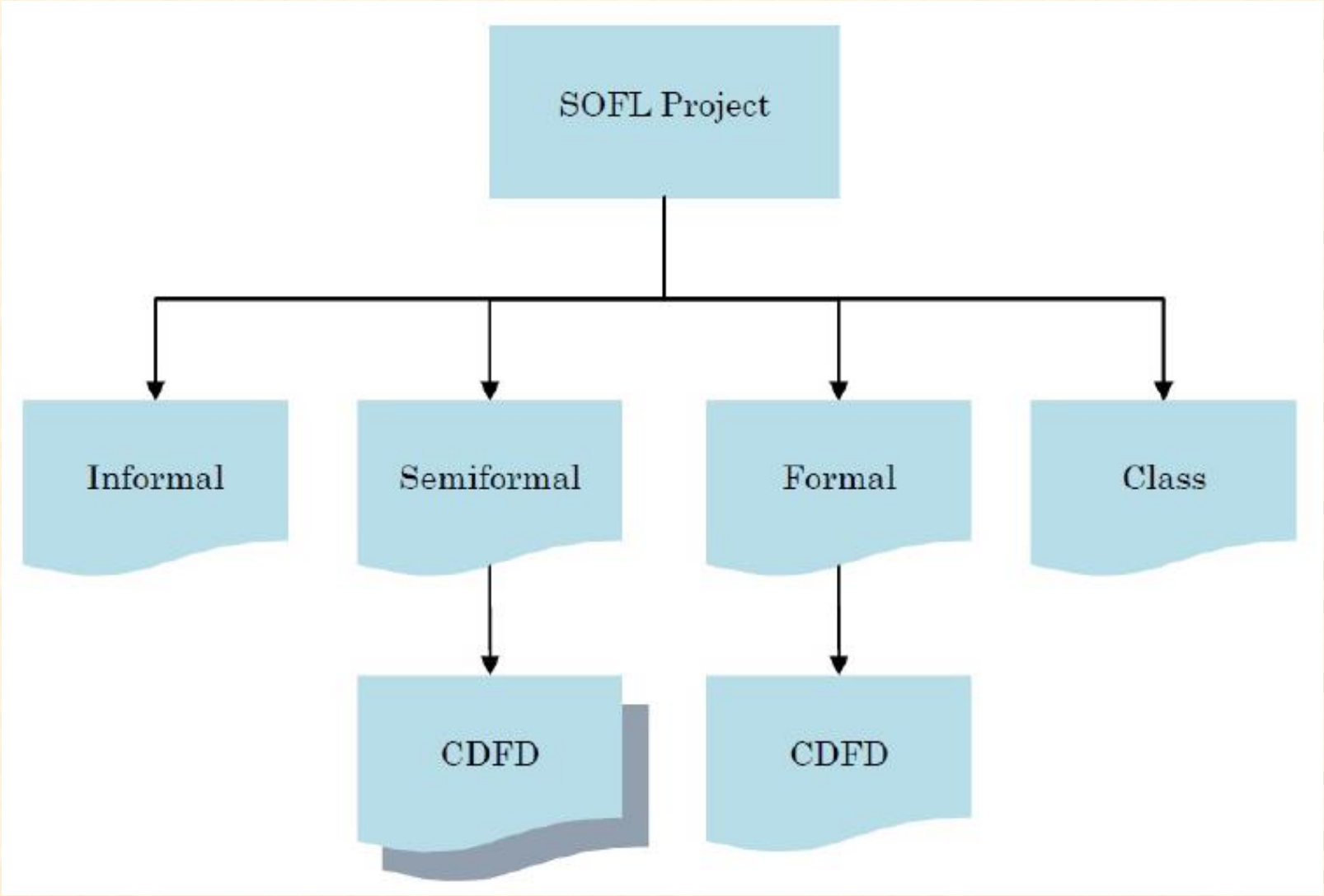
本支援ツールは、次の主な機能を提供しています。

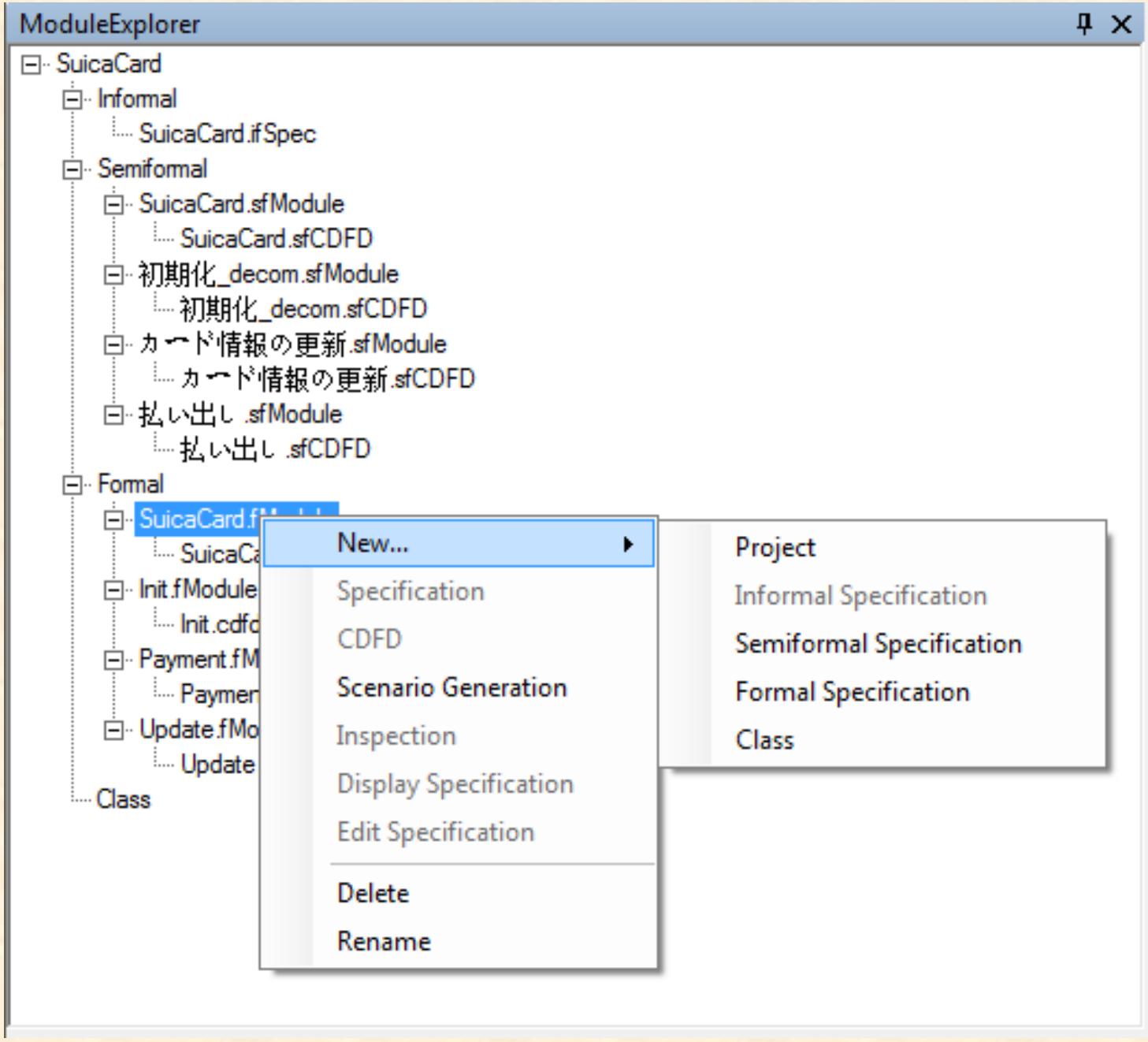
① 仕様の作成と編集のためのGUI構造
② プロジェクト階層
③ 非形式仕様エディタ
④ 半形式仕様エディタ
⑤ 形式仕様エディタ
⑥ CDFDを描くボード
⑦ プロセスの分解
⑧ 仕様のエクスポート

① 仕様の作成と編集のためのGUI構造

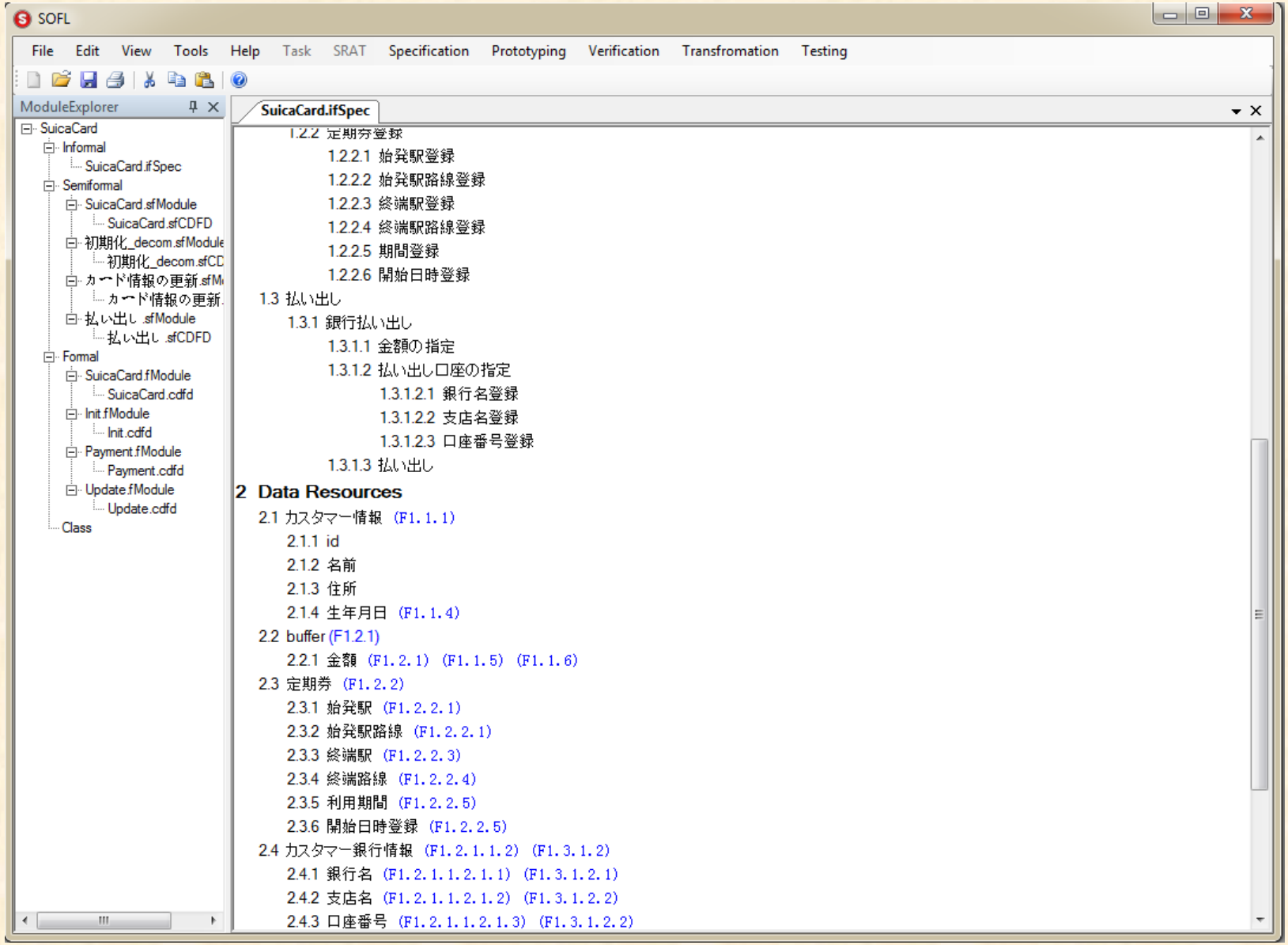


② プロジェクト階層

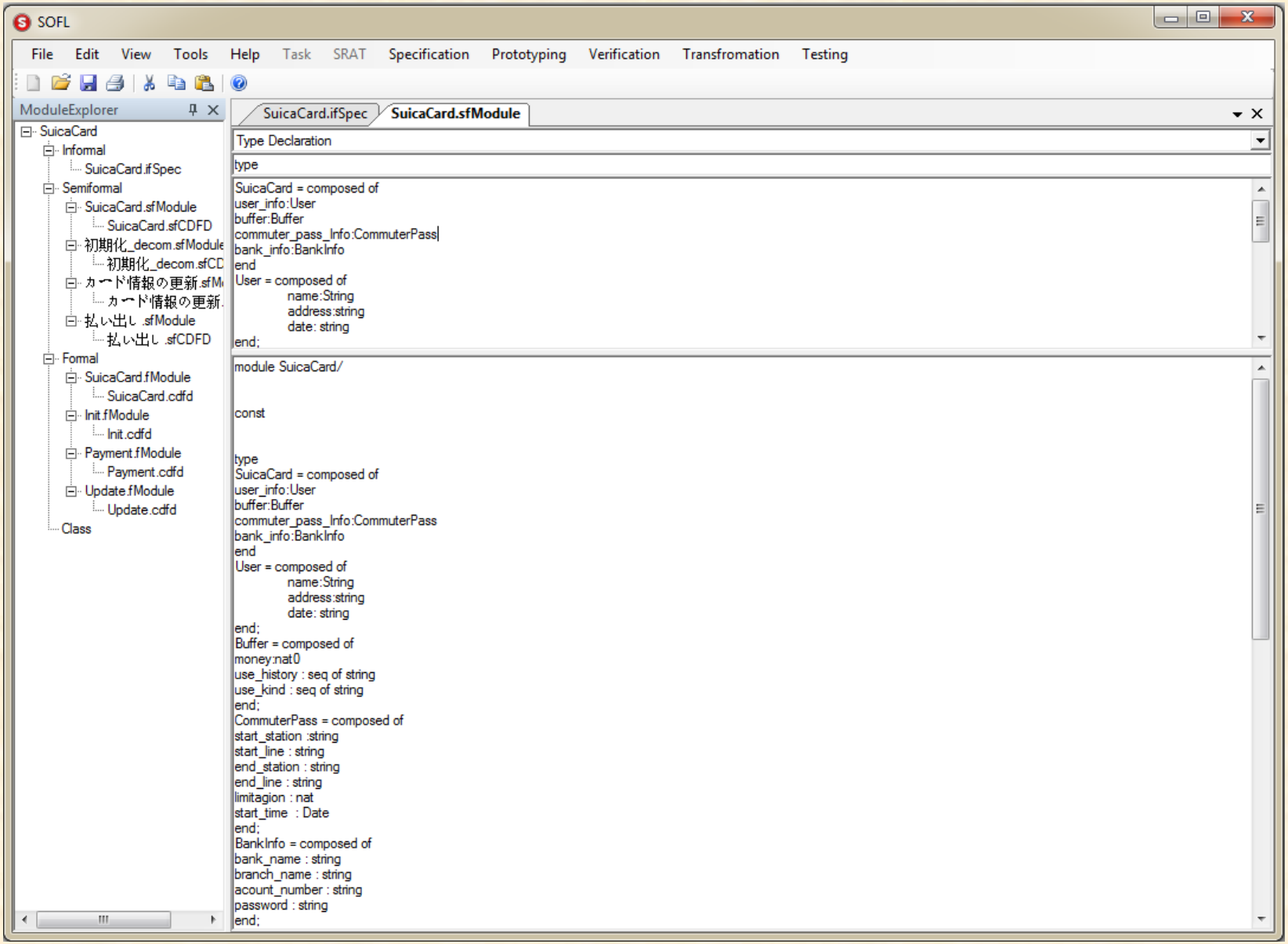



ModuleExplorer
SuicaCard
Informal
SuicaCard.ifSpec
Semiformal
SuicaCard.sfModule
SuicaCard.sfCDFD
初期化_decom.sfModule
初期化_decom.sfCDFD
カード情報の更新.sfModule
カード情報の更新.sfCDFD
払い出し .sfModule
払い出し .sfCDFD
Formal
Class
New...
Specification
CDFD
Scenario Generation
Inspection
Display Specification
Edit Specification
Delete
Rename
Project
Informal Specification
Semiformal Specification
Formal Specification
Class

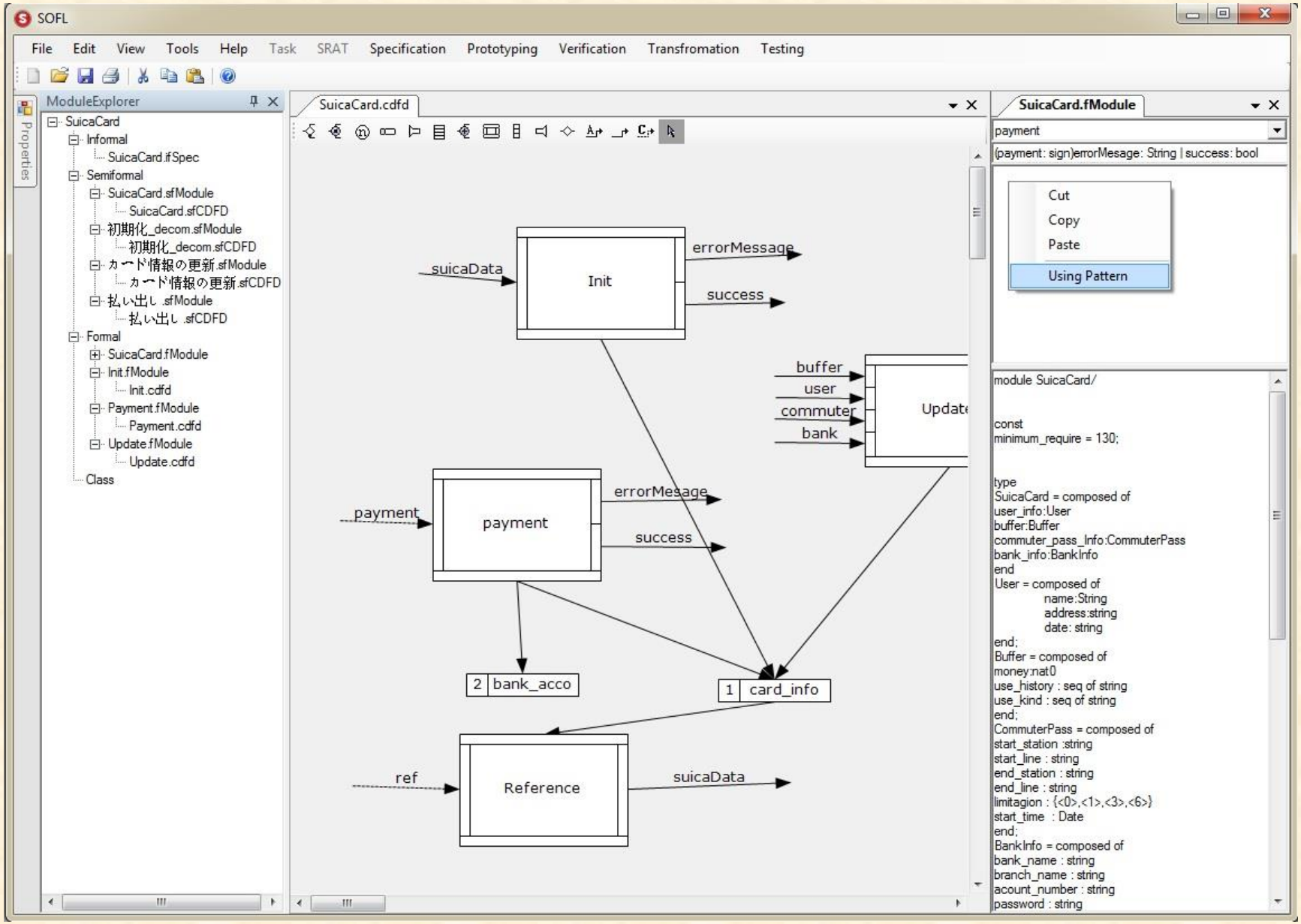
③ 非形式仕様エディタ


SOFL
File Edit View Tools Help Task SRAT Specification Prototyping Verification Transfromation Testing
ModuleExplorer
SuicaCard
Informal
SuicaCard.ifSpec
Semiformal
SuicaCard.sfModule
SuicaCard.sfCDFD
初期化_decom.sfModule
初期化_decom.sfCD
カード情報の更新.sfM
カード情報の更新
払い出し .sfModule
払い出し .sfCDFD
Formal
SuicaCard.fModule
SuicaCard.cdfd
Init.fModule
Init.cdfd
Payment.fModule
Payment.cdfd
Update.fModule
Update.cdfd
Class
SuicaCard.ifSpec
1.2.2 定期券登録
1.2.2.1 始発駅登録
1.2.2.2 始発駅路線登録
1.2.2.3 終端駅登録
1.2.2.4 終端駅路線登録
1.2.2.5 期間登録
1.2.2.6 開始日時登録
1.3 払い出し
1.3.1 銀行払い出し
1.3.1.1 金額の指定
1.3.1.2 払い出し口座の指定
1.3.1.2.1 銀行名登録
1.3.1.2.2 支店名登録
1.3.1.2.3 口座番号登録
1.3.1.3 払い出し
2 Data Resources
2.1 カスタマー情報 (F1.1.1)
2.1.1 id
2.1.2 名前
2.1.3 住所
2.1.4 生年月日 (F1.1.4)
2.2 buffer (F1.2.1)
2.2.1 金額 (F1.2.1) (F1.1.5) (F1.1.6)
2.3 定期券 (F1.2.2)
2.3.1 始発駅 (F1.2.2.1)
2.3.2 始発駅路線 (F1.2.2.1)
2.3.3 終端駅 (F1.2.2.3)
2.3.4 終端路線 (F1.2.2.4)
2.3.5 利用期間 (F1.2.2.5)
2.3.6 開始日時登録 (F1.2.2.5)
2.4 カスタマー銀行情報 (F1.2.1.1.2) (F1.3.1.2)
2.4.1 銀行名 (F1.2.1.1.2.1.1) (F1.3.1.2.1)
2.4.2 支店名 (F1.2.1.1.2.1.2) (F1.3.1.2.2)
2.4.3 口座番号 (F1.2.1.1.2.1.3) (F1.3.1.2.2)

④ 半形式仕様エディタ

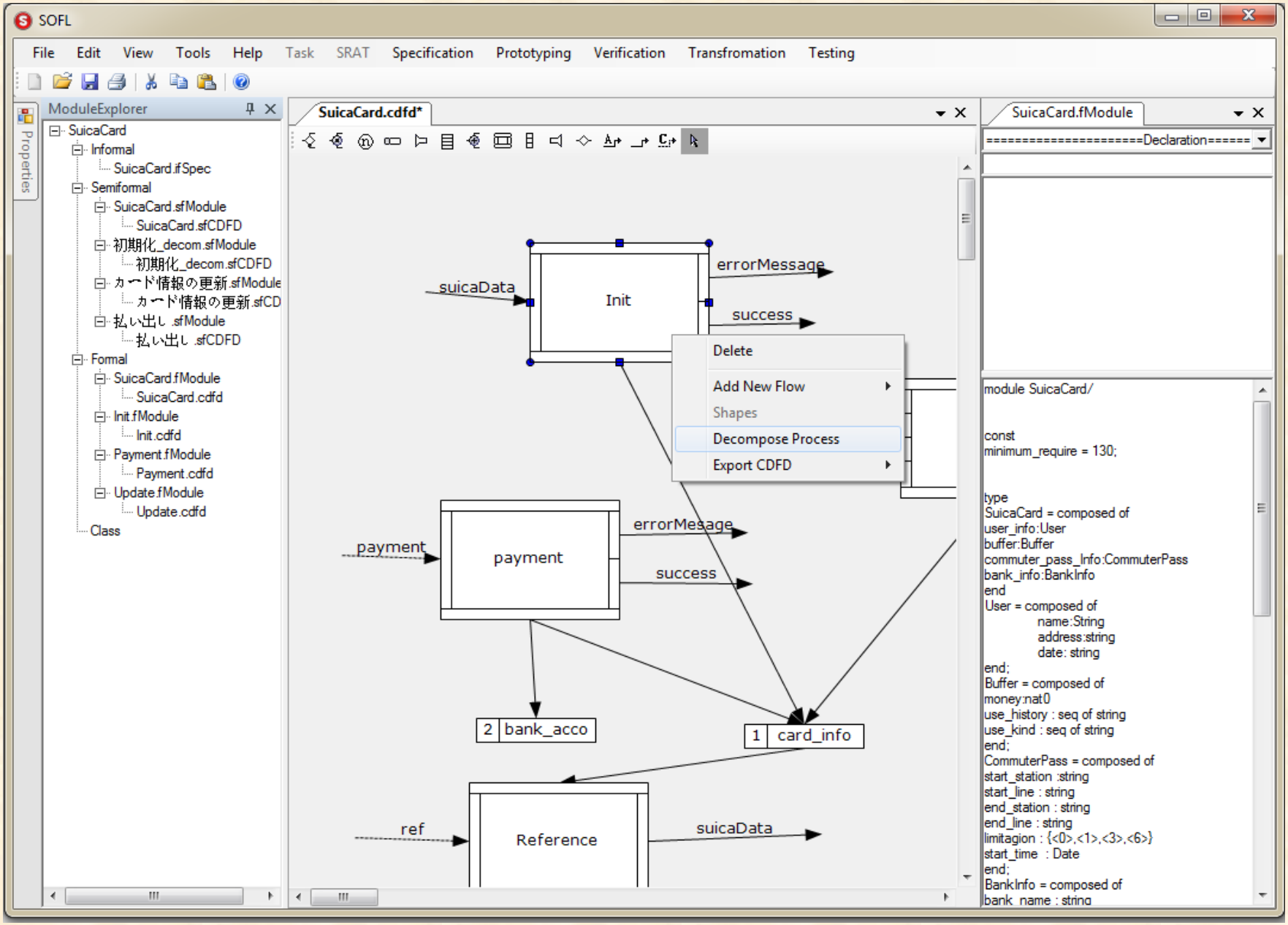

SOFL
File Edit View Tools Help Task SRAT Specification Prototyping Verification Transfromation Testing
ModuleExplorer
SuicaCard
Informal
SuicaCard.ifSpec
Semiformal
SuicaCard.sfModule
SuicaCard.sfCDFD
初期化_decom.sfModule
初期化_decom.sfCD
カード情報の更新.sfM
カード情報の更新
払い出し .sfModule
払い出し .sfCDFD
Formal
SuicaCard.fModule
SuicaCard.cdfd
Init.fModule
Init.cdfd
Payment.fModule
Payment.cdfd
Update.fModule
Update.cdfd
Class
SuicaCard.ifSpec
SuicaCard.sfModule
Type Declaration
type
SuicaCard = composed of
user_info:User
buffer:Buffer
commuter_pass_Info:CommuterPass
bank_info:BankInfo
end
User = composed of
name:String
address:string
date: string
end;
module SuicaCard/
const
type
SuicaCard = composed of
user_info:User
buffer:Buffer
commuter_pass_Info:CommuterPass
bank_info:BankInfo
end
User = composed of
name:String
address:string
date: string
end;
Buffer = composed of
money:nat0
use_history : seq of string
use_kind : seq of string
end;
CommuterPass = composed of
start_station :string
start_line : string
end_station : string
end_line : string
limitagion : nat
start_time : Date
end;
BankInfo = composed of
bank_name : string
branch_name : string
acount_number : string
password : string
end;

⑤ 形式仕様エディタとCDFDを描くボード



2012年度ソフトウェア工学分野の先導的研究支援事業

⑥ プロセスの分解


SOFL
File
Edit
View
Tools
Help
Task
SRAT
Specification
Prototyping
Verification
Transfromation
Testing
ModuleExplorer
SuicaCard
Informal
SuicaCard.ifSpec
Semiformal
SuicaCard.sfModule
SuicaCard.sfCDFD
初期化_decom.sfModule
初期化_decom.sfCDFD
カード情報の更新.sfModule
カード情報の更新.sfCD
払い出し .sfModule
払い出し .sfCDFD
Formal
SuicaCard.fModule
SuicaCard.cdfd
Init.fModule
Init.cdfd
Payment.fModule
Payment.cdfd
Update.fModule
Update.cdfd
Class
SuicaCard.cdfd*
suicaData
Init
errorMessage
success
Delete
Add New Flow
Shapes
Decompose Process
Export CDFD
payment
payment
errorMesage
success
2 bank_acco
1 card_info
ref
Reference
suicaData
SuicaCard.fModule
=====Declaration======
module SuicaCard/
const
minimum_require = 130;
type
SuicaCard = composed of
user_info:User
buffer:Buffer
commuter_pass_Info:CommuterPass
bank_info:BankInfo
end
User = composed of
name:String
address:string
date: string
end;
Buffer = composed of
money:nat0
use_history : seq of string
use_kind : seq of string
end;
CommuterPass = composed of
start_station :string
start_line : string
end_station : string
end_line : string
limitagion : {<0>,<1>,<3>,<6>}
start_time : Date
end;
BankInfo = composed of
bank_name : string

⑦ 仕様のエクスポート

作成されたすべての仕様（非形式仕様、半形式仕様、および形式仕様）を、MS Word ファイル および RTF ファイルへエクスポートすることができる。

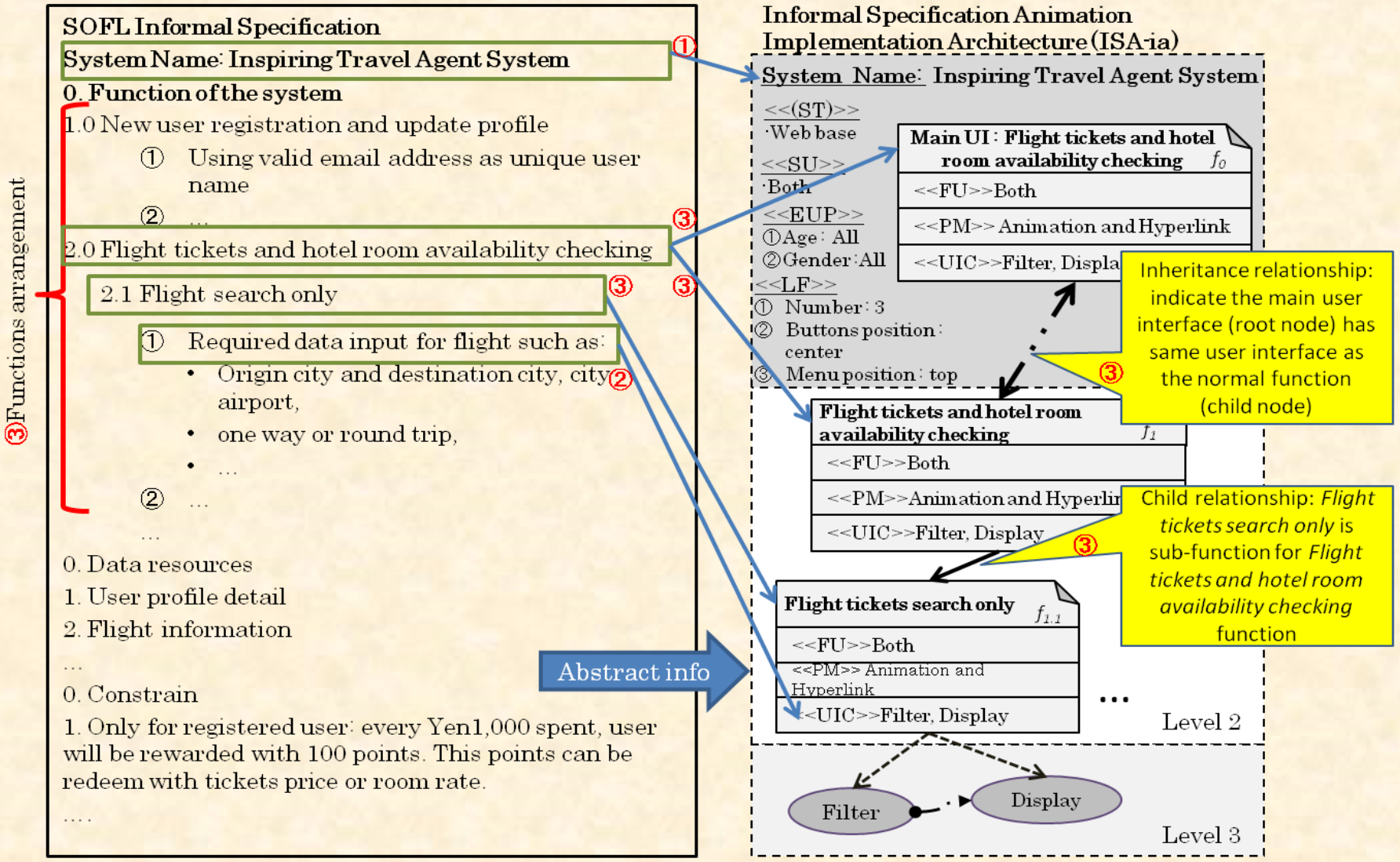
また、CDFDをJPEG、WMF、BMP、およびPNGフォーマットのファイルへエクスポートすることもできる。

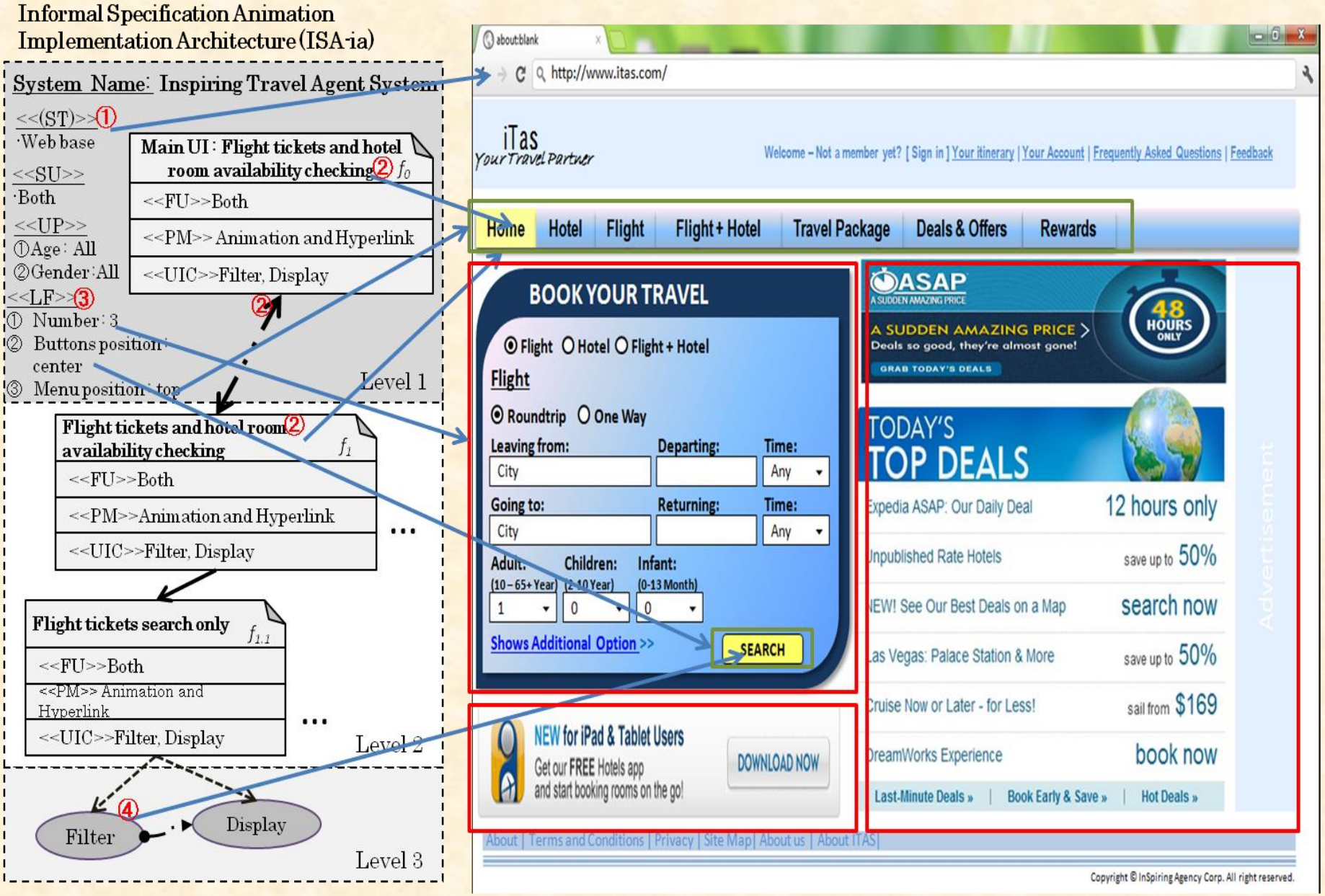
2.2 非形式仕様アニメーションの目標と方法、および支援ツールのプロトタイプ

(1) 非形式仕様アニメーションの目標と方法

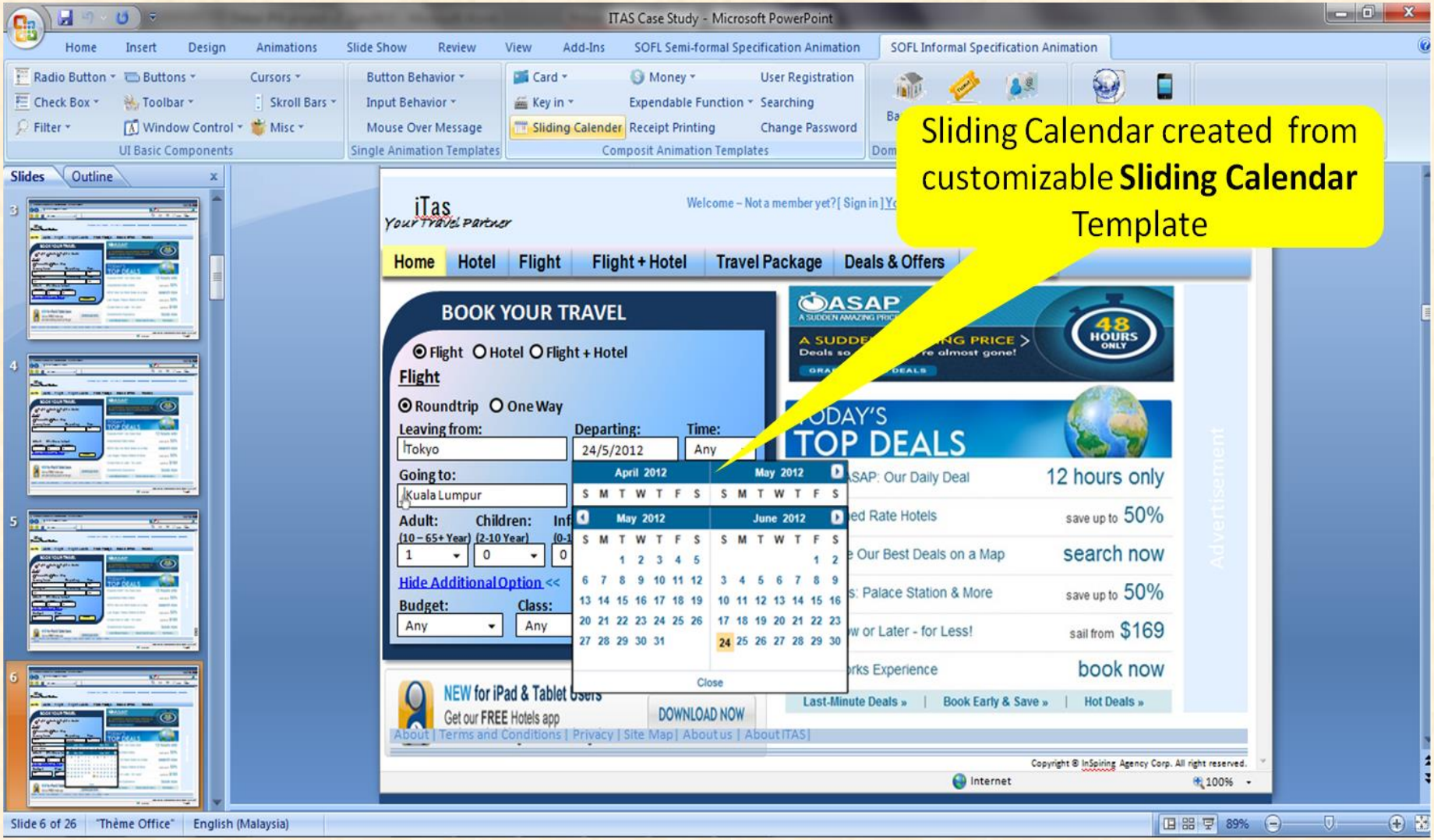
目標: 顧客が最も関心している機能のユーザインタフェースを動的に表現する。

非形式仕様のアニメーション方法：




Informal Specification Animation Implementation Architecture (ISA-ia)
System Name: Inspiring Travel Agent System
<<(ST)>>①
·Web base
<<SU>>
·Both
<<UP>>
①Age : All
②Gender :All
<<LF>>③
① Number : 3
② Buttons position : center
③ Menu position : top
Main UI : Flight tickets and hotel room availability checking ② f_0
<<FU>>Both
<<PM>> Animation and Hyperlink
<<UIC>>Filter, Display
Level 1
Flight tickets and hotel room availability checking ② f_1
<<FU>>Both
<<PM>>Animation and Hyperlink
<<UIC>>Filter, Display
Flight tickets search only $f_{1.1}$
<<FU>>Both
<<PM>> Animation and Hyperlink
<<UIC>>Filter, Display
Level 2
④
Filter
Display
Level 3
about:blank
http://www.itas.com/
iTas
Your Travel Partner
Welcome – Not a member yet? [Sign in] Your itinerary | Your Account | Frequently Asked Questions | Feedback
Home
Hotel
Flight
Flight + Hotel
Travel Package
Deals & Offers
Rewards
BOOK YOUR TRAVEL
Flight
Hotel
Flight + Hotel
Flight
Roundtrip
One Way
Leaving from:
Departing:
Time:
City
Any
Going to:
Returning:
Time:
City
Any
Adult: (10 – 65+ Year)
Children: (2-10 Year)
Infant: (0-13 Month)
1
0
0
Shows Additional Option >>
SEARCH
NEW for iPad & Tablet Users
Get our FREE Hotels app and start booking rooms on the go!
DOWNLOAD NOW
ASAP
A SUDDEN AMAZING PRICE
48 HOURS ONLY
A SUDDEN AMAZING PRICE >
Deals so good, they're almost gone!
GRAB TODAY'S DEALS
TODAY'S TOP DEALS
Expedia ASAP: Our Daily Deal
12 hours only
Unpublished Rate Hotels
save up to 50%
NEW! See Our Best Deals on a Map
search now
Las Vegas: Palace Station & More
save up to 50%
Cruise Now or Later - for Less!
sail from $169
DreamWorks Experience
book now
Last-Minute Deals »
Book Early & Save »
Hot Deals »
Advertisement
About | Terms and Conditions | Privacy | Site Map | About us | About ITAS
Copyright © InSpiring Agency Corp. All right reserved.

(2)非形式仕様アニメーションの支援ツールのプロトタイプ

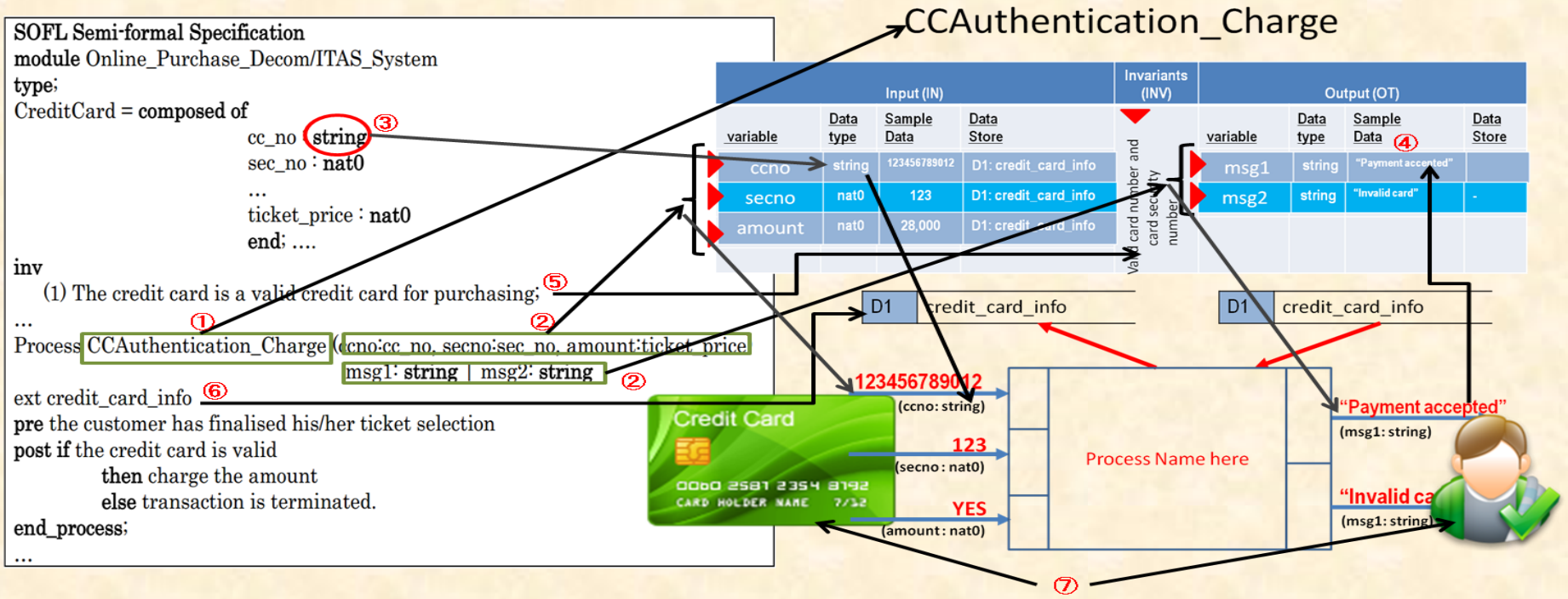


2.3 半形式仕様アニメーションの目標と方法、および支援ツールのプロトタイプ

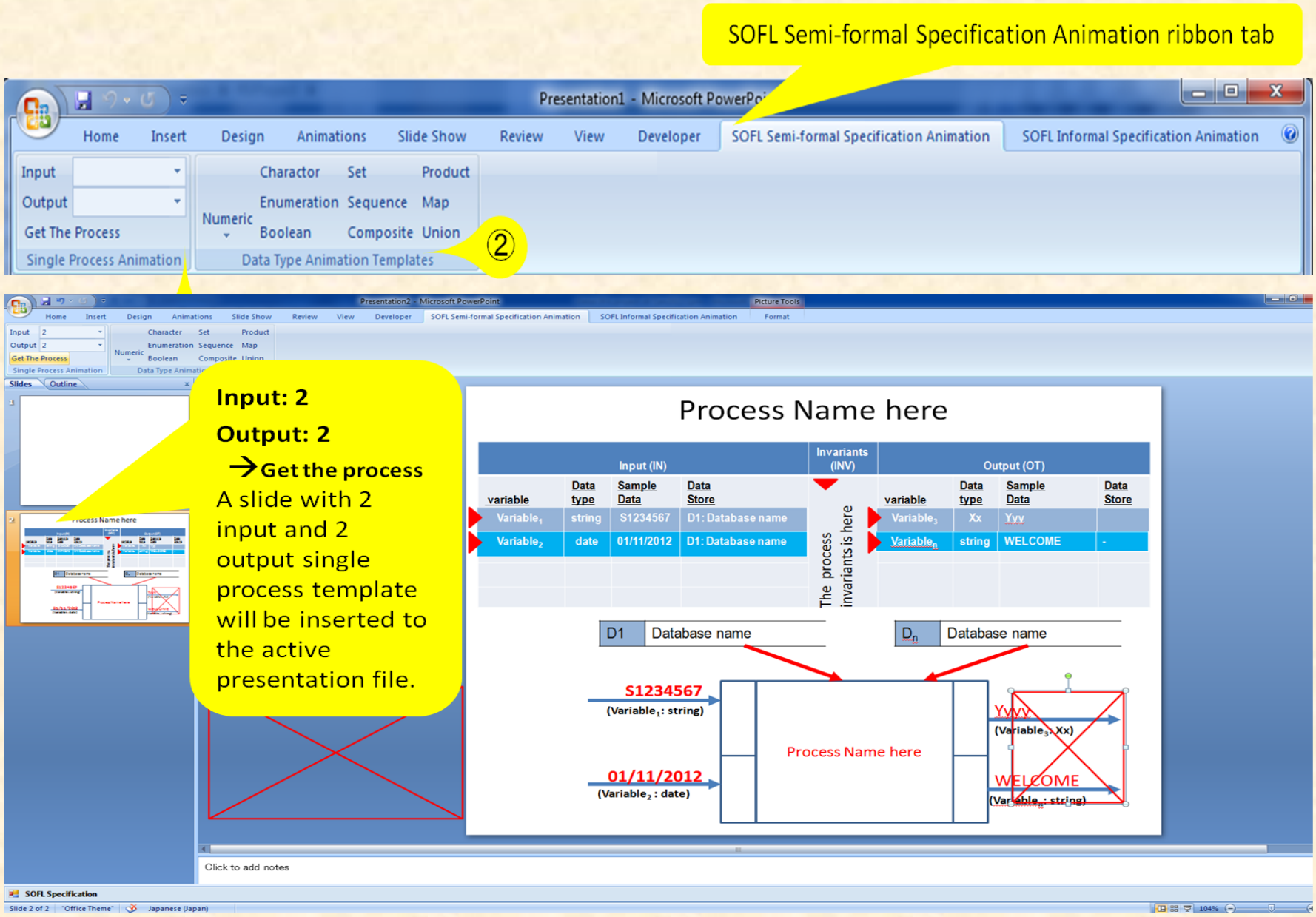
(1) 半形式仕様アニメーションの目標と方法

目標: プロセス(操作)の入力変数と型、出力変数と型、状態変数と型、事前条件と事後条件を確認する。

半形式仕様アニメーションの方法：



(2) 半形式仕様アニメーションの支援ツールのプロトタイプ



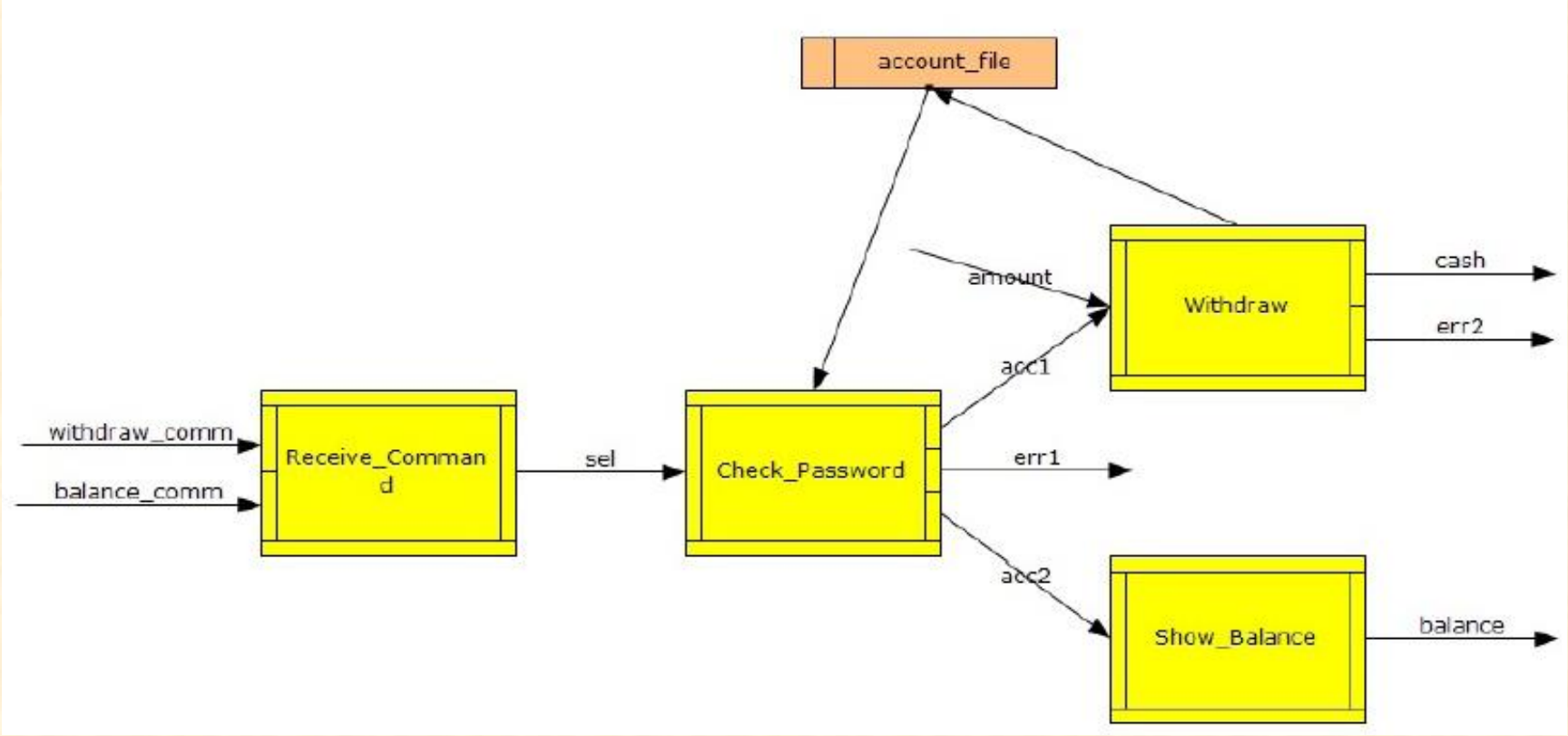
2.4 形式仕様アニメーションの目標と方法、および支援ツールのプロトタイプ

（1）形式仕様アニメーションの目標と方法

目標： システム設計仕様の整合性と正当性を検証する。

形式仕様アニメーションの方法：

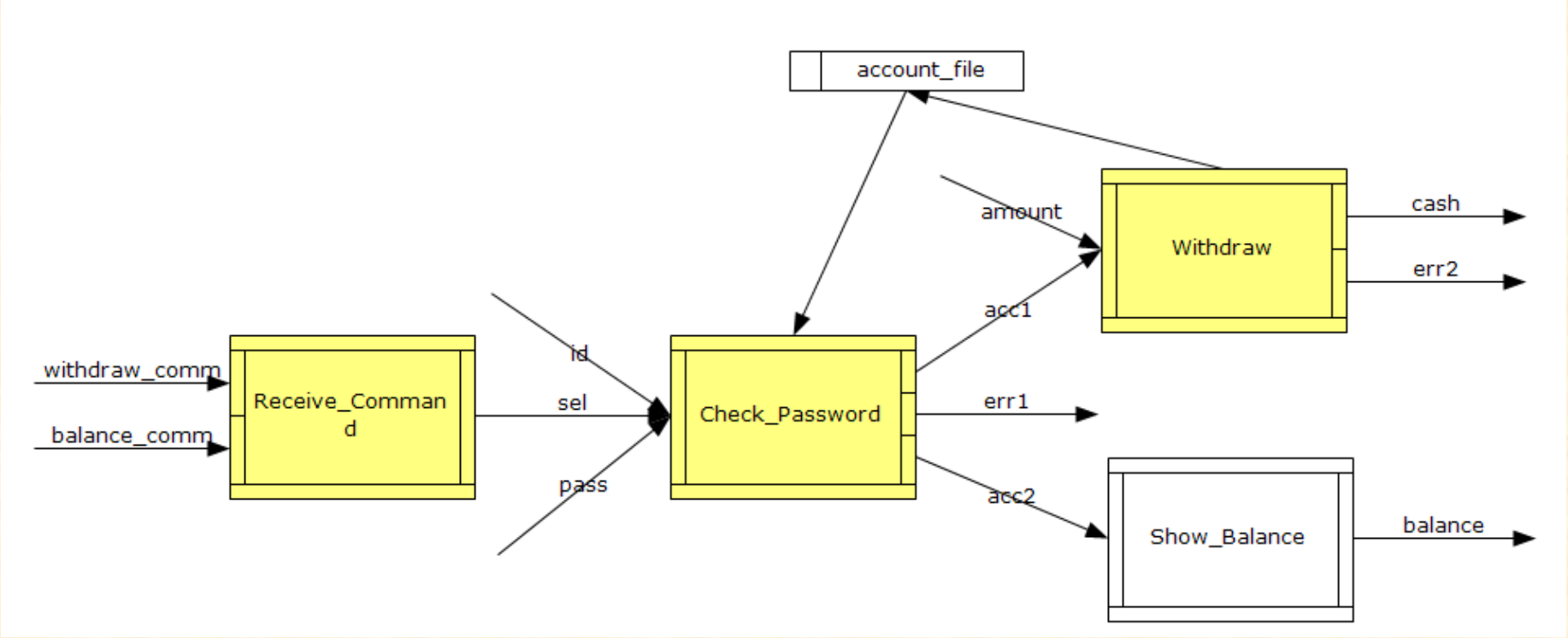
ステップ１：　システムのアーキテクチャーを定義するCDFDからシステム機能シナリオを抜き出す。

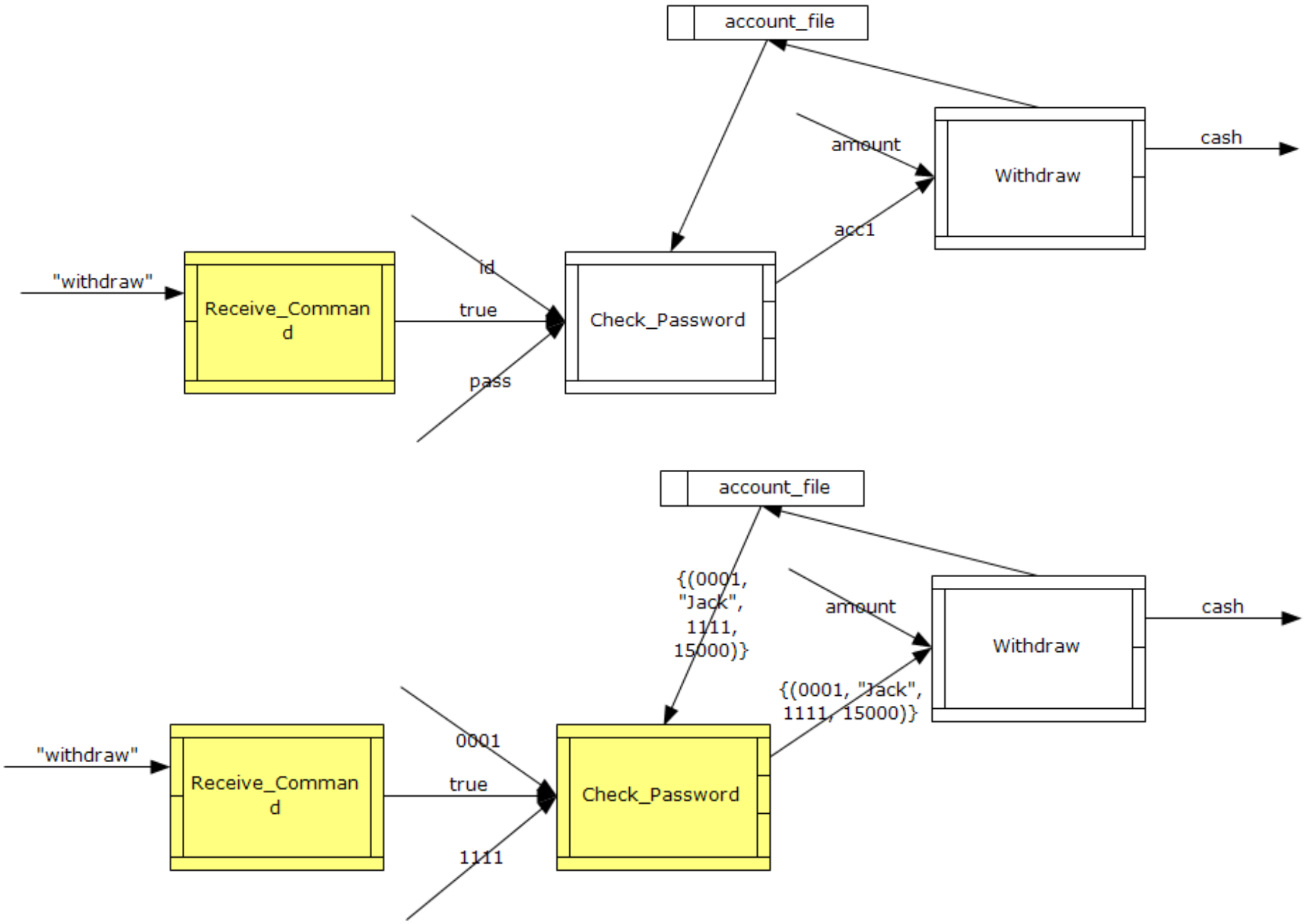


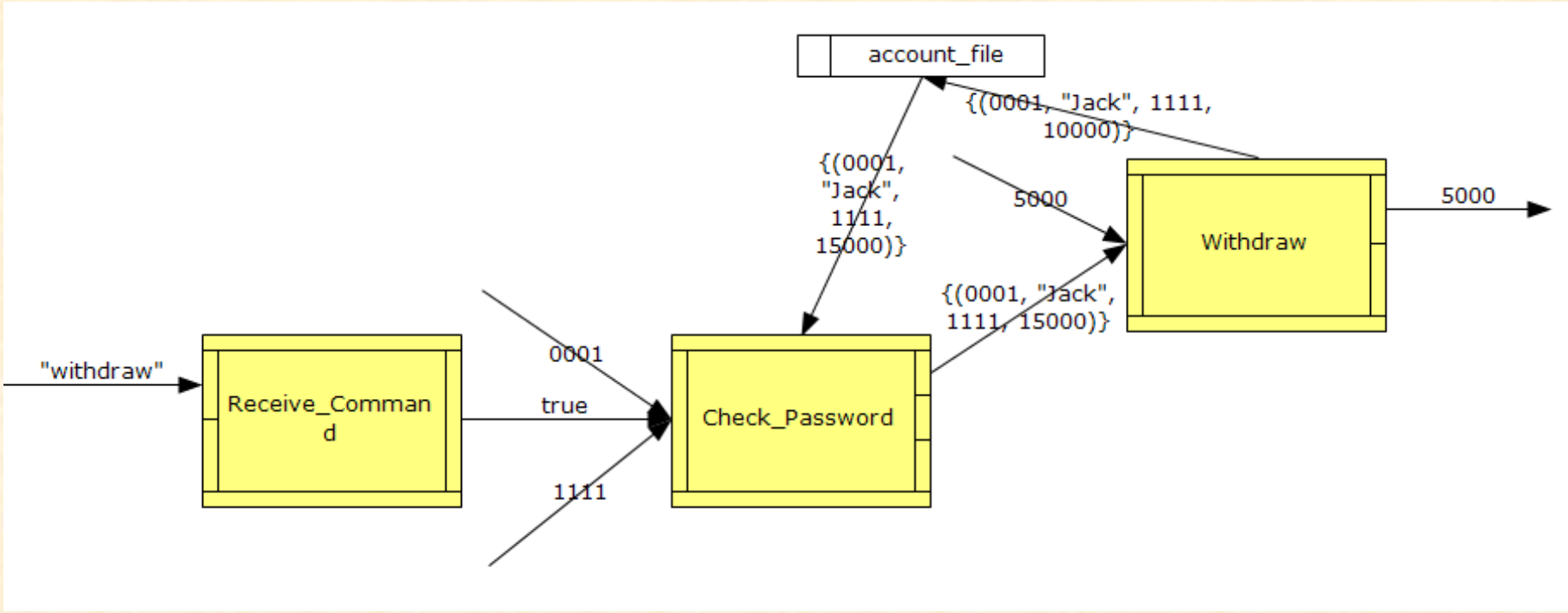
システム機能シナリオの事例:

1. {withdraw_comm}Receive_Command{sel}Check_Password{act1, amount}Withdraw{cash}
2. {withdraw_comm}Receive_Command{sel}Check_Password{act1, amount}Withdraw{err2}
3. {withdraw_comm}Receive_Command{sel}Check_Password{err1}
4. {balance_comm}Receive_Command{sel}Check_Password{act2} Show_Balance{balance}

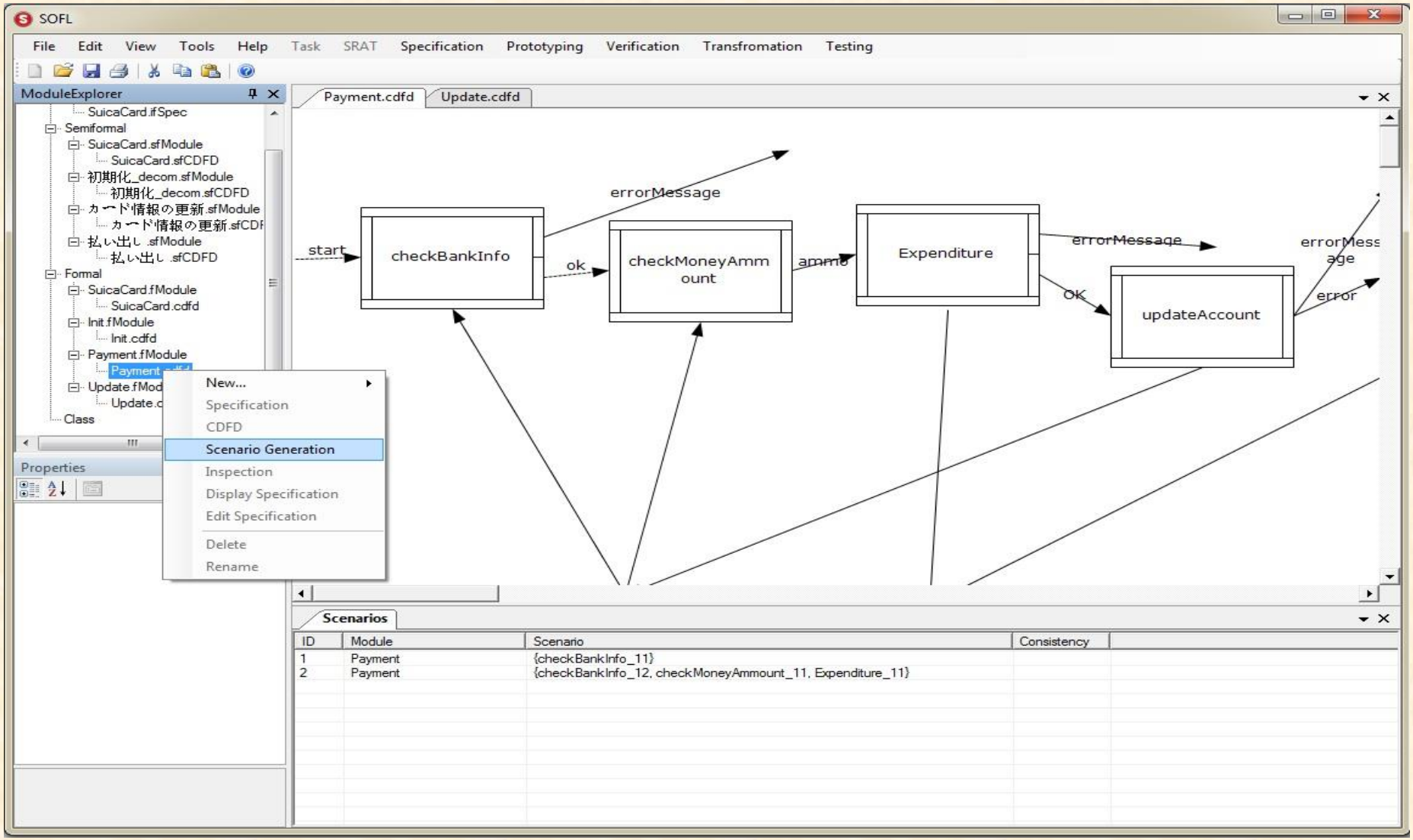
ステップ２：　抜き出されたシステム機能シナリオを動的に表現する。




account_file
amount
Withdraw
cash
"withdraw"
Receive_Comman
d
Id
true
pass
Check_Password
acc1
account_file
{(0001,
"Jack",
1111,
15000)}
amount
Withdraw
cash
{(0001, "Jack",
1111, 15000)}
"withdraw"
Receive_Comman
d
0001
true
1111
Check_Password


account_file
{(0001, "Jack", 1111, 10000)}
{(0001, "Jack", 1111, 15000)}
5000
Withdraw
5000
{(0001, "Jack", 1111, 15000)}
"withdraw"
Receive_Command
0001
true
1111
Check_Password

(2) 形式仕様アニメーションの支援ツールのプロトタイプ



2.5 形式仕様パターンの定義および仕様パターンシステム

(1)形式仕様パターンの定義:

定義: 仕様パターンは、次の項目を含む仕様のテンプレートです:

① パターン名
② パターンの説明
③ パターンの構成要素
④ パターン解答

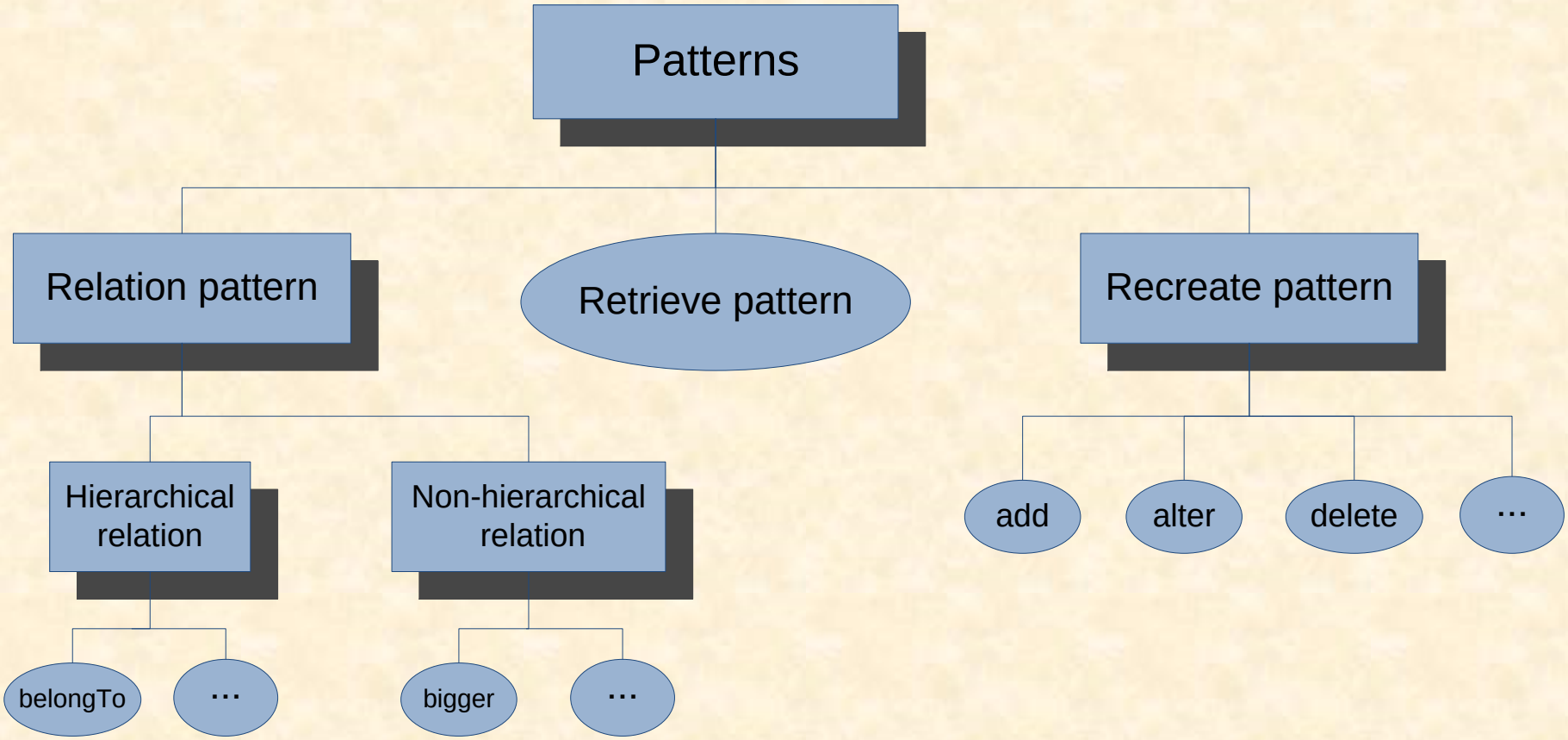
形式仕様パターンの事例:

name: belongTo
explanation: This pattern can be used to describe that an element is part of certain object
constituents: element, container
solution:

(dataType(element), dataType(container)) → formal expression

(T, **set of** T)	→	"element **inset** container"
(**set of** T, **set of** T)	→	"**subset**(element, container)"
(T, **seq of** T)	→	"element **inset elems**(container)"
(**seq of** T, **seq of** T)	→	"**exists**[i, j: **nat0**] | element = container(i, j)"
(T, **map** T **to** T')	→	"element **inset dom**(container)"
	…	

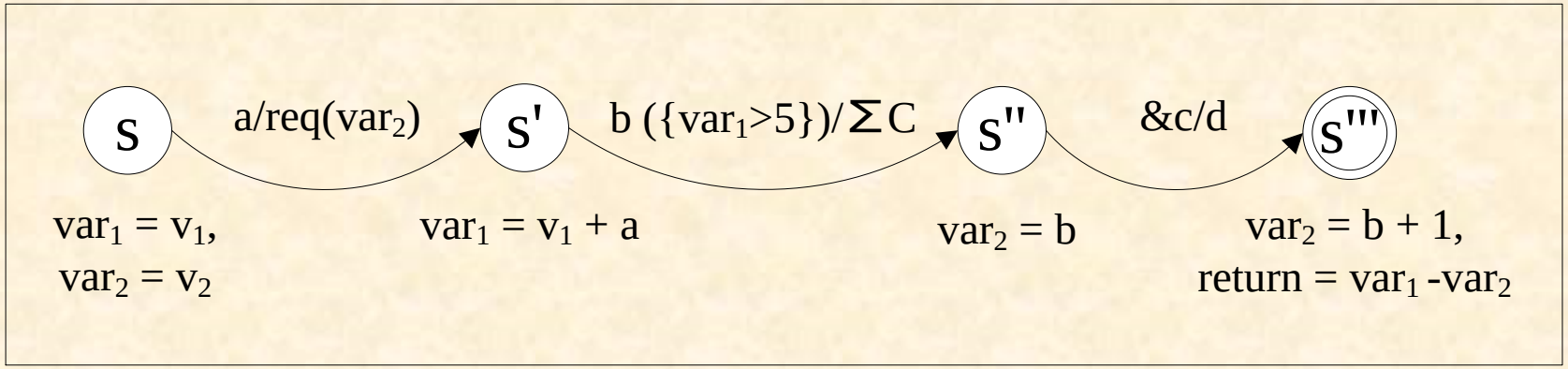
(2) 仕様パターンシステム



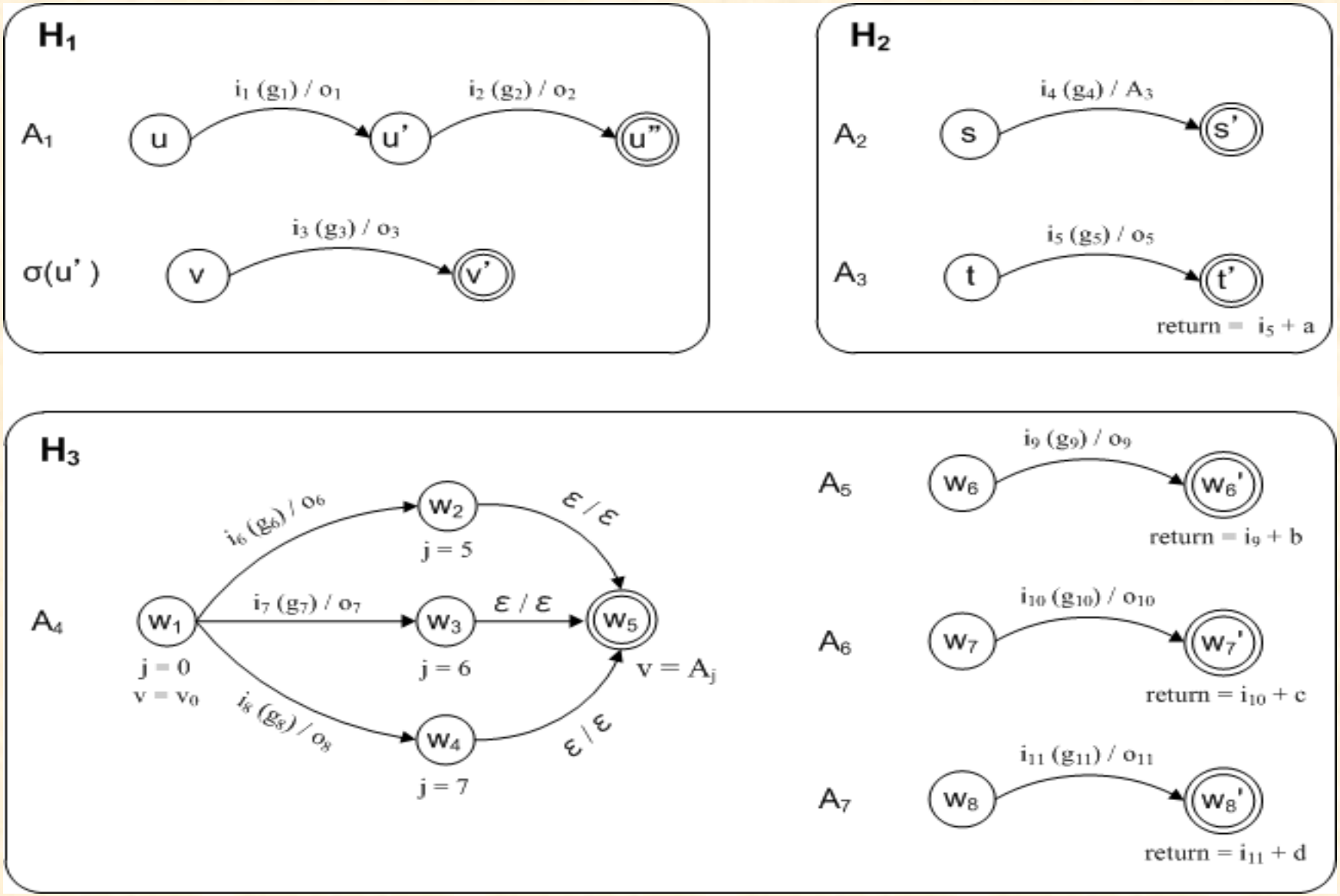
2.6 仕様パターン知識の表現方法、検索と適用アルゴリズム、および支援ツールのプロトタイプ

(1) 仕様パターン知識の表現方法:

有限状態遷移図でパターン知識を表現する。



階層的な有限状態遷移図で仕様パターンシステム知識を表現する。



(2) 仕様パターン知識の検索と適用アルゴリズム

仕様パターンの階層的な有限状態遷移図をXMLファイルに保存し、そのフォーマットによってパターン知識を検索したり、適用したりしています。

<table>
<tr>
<td>

```
<FSM name =""">
 <state type="initial">
  <name></name>
  <inf> </inf>
  <transition>
   <input></input>
   <output></output>
   <dest></dest>
  </transition>
 </state>
 <state type="accept">
  <name></name>
 </state>
</FSM>
```

</td>
<td>

```
<nestedFSM>
    <name></name>
    <para type =""">
     <name></name>
     <value>
      <para></para>
      <type></type>
      <value></value>
     </value>
    </para>
    <para>...</para>
</nestedFSM>
```

</td>
<td>

```
<inf type=""">
 <para></para>
 <type></type>
 <explain></explain>
 <value></value>
</inf>
```

```
<value>
 <nestedFSM></nestedFSM>
 <func></func>
 <forall></forall>
</value>
```

</td>
</tr>
<tr>
<td>an individual FSM</td>
<td>a low-level FSM</td>
<td>variable information on states and value information</td>
</tr>
</table>

```
String generateObjFromStr(String str, Module m, object currentObj){
    if(str is empty)
        return currentObj;
    else{
        if(currentObj == null){
            i = 0;
            while(str[0, i] is not an element name){
                i++;
            }
            set e as the element named str[0, i];
            return generateObjFromStr(str[i+2, str.length], m, e);
        }
        else if(currentObj is an element){
            preStr = str;
            if str contains "."
                set preStr as the string before the first "." of str;
            if(preStr == "value"){
                set v as the value of the element currentObj;
                return generateObjFromStr(str[preStr.length, str.length], m, v);
            }
            else if(preStr == "name"){
                set n as the name of the element currentObj;
                return generateObjFromStr(str[preStr.length, str.length], m, n);
            }
            else if(preStr == "type"){···}
            ···  // other possible values of preStr
        }
        else if(currentObj is a type){···}
        else if(currentObj is a variable){···}
        ··· // other situations
    }
}
```

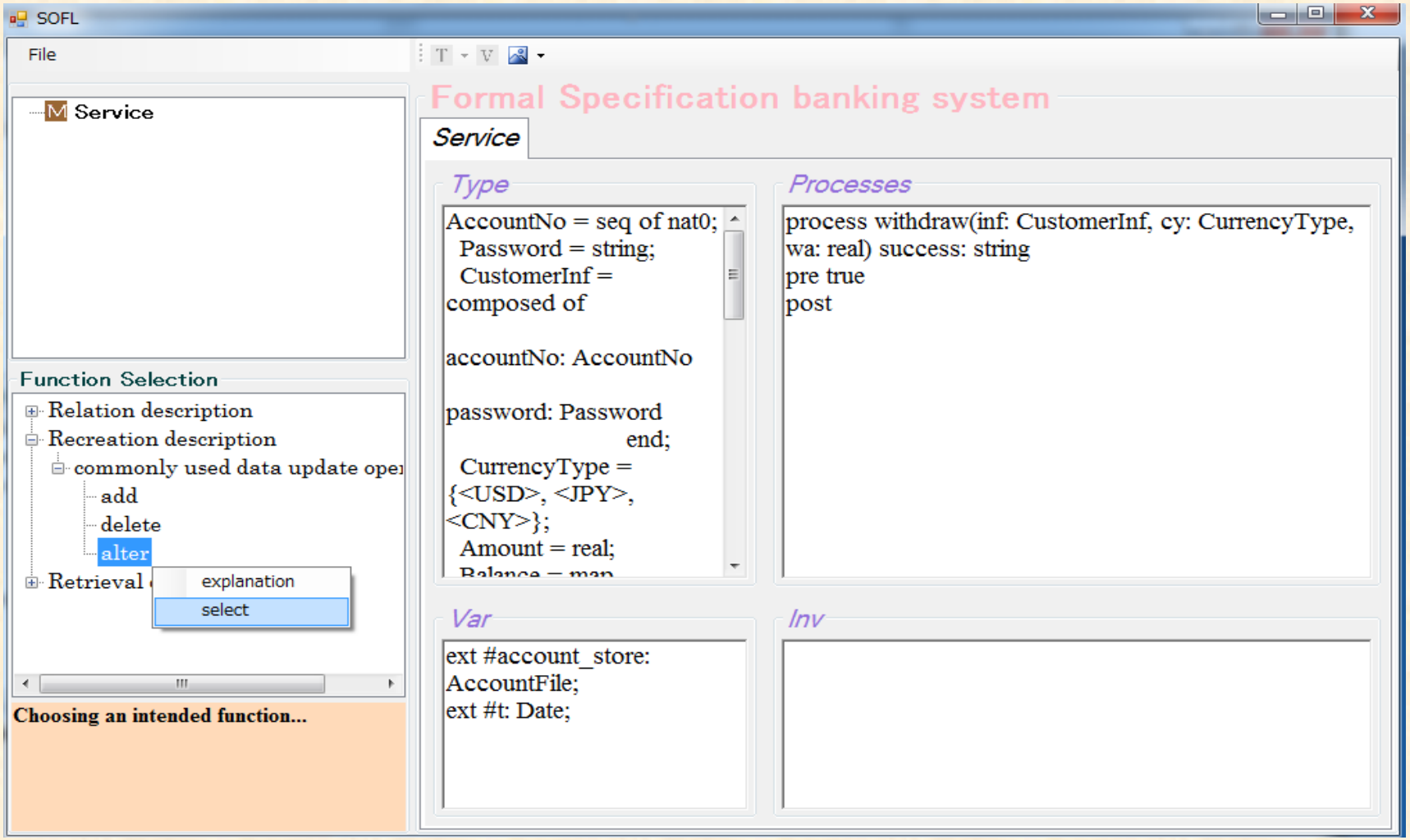
情報を検索する再帰アルゴリズム

Algorithm Utilization of Pattern knowledge represented in HFSM

```
cf = A_select; cs = q0_cf; input = return = null;
while(cs ∉ F_cf || states is not empty){
   set variable input as the input from the user;
   if(cs ∉ F_cf){
      ns = o = null;
      foreach (i, G) in Acc_cf(cs){
         if(i == input && ∀g ∈ G · g){
            ns = δ_cf(cs, (i, G)); o = λ_cf(cs,(i, G));}
      if(σ_HF(cs) ≠ ∅ || ∃v ∈ Vφ_cf(cs) · σ_HF(v)≠ ∅ || σ_HF(o)≠ ∅){
         states.push(ns);
         if(σ_HF(cs) ≠ ∅)
            set ns as the initial state of one of the FSMs in σ_HF(cs);
         if(∃v ∈ Vφ_cf(cs) · σ_HF(v)≠ ∅){
            for each v ∈ Vφ_cf(cs) that satisfies σ(v)≠ ∅{queue.push(v);}
            if(σ_HF(cs) = ∅){set ns as the initial state of
                              one of the FSMs in σ_HF(v) where v ∈ Vφ_cf(cs);}
         else{specify variables according to φ_cf(cs)}
         if(σ_HF(o)≠ ∅){
            queue.push(o);
            if(σ_HF(cs) = ∅ && σ_HF(o)= ∅){
               set ns as the initial state of a FSM in σ_HF(o);}}
         else{display o to the user;}
   else{
      if(return ≠ null ){
         replace the corresponding part in queue[top] with return;
         if(σ_HF(queue[top]) = ∅){
            temp = queue.pop();
            if(temp is output){display temp;}
            else{specify variables based on temp;}}}
   ns = states.pop();}
cs = ns; cf = A where cs ∈ Q_A;}
```

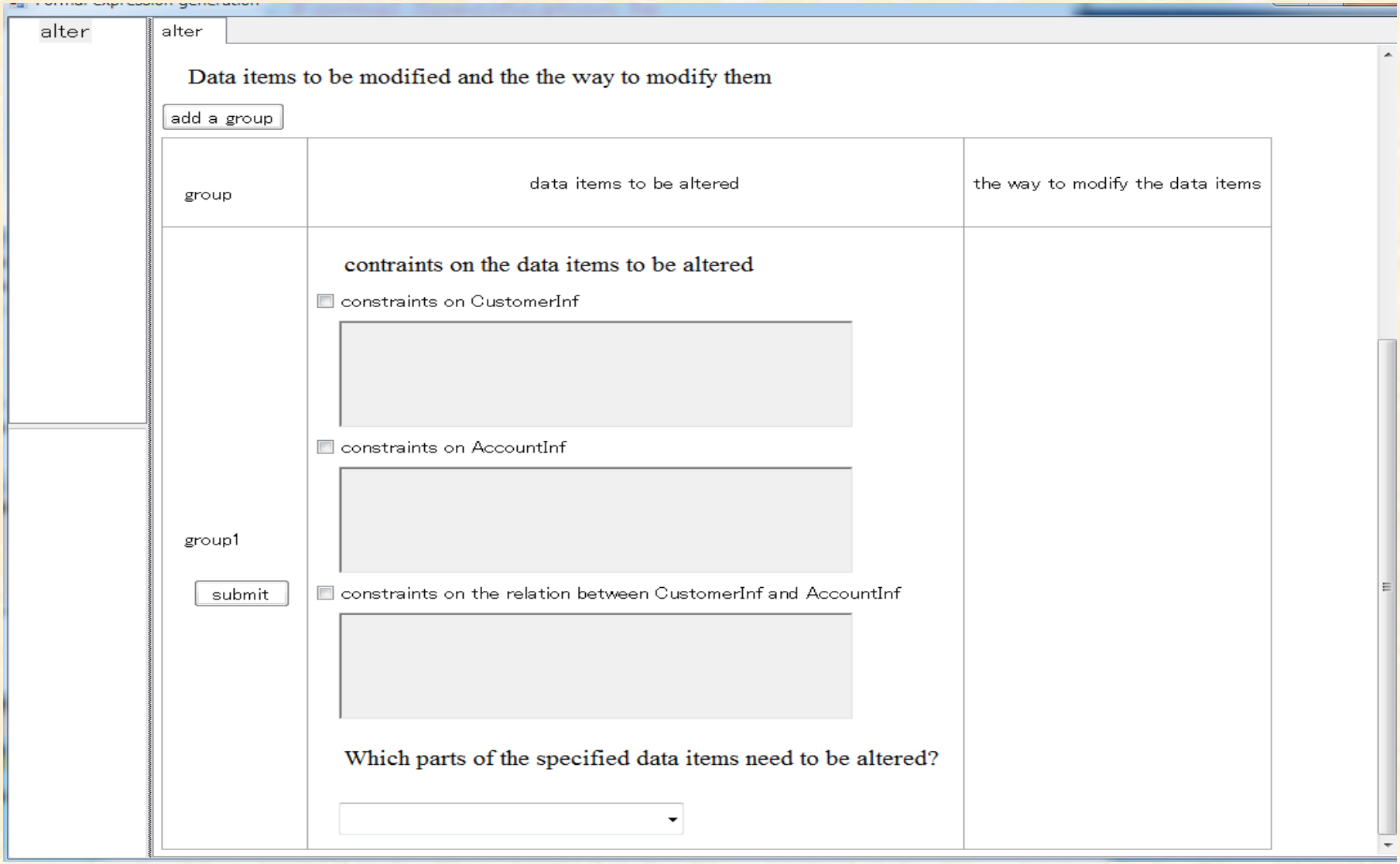
HFSMにおいて表現された仕様パターンを適用するアルゴリズム

(3) 仕様パターンに基づく形式仕様作成の支援ツール

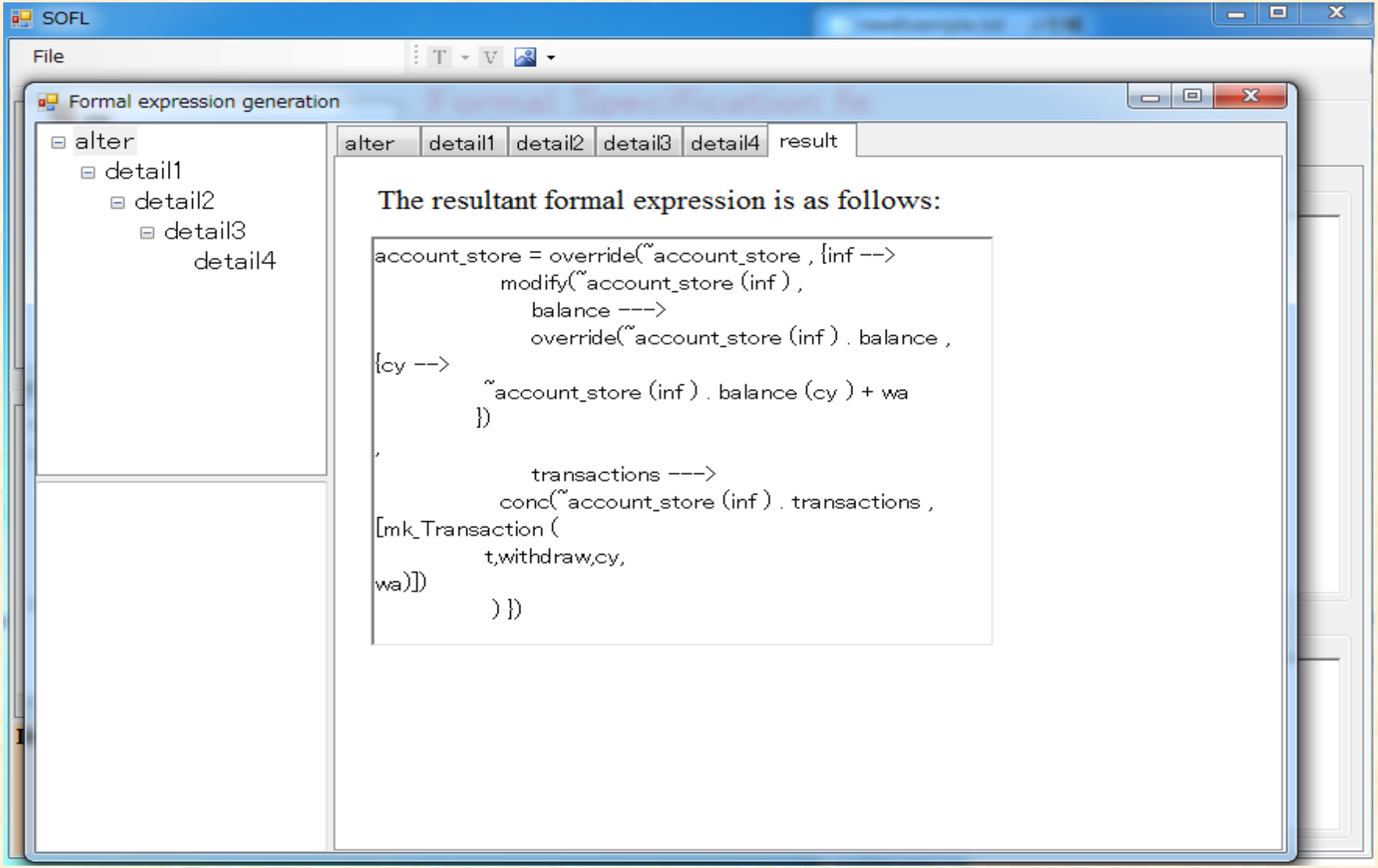


支援ツールのGUI

2012年度ソフトウェア工学分野の先導的研究支援事業



ガイダンスの図表現のスナップショット



作成された形式論理式のディスプレイ

３. 研究成果の評価

(1) 事例研究によって、本研究の研究成果の方向性と有効性を評価した。

（a）**JR東日本のSuicaカードシステム**

（b）**旅行会社システム**

（c）**スマート交通信号システム**

(2) 既存の手法との比較（分析のみ）

VDM、VDM＋＋、B-Method、Event-Bと比べて、SOFL三段階漸進的な形式記述技術は、開発現場で形式仕様を作成する具体的な手法を提供しています。

VDM＋＋、UML＋B-Method、UML＋OCLと　比べて、SOFL三段階漸進的な形式記述技術は、

開発現場で形式仕様を作成する具体的かつ簡単な手法を提供し、仕様アニメーションおよび仕様パターンに基づく形式仕様作成支援技術も提供しています。

Stepney研究グループが提案したZ仕様の作成に使える形式仕様パターンと比べて、本研究で開発した仕様パターン知識に基づく形式仕様作成支援技術は、コンピュータが仕様パターンを直接に使い、開発者が自然言語で要求された入力だけを提供することによって、形式仕様が自動的に作成される特徴をもっています。

Bumbulis 研究グループは、ユーザインタフェースの開発における形式手法とプロトタイピングを統合した手法を提案しました。これと比べて、本研究で開発した仕様アニメーション技術は、仕様からプログラムへの変換が必要ないまま簡単にアニメーションを実施することができます。

4. 今後の研究課題

(1) 非形式仕様アニメーションの支援ツール機能の強化

① SOFL非形式仕様からアニメーションシステムを構築する基礎とする「機能構造図」への変換

② 新たなコンポネットアイコンと機能テンプレートを非形式仕様アニメーションの支援ツールに自由に追加する機能

（2）SOFL形式仕様アニメーションの支援ツール機能の強化

（3）ドメイン限定仕様アニメーションの支援ツール

（4）SOFL形式仕様の構文と型分析

（5）プロセス機能シナリオの自動生成

（6）形式仕様アニメーションによる仕様トレーサビリティの構築と検証

（7）仕様トレーサビリティに基づく仕様のインスペクション技術

（8）形式仕様からプログラムへの自動変換技術