



一般社団法人 組込みシステム技術協会(JASA)

技術本部成果発表会

2013年5月22日



技術セミナーの活用方法

2013年5月22日
技術セミナーWG
富岡 理



- 1回/月ペースでの会合。
- セミナー企画の立案、他委員会からの企画案の運用。
- 広報委員会とメンバー共通で同時開催。
- JASA内での「文系の集まり」
 - ・ 営業、企画、総務、人事など
- メンバー
 - ・ アンドールシステムサポート、イーソル、SRA、グレープシステム、ステップワン、東横システム、ビッツ、ブライセン、ユークエスト



- HP/Bulletin JASAと併せて、JASAの対外的情報発信源とする
- 2005年6月に第1回。最新は第31回。
- JASAのビルの4階大会議室(最大80名)
- 基本体系
 - 13時半～17時半の半日コース
 - JASA会員6000円、非会員8000円
- 時間・価格とも柔軟に。(無料もあり)

今年の計画



■ 有料セミナー

- 6月25日 : IRC2013(日経BP共催) → **プラットフォーム
研究会企画**
- 7月25日 : HTML5関連セミナー **セミナWG**
- 9月5日 : ET設計検証トラックベスト3セミナー **セミナWG**
- 10月予定 : オープンソース関連 **セミナWG**

■ 無料セミナー

- 12月頃 : セキュリティ関連 → **安全性向上
委員会企画**
- 未定 : 未定

■ 地方セミナー

- 9月13日 : 札幌(形式手法) → **門田専務理事
北海道経産局
企画**
- 10月29日/30日 : 仙台 **セミナWG**



■ ロボットの場合

- ・ プラットフォーム研究会でOpenEL
- ・ ロボット関連企業やキーパーソンへのJASA活動の周知(講師招聘)
- ・ 周辺業界への周知(メディア活用)

■ セキュリティの場合

- ・ 安全性向上委員会でのCMSiS
- ・ 事例を含めたアピール
- ・ VSE(小規模事業者)向け開発プロセスとの連携(IPA/JISA)

各委員会から見たメリット



- JASAの名前を使える
 - ・ 他団体との連携(IPA、JISA、日経BP…)
 - ・ 講師依頼(飛び込みはちょっとキツい場合も)
- 受講者アンケート
 - ・ どのようなことに興味をもっているのか
 - ・ レベルは？



■ 実装品質強化WG

- ・ 非機能用件の定義の仕方いろいろ
- ・ 既存の研究、実際の企業でのやり方(話していく
れない?)

■ 安全性向上委員会

- ・ SSQの上流工程研究のオーバービュー
- ・ 委員会メンバーが聞きたい企画

■ アジャイル研究会

- ・ アジャイルで進めたプロジェクト事例集



委員会・WGからの企画
募集集中！！

WGメンバーも
募集集中！！



「タイトル」

20YY/MM/DD 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



要求の仕様化における 形式手法の適用検討

2013年05月22日
安全性向上委員会
株式会社クレスコ 豊 裕介
y-toyo@cresco.co.jp



1. 本成果の位置づけ
2. 要求の仕様化における課題
3. 形式手法とは
4. 要求の仕様化へのVDMの適用検討
 - 4-1. 仕様記述実験の題材
 - 4-2. 実験詳細
5. まとめ
6. 今後の課題
7. (参考)洗練後の仕様
8. 参考文献

1. 本成果の位置づけ



安全性向上委員会では、
「要求の仕様化に関する調査研究」として、以下のことを実施(予定)

- ①機能安全や情報セキュリティ関連規格における要求事項の洗い出し
(規格が何を要求しているか = 何を仕様化しなければならないか?)
- ②REBOKや要求工学等の専門書を調査、要求の仕様化プロセスと課題の整理
- ③モデルベース開発で使われている手法やツールの調査
- ④実際に使われている形式手法とそのツールの調査
- ⑤JASA会員を対象としたアンケートによる、
プロセス、手法やツール、課題に関する実態調査
- ⑥要求仕様を顧客と合意する為のアプローチの検討

⇒本成果は、④の延長線上

本日は、要求の仕様化に形式手法を適用した結果について発表

2. 要求の仕様化における課題



言ったことがうまく伝わってないなあ。

そんな事、確認しなくていいよ、常識で判断して。

発注者



ここは、逆に聞いて欲しかった。

分厚い仕様書貰っても、隅々まで確認できないよ。

動くモノが出来てから、再検討しよう。

要求

この表現だと、幾つかの解釈が出来るんだけど…。

不明点は確認すべきか、手を煩わせない為に、解釈すべきか？

受注者



意図が分かれば、仕様化しやすいんだけど…。

仕様書作っても、全部は読んでもらえないだろうなあ…。

あれ？似たようなことを幾つか書いてしまったけど、矛盾していないかな？

要求仕様

形式手法を適用することによって、これらの課題の幾つかを解決することができるか？

3. 形式手法とは？



「システム、特に、ソフトウェアの開発法であって、数理論理学に基づく科学的な裏付けを持つ。明確で厳密な意味を持つ言語を用いて設計対象を表現することにより、設計記述の正しさを系統的に示すことが可能になる。従って、開発したシステム、あるいは、ソフトウェアが高い信頼性を持つことを保証出来る。」^[1]

「品質の高いソフトウェアを効率よく開発するために、数学を基盤とした矛盾のない仕様書を書いて、それが正しいかどうかを検証する」^[2]

⇒システム・ソフトを開発する為の手法、数学ベース、仕様や設計を表現、厳密・矛盾のない記述、正しさの検証

上記と対極的な手法が、自然言語(日本語や英語などのように人間が日常的に用いている言語^[3])を用いて仕様書や設計書を作成する手法。自然言語を用いた場合は、記述内容に曖昧さや矛盾を含みやすいと言われている。



形式手法には、大きく分けて2種類の手法が存在する^[4]

(1) 記述手法

仕様を正確に記述することに特化した手法

例: **VDM**、Z記法、Bメソッド…など

(2) モデル検査

システムの振る舞い仕様を検証することを目的とした手法

例: Spin、LTSA、SMV…など

VDM、Spinに関しては、IPA/SECの適用実験^[5]にも使われ、
適用事例も多く有名

今回は要求の仕様化なので、VDMの適用を検討

VDMとは…^[6]

- ・ 1970年代にIBMのウィーン研究所で開発された形式手法
- ・ 記述言語として、VDM-SLとVDM++がある
- ・ VDM++はオブジェクト指向に基づいたモデル化が可能

VDM記述の構成要素^[7]

仕様の中で扱うデータに関する記述

データ型定義

状態(変数)定義

不变条件



不变条件とは、常に成り立つ
欲しい条件の事である。

例: 変数Aは10未満にならない

具体的な仕様を書く為の記述

引数

操作

戻り値

事前条件

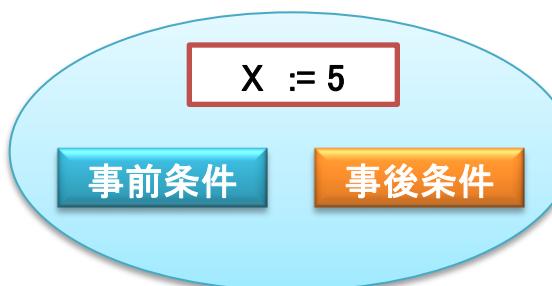
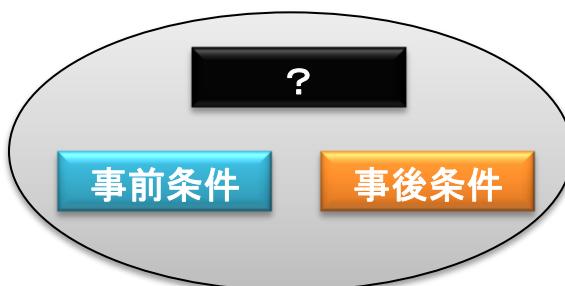
事後条件

事前条件とは、操作をする前に
成り立つ条件

事後条件とは、操作後に成り立つ
条件

※操作の他に関数というのもあるが、
今回は説明を省略

陰定義(仕様)と陽定義(仕様)^[8]



操作の書き方は、操作の中身を
ブラックボックスとして、
事前/事後条件のみを記述する
陰定義(仕様)と、
操作の中身を記述した陽定義(仕様)
の2つが存在する

4. 要求の仕様化へのVDMの適用検討



4-1. 仕様記述実験の題材

以下のような簡易な電気ポットを題材とする

電気ポットに関する商品企画部門からの要求

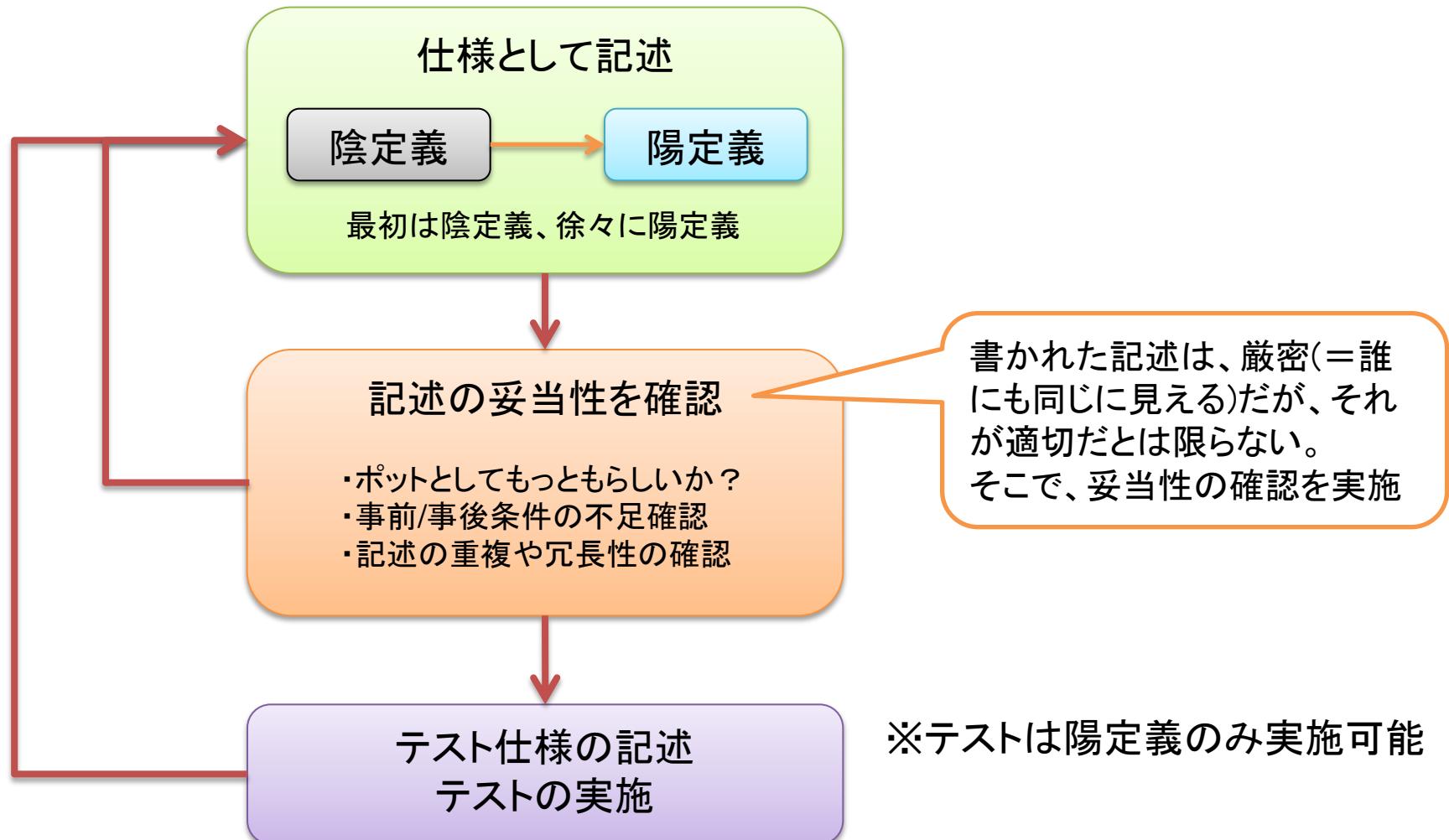
要求ID	要求事項
PR01	電気ポットの容量は2リットルとし、10°Cから沸騰するまでの時間は、15分以内とする。
PR02	電源コンセントをつなぐと、直ちにヒータで加熱を始め、沸騰したら、90°Cに保温する。
PR03	再沸騰ボタンが押されたら、再沸騰を始める。
PR04	水が加えられ、温度が低下したら、再沸騰を始める
PR05	保温中であれば、お湯を注ぐことができる。

上記内容をVDMで記述し、課題が解決出来るか？
また、VDMで記述したことで何が得られるか？

4-2. 実験詳細



仕様記述実験の流れは以下のようにした



流れの一例(PR02の場合)

PR02:コンセント接続 → 沸騰 → 90°C保温

```

PR02 : () ==> ()
PR02() == is not yet specified
pre
  P_Power = <OFF> and
  Heater = <OFF> and
  Keeping_Mode = false and
  Current_Temp < 100 and
  Pot_Capacity > 0
post
  P_Power = <ON> and
  Current_Temp = 90 and
  Keeping_Mode = true and
  Pot_Capacity > 0;

```

陰定義の記述

リファイン



```

PR02_1 : () ==> ()
PR02_1() ==

```

```

atomic(
  P_Power := <ON>;
  Heater := <ON>););
pre
  P_Power = <OFF> and
  Heater = <OFF> and
  Keeping_Mode = false and
  Current_Temp < 100 and
  Pot_Capacity > 0;

```

コンセント状態
(電源ON、ヒータON)

```

PR02_2 : () ==> ()
PR02_2() ==

```

```

atomic(
  Heater := <OFF>;
  Keeping_Mode := true););
pre
  P_Power = <ON> and
  Heater = <ON> and
  Keeping_Mode = false and
  Current_Temp = 100 and
  Pot_Capacity > 0;

```

沸騰
(ヒータOFF、
保温モード)

```

...

```

左の記述のように、まずは要求の事前条件と事後条件を定義

事前:コンセントが繋がっていない状態

事後:保温モードで90°C

→途中に存在する沸騰の状態の表現なし

記述を2つに分割して、沸騰の状態を表現

流れの一例(PR03の場合)

PR03:再沸騰ボタン押下 → 再沸騰

```
PR03 : () ==> ()
PR03() == is not yet specified
pre
  ? ? ?
post
  ? ? ?
```



```
PR03 : () ==> ()
PR03() == is not yet specified
pre
  P_Power = <ON> and
  Current_Temp < 100 and
  Keeping_Mode = true
post
  Heater = <ON> and
  Keeping_Mode = false;
```

再沸騰ボタン押下前のあるべき姿を考えると、
電源ON、100°C未満、保温モード中が
妥当であると考えた。(保温モード以外は100°C
に沸騰させようとしているという考え方)

更に、具体的な操作を考えると、
それは事後条件の内容と一致するという
ことがわかったので、事後条件を操作
として記述



```
PR03 : () ==> ()
PR03() ==
  atomic(
    Heater := <ON>;
    Keeping_Mode := false;
  )
pre
  P_Power = <ON> and
  Current_Temp < 100 and
  Keeping_Mode = true;
```



流れの一例(PR01とPR02の比較)

PR01: 10°C、2L → 15分で沸騰

PR02: コンセント接続 → 沸騰 → 90°C 保温

```
PR01_1 : () ==> ()  
PR01_1() ==  
  Heater := <ON>  
pre  
  Heater = <OFF> and  
  Keeping_Mode = false and  
  Current_Temp = 10 and  
  Time = 0 and  
  Pot_Capacity = 2;
```

```
PR01_2 : () ==> ()  
PR01_2() ==  
atomic(  
  Heater := <OFF>;  
  Keeping_Mode := true;)  
pre  
  Heater = <ON> and  
  Current_Temp = 100 and  
  Time < 15 and  
  Pot_Capacity = 2;
```

```
PR02_1 : () ==> ()  
PR02_1() ==  
atomic(  
  P_Power := <ON>;  
  Heater := <ON>;)  
pre  
  P_Power = <OFF> and  
  Heater = <OFF> and  
  Keeping_Mode = false and  
  Current_Temp < 100 and  
  Pot_Capacity > 0;  
  
PR02_2 : () ==> ()  
PR02_2() ==  
atomic(  
  Heater := <OFF>;  
  Keeping_Mode := true;)  
pre  
  P_Power = <ON> and  
  Heater = <ON> and  
  Keeping_Mode = false and  
  Current_Temp = 100 and  
  Pot_Capacity > 0;
```

PR01とPR02を記述していくと、共通部分が多いことに気付く
→(自然言語上でもわかることだが)PR01は、
具体的な数値で、性能に関する要件を表している。

以上の事から、PR01に関するVDM記述は削除した。
→(少なくとも)性能要件のようなものの記述には不向き。
→自然言語との共生が必要

流れの一例(テスト仕様の記述/実施)



```
-- 初期:電源OFF、ヒータOFF、10°C、非保温モード、2L
print debug_Init(<OFF>, <OFF>, 10, false, 2)
-- 電源ON
print Power_On()
-- ヒータON
print Not_keeping_Heater_On()
-- 沸騰
print debug_Water_Temp_Change(100)
-- 保温モードに移行
print Change_Keeping()
-- 保温下限まで低下
print debug_Water_Temp_Change(88)
-- 下限に達したのでヒータON
print Keeping_Heater_On()
-- 保温上限まで上昇
print debug_Water_Temp_Change(92)
-- 上限に達したのでヒータOFF
print Keeping_Heater_Off()
...
```

左記は、記述した仕様を動かす為に記述したテストケース

電源OFF、ヒータOFF、10°C、2Lの水がポットに入った状態からの変化を記述している。

この変化の過程において、不变条件、事前条件、事後条件を逸脱することがあれば、ツールがその時点でエラーと判定してくれる。

つまり、検証を行う事が出来る。

以上のような流れを繰り返し、仕様を洗練化させていった。
(参考として、洗練化した仕様は「7. 洗練化後の仕様」として掲載しておく。)

5. まとめ



- ・ 書かれたことは厳密である(自然言語のような表記ゆれも許されない)
- ・ 不変条件、事前条件、事後条件という枠組みに落として考えるので、適切な仕様を記述出来る可能性が高くなる。(自然言語に適用するのもあり)
- ・ 仕様を動かして検証出来る部分は、自然言語に対するアドバンテージ

上記のような良いことばかりではない。

- ・ 今回の性能要件のように、自然言語で書いた方が良いものも存在する。
→自然言語との棲み分け(=共生)をどのようにしていくか？
- ・ 厳密さの追求 = 顧客の理解の妨げ、追求しない = 受注者が後工程で苦労
(PR03 再沸騰ボタン押下 ⇒ 保温モード解除、ヒータON)
- ・ 仕様は動かせるが、それは「1つのテストケース」にすぎない
→網羅的な検証が必要(=今回説明していない、モデル検査の得意分野)
- ・ 記述の妥当性確認の属人性(ドメイン知識、個人の観察力/洞察力)
→記述の妥当性確認で気付いた項目を、他の手法を使って、出来るだけ解消する

6. 今後の課題



(1) 顧客との合意方法(=仕様書としての記述方法)の検討

- ・ VDMの記述をそのまま見てもらうのは難しい
- ・ 厳密さを追求しつつ、理解もしやすく

(2) 他の手法との組み合わせ

- ・ 仕様の妥当性確認における属人性を出来るだけ減らす
- ・ 自然言語の得意分野、形式手法・その他の手法の得意分野

(3) 実環境に近い状況での実験

7. 洗練後の仕様(掲載しやすいうように加工してあります)



```

module pot
  exports all

  definitions
  types -- 型定義ブロック
  OnOff = <OFF> | <ON>;
  Water_Temp = nat   inv WT == WT <= 100;
  Capacity = nat   inv C == C <= 2;

  values -- 値定義ブロック
  Kepp_Temp : Water_Temp = 90;    -- 保温温度
  Boiled_Temp : Water_Temp = 100;   -- 沸騰温度
  Reboil_Temp : Water_Temp = 88;    -- 再沸騰させる為の温度
  Keep_Max_Temp : Water_Temp = 92; -- 保温中、ヒータOFFにする温度

  -- 状態定義ブロック
  state Pot of
    P_Power : OnOff           -- コンセントの状態
    Heater : OnOff            -- ヒータの状態
    Current_Temp : Water_Temp -- 現在温度
    Keeping_Mode : bool       -- 保温モードか？
    Pot_Capacity : Capacity  -- ポットの中身の量

  inv p == (p.P_Power = <OFF> => p.Heater = <OFF>)
  init p == p = mk_Pot(<OFF>, <OFF>, let x in set {1,...,100} be st
  true in x, false, 2) end

```

```

operations -- 操作定義ブロック

Power_On : () ==> ()      -- 電源ON
Power_On() ==
  P_Power := <ON>
pre
  P_Power = <OFF>;

NotKeeping_Heater_On : () ==> ()
NotKeeping_Heater_On() ==
  Heater := <ON>
pre
  P_Power = <ON> and
  Current_Temp < 100 and
  Pot_Capacity > 0 and
  Keeping_Mode = false;

Change_Keeping : () ==> ()
Change_Keeping() ==
  atomic(
    Heater := <OFF>;
    Keeping_Mode := true;
  )
pre
  P_Power = <ON> and
  Current_Temp = 100 and
  Pot_Capacity > 0 and
  Keeping_Mode = false;

```



```
Keeping_Heater_On : () ==> ()
Keeping_Heater_On() ==
  Heater := <ON>
pre
  P_Power = <ON> and
  Heater = <OFF> and
  Keeping_Mode = true and
  Current_Temp = 88;

Keeping_Heater_Off : () ==> ()
Keeping_Heater_Off() ==
  Heater := <OFF>
pre
  P_Power = <ON> and
  Heater = <ON> and
  Keeping_Mode = true and
  Current_Temp >= 92;

Keeping_Off : () ==> ()
Keeping_Off() ==
  Keeping_Mode := false
pre
  P_Power = <ON> and
  Current_Temp < Reboil_Temp and
  Pot_Capacity > 0 and
  Keeping_Mode = true;
```

```
Push_Reboil_Botton : () ==> ()
Push_Reboil_Botton() ==
  atomic
    Heater := <ON>;
    Keeping_Mode := false;
pre
  P_Power = <ON> and
  Current_Temp < 100 and
  Keeping_Mode = true;

Add_Water : () ==> ()
Add_Water() ==
(
  dcl capacity : Capacity := let y in set {0,...,2} be st y >
  Pot_Capacity in y;
  Pot_Capacity := capacity;
)
pre
  Pot_Capacity < 2;
```



```
Use_Water : () ==> ()
Use_Water() ==
(
  dcl capacity : Capacity := let y in set {0,...,2} be st y <
  Pot_Capacity in y;
  if capacity = 0
  then (Heater := <OFF>; Keeping_Mode := false)
  else skip;
  Pot_Capacity := capacity;
pre
  P_Power = <ON> and
  Keeping_Mode = true and
  Current_Temp >= Reboil_Temp and
  Pot_Capacity > 0;
```

```
-- 以下、デバッグ用
-- 初期化メソッド
debug_Init : OnOff*OnOff*Water_Temp*bool*Capacity ==>
()
debug_Init(p_power, heater, current_temp, keeping_mode,
pot_capacity)==
  atomic(
    P_Power    := p_power;
    Heater     := heater;
    Current_Temp := current_temp;
    Keeping_Mode := keeping_mode;
    Pot_Capacity := pot_capacity);

-- 水温変更メソッド
debug_Water_Temp_Change : Water_Temp ==> ()
debug_Water_Temp_Change(water_temp) ==
  Current_Temp := water_temp;
end pot
```

8. 参考文献



- [1] 中島 震, ソフトウェア工学の道具としての形式手法-彷徨える形式手法-,
Technical Report NII-2007-007J, National Institute of Informatics, 2007.
(<http://www.nii.ac.jp/TechReports/07-007J.pdf> からも入手可能)(2013/5/17最終アクセス)
- [2] 独立行政法人 情報処理推進機構 2012 形式手法活用ガイド 導入の手引き p1
(http://sec.ipa.go.jp/reports/20120928/reports_20120928_01.zip から入手可能)
(2013/5/17最終アクセス)
- [3] IT PRO 情報・通信用語辞典 自然言語
(<http://ipro.nikkeibp.co.jp/word/page/10000688/>) (2013/5/19最終アクセス)
- [4] @IT MONOist 誰でも使える形式手法(1):ライトウェイトな形式手法で高品質な仕様をこの手に!
(<http://monoist.atmarkit.co.jp/mn/articles/0809/17/news125.html>) (2013/5/17最終アクセス)
- [5] 独立行政法人 情報処理推進機構
現場への形式手法導入課題を解決する「形式手法活用ガイドならびに参考資料」の公開
(<http://sec.ipa.go.jp/reports/20120928.html>) (2013/5/17最終アクセス)
- [6] 石川冬樹(著), 荒木啓二郎(監), VDM++による形式仕様記述, まえがきpx, 近代科学社, 2011
- [7] 石川冬樹(著), 荒木啓二郎(監), VDM++による形式仕様記述, p14~15, 近代科学社, 2011
- [8] 石川冬樹(著), 荒木啓二郎(監), VDM++による形式仕様記述, p59~62, 168~169, 近代科学社, 2011

ご清聴ありがとうございました



「要求の仕様化における形式手法の適用検討」

2013/05/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



2012年度 状態遷移設計研究会成果報告

「状態遷移表設計におけるSPL/E実践ガイド」 の紹介

2013年5月22日
状態遷移設計研究会
キヤツツ株式会社 竹田彰彦



© Japan Embedded Systems Technology Association 2013



1. 2012年度、設計手法調査アンケート集計結果

2. 「状態遷移表によるSPL-E実践ガイド」概説

- ◆ソフトウェアプロダクトライン開発(SPL-E)

- ◆状態遷移表設計手法

- ◆フィーチャモデルとアーキテクチャのマッピング

- ◆リファクタリングによる干渉の低減

3. 2013年度、新規テーマの紹介

1. 研究会の趣旨、活動内容



■ 状態遷移表の良さを伝え普及していくこう！

⇒**状態遷移の検討漏れに気づきやすいという特性**を持つ表形式の設計手法を普及させることで、組み込みソフトウェア業界の発展に寄与することを目的として活動を進めています。

1. 1 平成19年度からの取り組み(毎年)

設計手法、ツールの普及度のアンケート調査

⇒組込み総合技術展/Embedded Technology(JASAブース)

1. 2 平成21年度までの取り組み(終了)

⇒状態遷移表の標準的な記法の定義、公開

1. 3 平成22年度からの取り組み(公開中)

⇒設計手法の解説書の執筆と公開

「**状態遷移表設計におけるSPL/E実践ガイド**」





1. 1 アンケート調査について



実施したアンケートの内容（抜粋）

以下に、実施したアンケートの内容を示す。

4. あなたが採用してゐる、今後採用してみたい方法論は？	- 構造化技法(構造化プログラミング)	- オブジェクト指向設計	- その他()
------------------------------	---------------------	--------------	----------

5. あなたが採用してゐる、今後採用してみたいソフトウェア開発プロセスは？	- ウォーターフォール(V字型モデル)	- プロトタイプモデル	- その他()
	- スパイラルモデル(反復型、インクリメンタル、アジャイル(XP, scrum)他)		- その他()

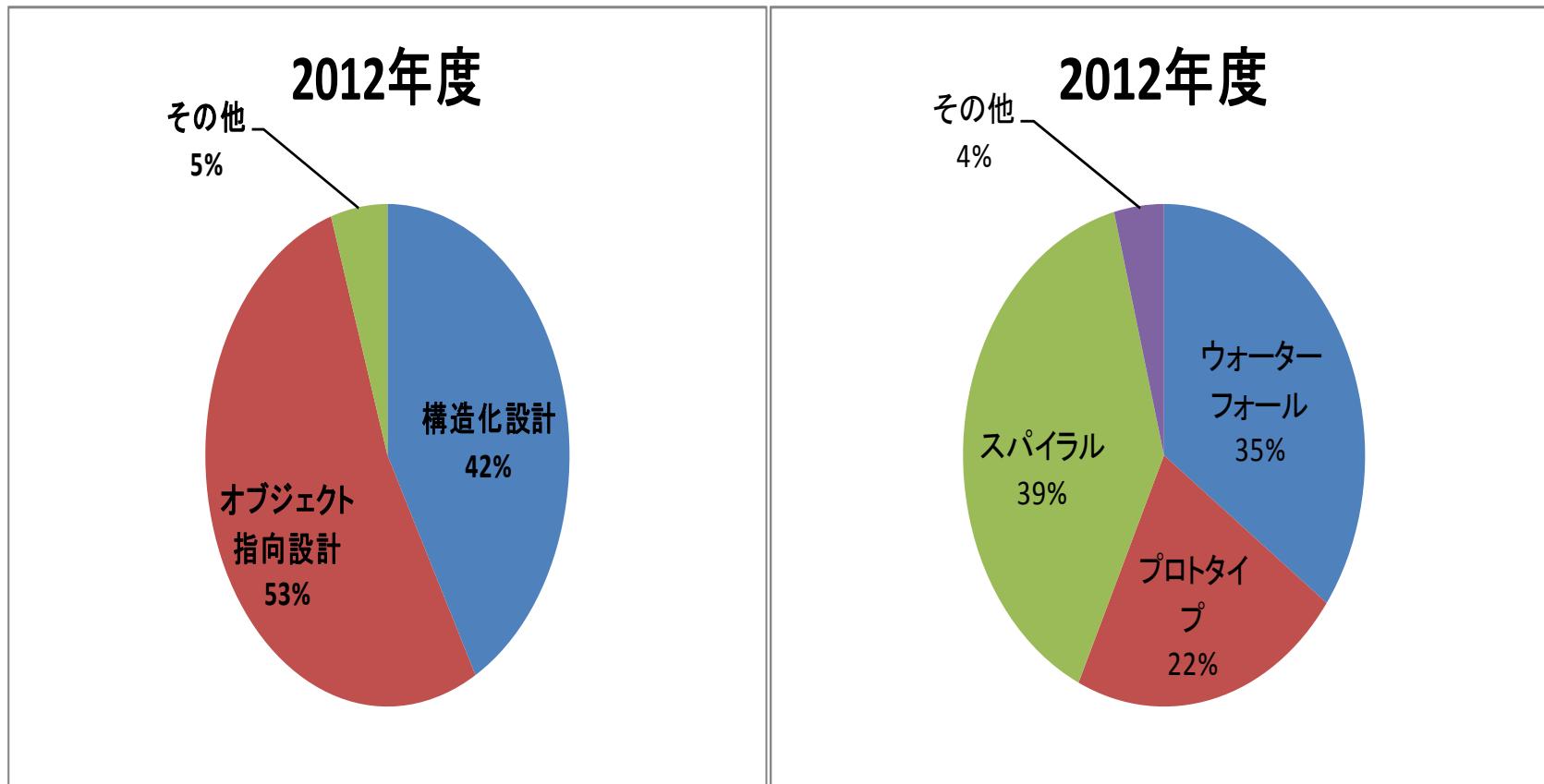
6. あなたが要求定義、アーキテクチャ設計、詳細設計の設計工程で適用してゐる、今後適用してみたいダイアグラムは？	UML系ダイアグラム			
	構造図	- 配置図	- 複合構造図(コンポジット構造図)	- コンポーネント図
		- クラス図	- オブジェクト図	- パッケージ図
	振る舞い図	- ユースケース図	- ステートチャート(状態遷移図／状態遷移表)	- アクティビティ図
	相互作用図	- 相互作用概要図	- コミュニケーション図(コラボレーション図)	- シーケンス図
	sysML系ダイアグラム	- 要求図	- 内部ブロック図	- パラメトリック図
	データフロー系	- DFD		
	実体関連	- E-R図		
	その他	- USDM(XDDP)	- フィーチャ図(ソフトウェアプロダクトライン)	- その他()

7. あなたが要求定義、アーキテクチャ設計、詳細設計の設計工程で使用してゐる、今後採用してみたいCASEツールは？	要求定義			
		- Doors(IBM/Rational)	- RequisitePRO(IBM/Rational)	- RaQuest(SparxSystems)
		- Caliber(マイクロフォーカス/Boland)	- Mindmap(フリー)	- その他()
	UML/sysML系設計ツール	- Rhapsody(IBM/Rational)	- ROSE(IBM/Rational)	- Software Architect(IBM/Rational)
		- Enterprise Architect(SparxSystems)	- Together(マイクロフォーカス/Boland)	
		- Astah*/JUDE(エンジビジョン)	- EclipseUML(フリー)	- その他()
	連続系	- Matlab/Simulink(Mathworks)	- ASCET(ETAS)	- その他()
	状態遷移系	- ZI PC/ZI PC++(CATS)	- visual STATE(IAR)	- その他()
	その他	- その他、内製ツール、フリーソールなど()		





(1) 方法論と開発プロセスについて





今年度の調査で、方法論、開発プロセスを調査した。
方法論では、構造化設計とオブジェクト指向設計の割合がほぼ半々、むしろ
オブジェクト指向設計が主体となりつつある傾向を示している。

また、開発プロセスにおいては、**スパイラルモデルが主体**となりウォーターフォールモデルを逆転しており、プロトタイプモデルと合わせると、60%強が非ウォーターフォールモデルを採用している。

この傾向は、要求が決まらない、要求の変更が多々発生し手戻り作業が起こるなど、生産性の低下や納期遅延、コスト超過といった**ウォーターフォールの欠陥を回避するための取り組み**が行われていることを示していると思われる。

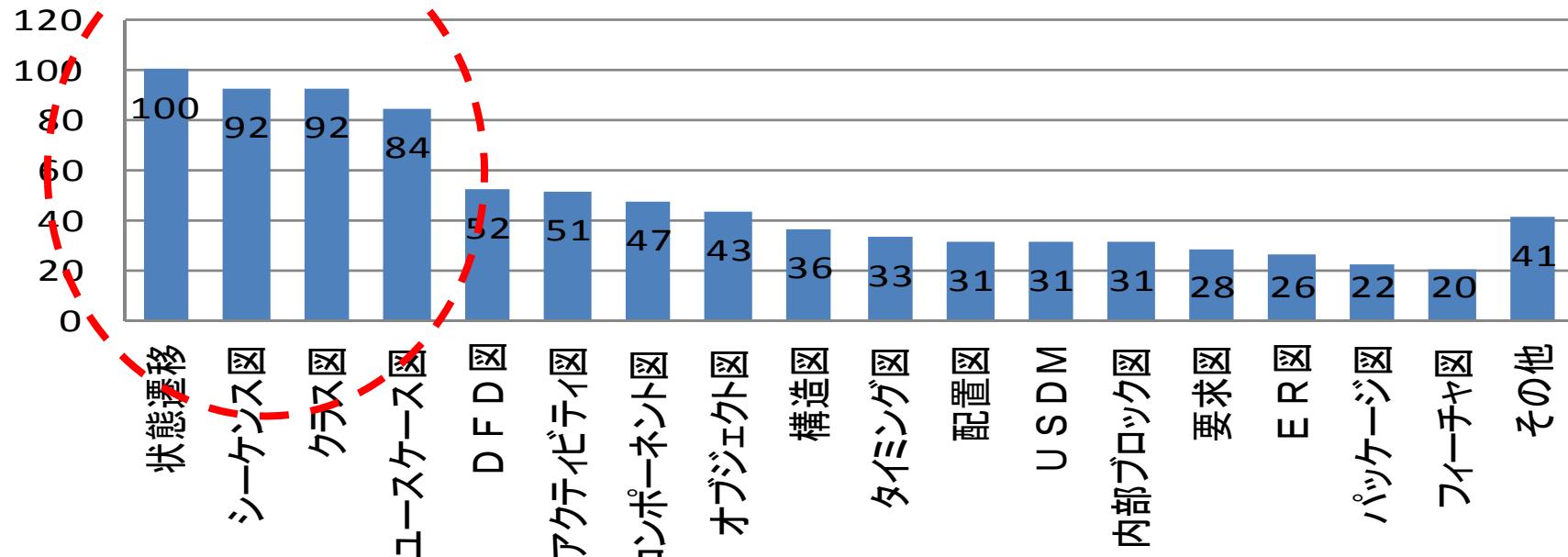
また、スマートフォンアプリの画面設計やMMIなど、スパイラルモデルが必要な開発案件が拡大している傾向も大きな要因と思われる。

また、その他の方法論、プロセスでは**XDDP、SPLE**などの差分・派生開発、系列製品開発を採用している。



(2) 表記法(設計書)について

使用している表記法





現在使用している表記法からは、構造はクラス図、振る舞いはユースケース図と状態遷移、相互作用はシーケンス図の採用が標準的な傾向と見えるが、他の表記法についても、平均的に使用されており、ドメインの特性や、採用プロセスに応じた表記法を取捨選択して採用していると思われる。

今後は、以下の観点でのより詳細な分析が必要と思われる。

- ①ドメインの特性と採用プロセスの把握
- ②プロセス、工程毎のダイヤグラムの適用傾向の把握
- ③ダイヤグラムの組み合わせ傾向の把握



1. 3 「状態遷移表設計におけるSPL/E実践ガイド」 の取り組みについて



■ソフト開発の課題と傾向

⇒品質の高いソフトウェア開発は本質的に困難。

早期の市場投入(短納期)、高い顧客満足度の実現。

⇒<要素技術>はあり、一定の成果を上げている。

オブジェクト指向、コンポーネント・ベース開発、フレームワークの採用、
繰り返し型、テスト駆動、ユースケース駆動など

しかしながら、<全体構想>の欠如が鮮明となってきた。



■SPEが注目されてきている。

⇒品質、生産性の向上施策として、**系列製品開発における**

再利用の拡大。共通部、可変部の分離。

⇒**アーキテクチャの構築という長期的な視野を重視**

組込ソフトウェアだけでなく、アーキテクチャの構築が有効とされる大規模システム
、長期間の利用が前提となる企業システムへの適用が試みられている。





状態遷移表設計の普及・啓蒙活動にあたり、系列製品開発の体系だった開発手法と融合させたプロセスを定義することで、
再利用しやすい状態遷移モデルを構築し、品質・生産性の向上に貢献する。



■ 2012年度のテーマ

「**状態遷移表設計におけるSPL-E実践ガイド**」の策定
本ガイドを参照することで、

⇒再利用しやすい設計を導出する手法が理解できる。





状態遷移表設計におけるSPL/E実践ガイド Ver. 1.0

一般社団法人 組込みシステム技術協会 技術本部 状態遷移設計研究会 編著



はじめに	… 2	3章 SPL開発プロセス【プロセス定義編】	… 41
目的	… 3	3.1 本ガイドのプロセス定義範囲	… 42
		3.2 SWP2のタスク構成の変更点	… 43
1章 SPLと状態遷移表設計【概念編】	… 5	あとがき	… 52
1.1 ソフトウェアプロダクトライン開発	… 6	付録	… 53
1.2 モデル指向開発と状態遷移表設計	… 11	付録1 用語	… 54
1.3 状態遷移表設計のSPL導入	… 15	付録2 参考情報・文献	… 56
2章 SPL開発プロセス【解説編】	… 16		
2.1 プロセスの概要	… 17		
2.2 フィーチャ指向分析(要求モデリング)	… 20		
2.3 要求仕様からのシナリオ導出	… 21		
2.4 シナリオからのシーン抽出	… 22		
2.5 状態の導出	… 24		
2.6 SPL構造設計	… 25		
2.7 リファクタリング	… 29		
2.8 自動販売機でのSPL実践例	… 30		





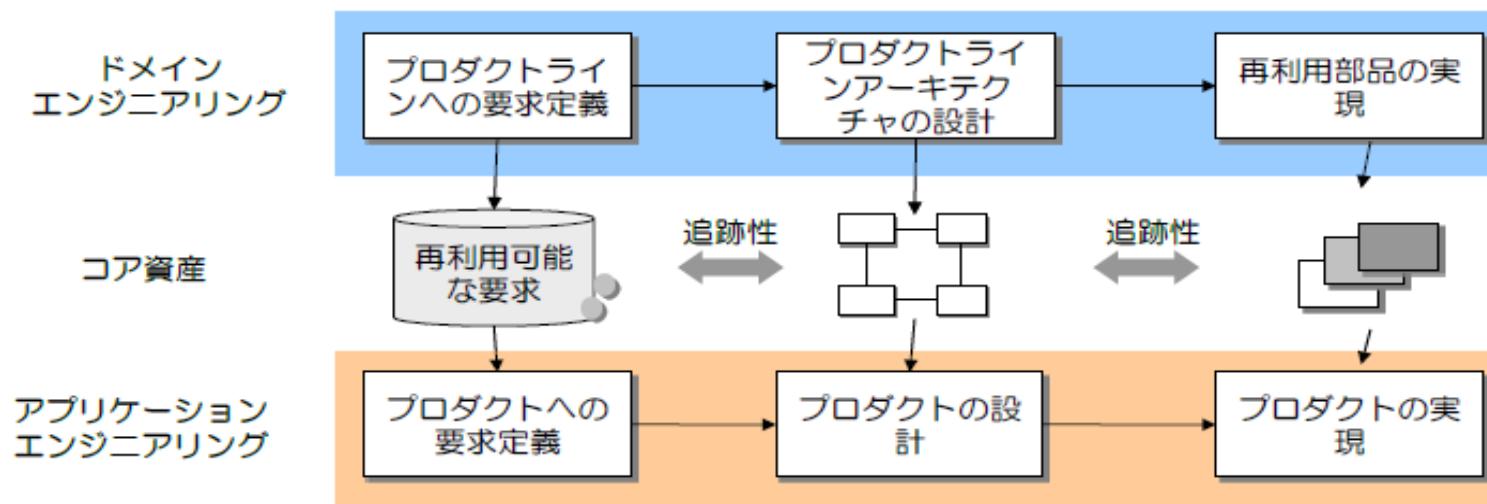
- SPLEとは?
⇒米国カーネギーメロン大学ソフトウェア工学研究所で開発
- 特長
⇒①アーキテクチャの構築という長期的な視野を重視
②**共通性と可変性(Commonality and Variability)**に注目
　　フィーチャモデルで共通部と可変部を分離
③**系列製品開発向け**に複数回のプロジェクトで
　　アーキテクチャを資産化、再利用する。
④**2層開発体制**(ドメイン・エンジニアリングとアプリケーション・エンジニアリング)
　　アーキテクチャ構築のための開発プロセスと
　　アーキテクチャが提供するフレームワークやコンポーネント
　　を利用したプロダクトの開発プロセスを分離



SPLEの2層開発体制

2つの段階: ドメイン & アプリケーション

- ドメインエンジニアリング: 再利用のための開発
 - 再利用範囲の定義、再利用可能な部品の特定
 - 適切な汎用性レベルの特定
- アプリケーションエンジニアリング: 再利用による開発
 - 共通フィーチャの利用
 - 選択的フィーチャとオプションフィーチャの統合



Copyright (C) 2007 National Institute of Informatics, All rights reserved.



■ フィーチャとは？

⇒**ユーザが識別できるシステムの目立った特性(Characteristics)。**

機能や品質上の要求の集合として表現される振る舞いの論理的単位。

「機能」や「非機能特性」。



■ フィーチャモデルとは？

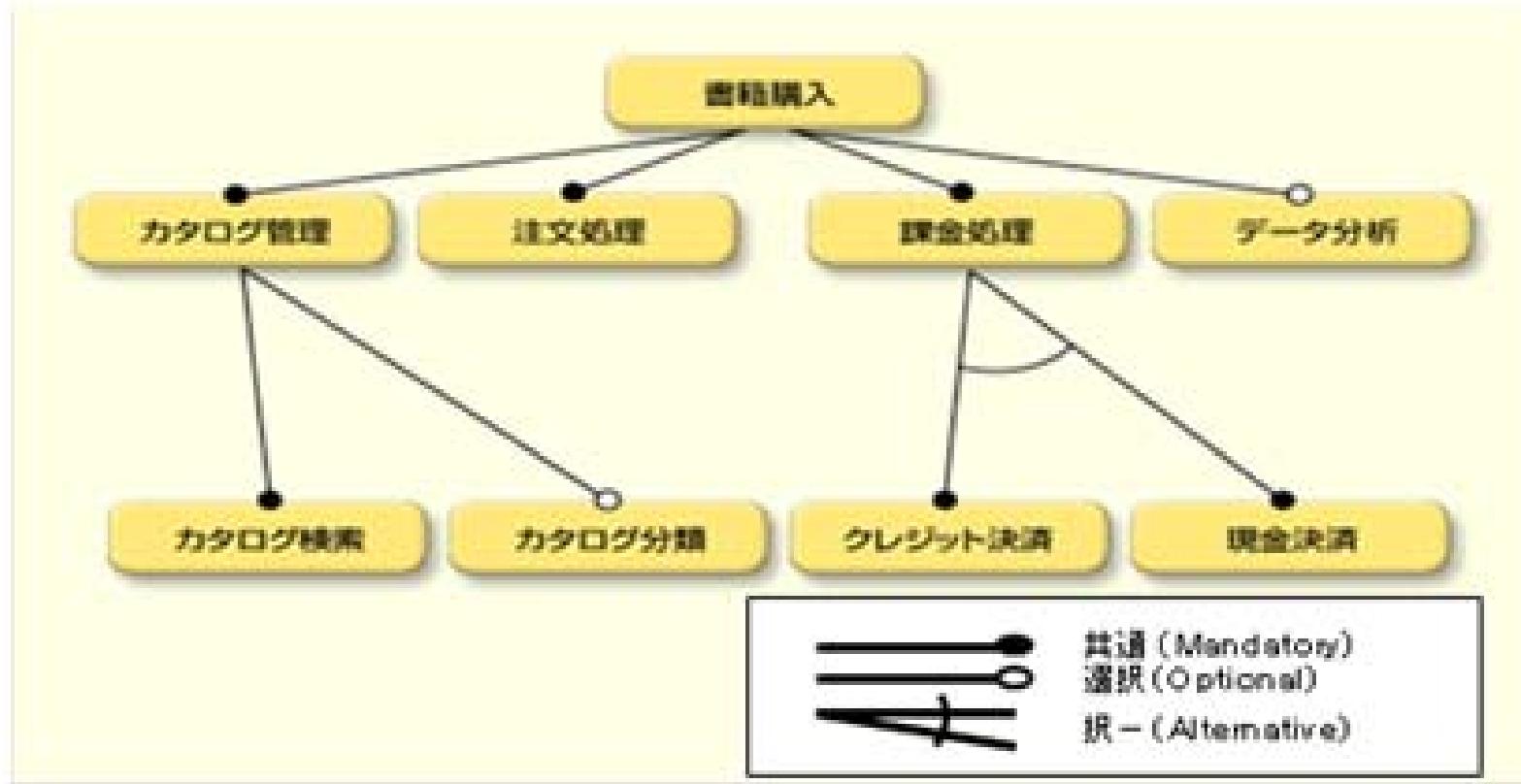
⇒**従来の開発プロセスの多くは、長期的視点と短期的視点が明確に分離できていない。**

ソフトウェア・システムの各構成部分における要求や技術の変化のスピードの違いをまとまりとして管理していない。

⇒**長期／短期の視点の違いを分析し、フィーチャを共通・選択・択一に分離、ツリー構造図で階層的に表現したモデルが「フィーチャモデル」。**



書籍購入webサイトのフィーチャモデルの例



次世代開発基盤技術"Software Factories" 説明より



■ モデリングとは？

設計品質向上のスタートポイント。

正確で曖昧性の少ない記述方法に基づいて設計すること。

■ モデリングのメリット

- ・正確な設計
- ・レビュー効率向上
- ・コミュニケーションの円滑化
- ・標準化
- ・設計、検証作業のツール化

■ モデリングの対象



モデリング対象	意味	ダイアグラム
機能	入力と出力の関係で表すモデル	ユースケース図、データフロー図
構造	概念、データ、情報の関係で表すモデル	クラス図、コンポーネント図、パッケージ図、配置図など
振る舞い	入力イベントに対して、処理と出力イベントの関係で表すモデル	アクティビティ図、シーケンス図、タイミング図、コミュニケーション図、状態遷移図、状態遷移表など



■ 状態遷移設計

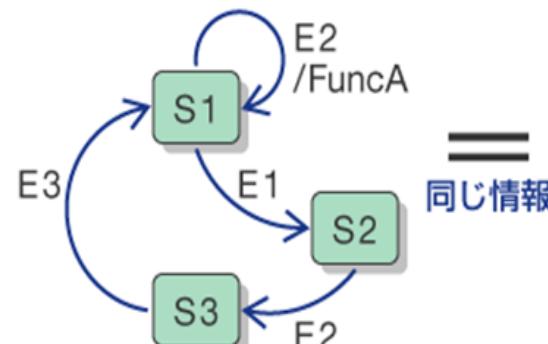
⇒組み込みシステムの特徴は、ユーザによる操作やセンサ信号などの外部からの事象(イベント)に対して、ランプを点けたり、モータを回転させたりなどの処理(アクション／振る舞い)を行う**反応型(リアクティブ)**システムであることが多い。

このようなリアクティブなシステムにおいては、内部の状態と外部の事象によって制御をおこなうよう、**振る舞いをベースにしたモデリング**が適している。

状態遷移図

状態遷移表

■ 状態遷移表のメリット



	S1	S2	S3
E1	⇒S2		
E2	FuncA⇒S3		
E3			⇒S1

モレ・ヌケ発見＝バグ発見！



- フィーチャモデルにより、システムの共通性(Commonality)と可変性(Variability)が識別できる。
- 「共通性」は個々の要求に対してアーキテクチャが提供する共通機能
- 「可変性」は要求によるカスタマイズやアーキテクチャの変更が必要な機能



■ 状態遷移設計との連携

⇒ フィーチャは、振る舞いの論理的単位。

振る舞いのモデル化には、状態遷移表が適している。

フィーチャと状態遷移モデルの対応をとり、モデルを資産として再利用。

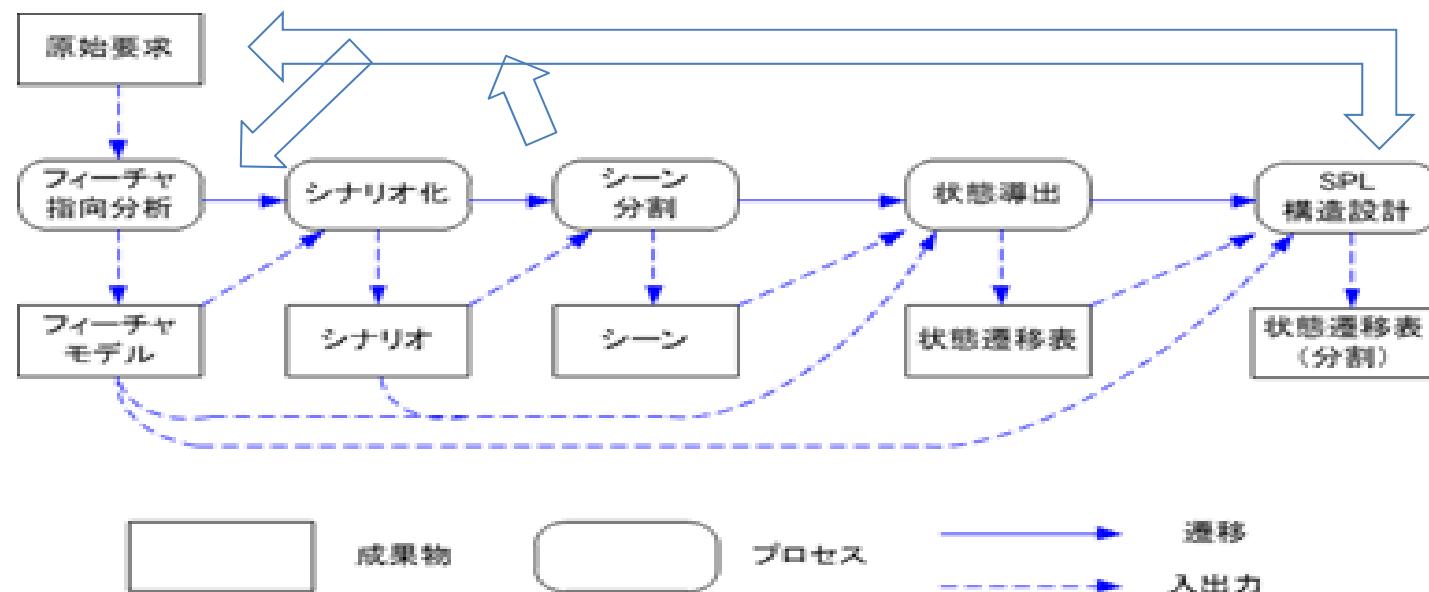
⇒ 可変性フィーチャのカスタマイズなど、メンテナンス性が向上する。



本プロセスの概要



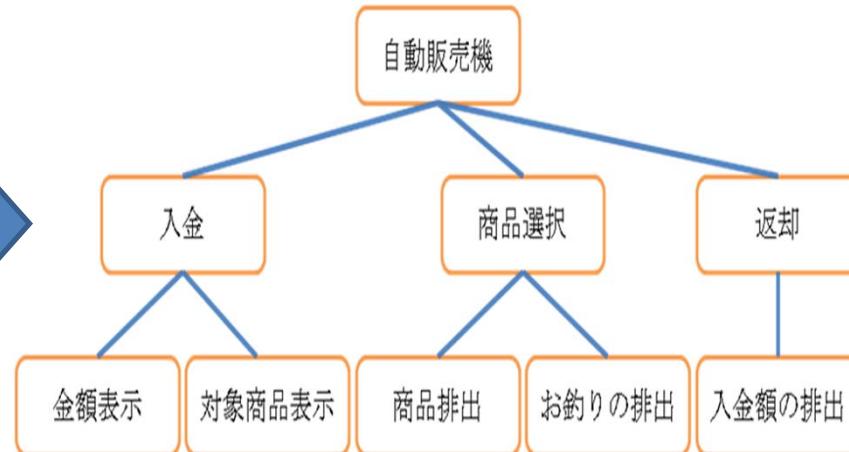
- 要求からのフィーチャの抽出
- フィーチャのブレークダウンと展開
- シナリオの作成とシーンの抽出
- 状態定義と状態遷移表の作成、階層化、最適化
- リファクタリング





■ 要求からのフィーチャモデルの作成

- ・入金額が価格以上の場合、対象商品の選択を有効とする。
- ・対象商品を選択し、購入できる。
- ・購入をキャンセルできる。
- ・入金額がわかる。
- ・お釣りや、未購入の投入金額を排出できる。



■ ユースケースのシナリオ化

⇒ユーザの目的ベースにシナリオを作成。

シナリオ1：商品の購入

- ・入金する。
- ・入金額が表示される。
- ・入金額が価格以上になると、対象商品の選択が有効となる。
- ・商品を選択する。
- ・商品が排出される。
- ・お釣りがある場合には、お釣りが排出される。

シナリオ2：商品の購入キャンセル

- ・入金する。
- ・入金額が表示される。
- ・入金額が対象商品の価格未満の場合、商品を選択しても、商品は排出されない。
- ・返却ボタンを押す。
- ・入金額が排出される。



- シナリオからシーン(条件)を抽出
- シーンの分析、状態とイベントの仕分け
- 状態遷移表の作成

トリガ条件:外部からのイベント

入金する、商品を選択する、返却する

継続条件:内部のイベント(データの変化)

価格以上(商品選択可能状態)

価格未満(商品選択不可状態)

シナリオ1

- ・入金する(入力)。
- ・入金額が表示される(出力)。
- ・入金額が価格以上になる(入力)と、対象商品の選択が有効となる(出力)。
- ・商品を選択する(入力)。
- ・商品が排出される(出力)。
- ・お釣りがある場合には、お釣りが排出される(出力)。

シナリオ2

- ・入金する(入力)。
- ・入金額が表示される(出力)。
- ・入金額が対象商品の価格未満の場合(入力)、商品を選択(入力)しても、商品は排出されない(出力)。
- ・返却ボタンを押す(入力)。
- ・入金額が排出される(出力)。

- ・継続条件=状態
- ・トリガ条件=イベント

	①商品選択不可 (価格未満)	②商品選択可能 (価格以上)
入金	入金額の表示。 ⇒価格以上で②へ	入金額の表示。 商品の選択を有効とする。
商品の選択	-	商品の排出。 お釣りがある場合はお釣りの排出。 ⇒①へ
返却	入金額の排出。	入金額の排出。 ⇒①へ



フィーチャモデルの階層化

■SPELの問題点

フィーチャモデルとアーキテクチャが遊離してしまう。

- ・ユーザへの提供サービス(ユースケース)ベースのフィーチャモデルでは、
非機能要件は独立したフィーチャとして抽出しづらい。
ですから、サービスベースのフィーチャをそのまま実装設計していくと、
機能間に跨る、干渉が発生する。





サービスフィーチャからその実現手段をフィーチャ分析する際には、
フィーチャの視点をユーザからシステムに切り替えて細分化することが
重要です。

また、その際には、非機能要件をアスペクトとして切りだしたり、
利用するPFやFWの提供機能を独立したフィーチャとして
アスペクトで定義します。



■フィーチャモデルとアーキテクチャの対応関係の問題について、
**フィーチャモデルの階層化と、ハード構成を含むアーキテクチャ
構造へのマッピング**によって解決することとした。



■ フィーチャモデルを、以下の4階層で定義する。

①特性層 … サービスレイヤ

- ・ユースケースから抽出した実現したい状態や状況

②機能特性層 … 実現レイヤ

- ・特性層で抽出したフィーチャを実現するために必要な
フィーチャとして抽出

- ・非機能特性に関してもこのレイヤでフィーチャ化を考慮

③ハードコンポーネント層 … 環境レイヤ

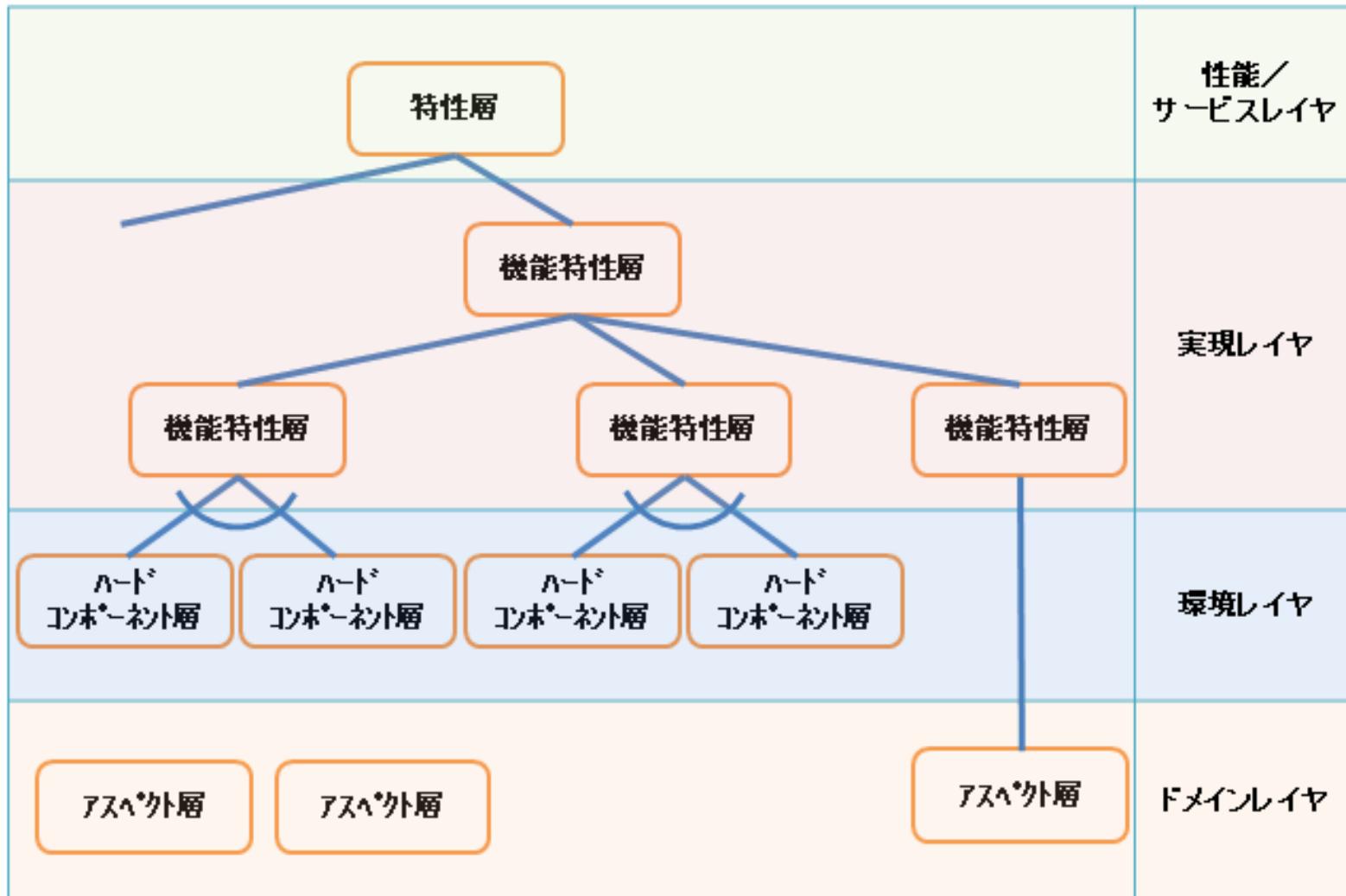
- ・フィーチャの最下位層と1:1に対応させた、ハードウェアコンポーネント

④アスペクト層 … ドメインレイヤ

- ・共通の機能要件、ドメインの標準技術などをフィーチャとして抽出



4階層構造のフィーチャモデルの概念図





状態遷移表の階層化

■ 状態遷移表の肥大化の原因と対策

【原因】

- ・状態や事象の定義粒度がばらばら。
- ・市販の表計算ソフトで、セル内の条件記述、分割が困難。
- このため、新たな状態を定義する傾向。



結果、一面の状態遷移表に並べて展開することで、レビュー、メンテナンスが困難な状態。

【対策】

- ・状態遷移表の階層化により、無効セルを削減し、メンテナンス性を向上させる。



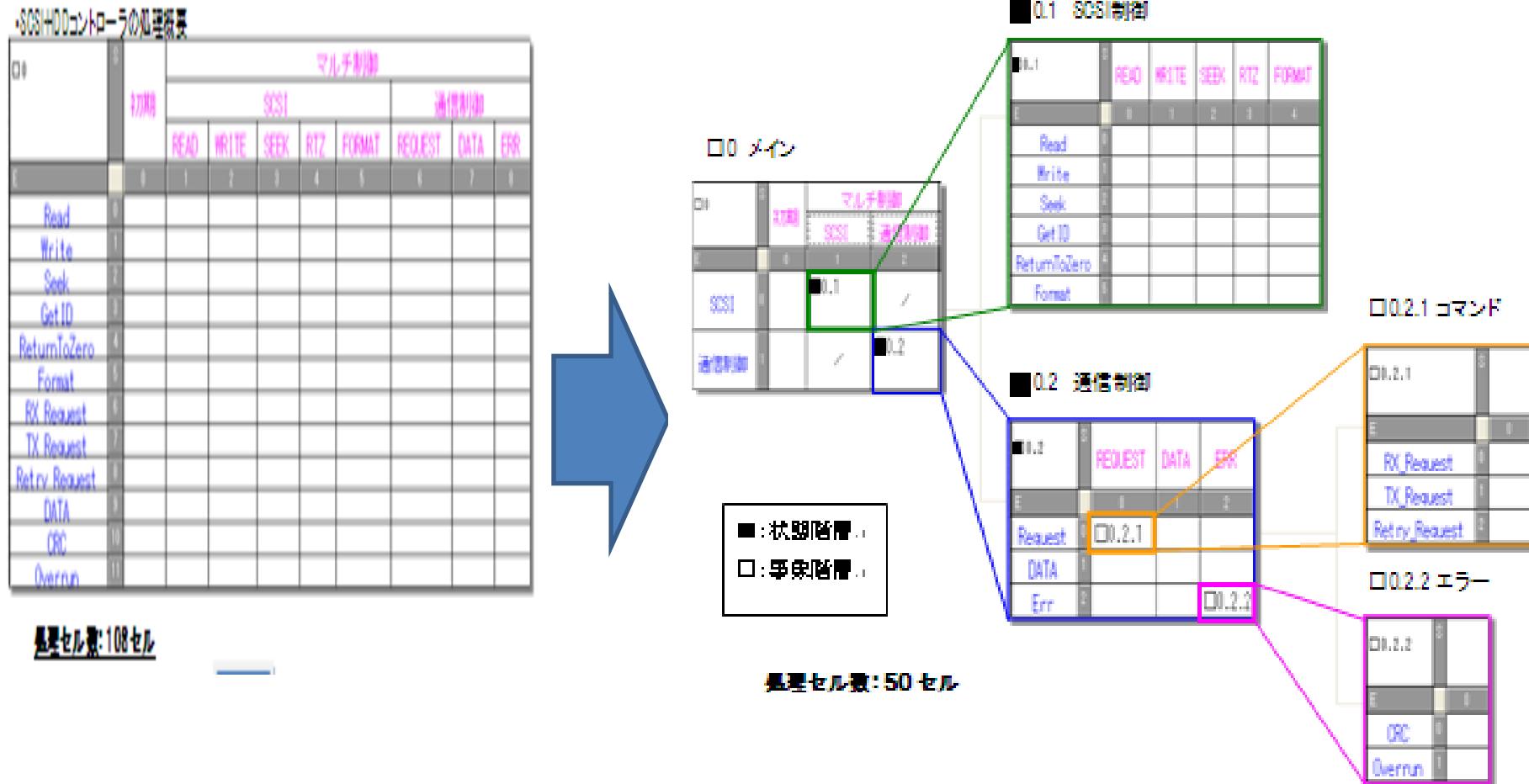


図 1.11 状態遷移表の階層化例と効果



■ 干渉とは？

特定のフィーチャの有無や振る舞いの変更が、他のフィーチャに影響を与えるという事。

ユースケースに基づく、フィーチャ分析モデルでは、組込みシステムに特有の性能、安全性、例外処理、環境などの非機能要件が複雑に絡み、フィーチャに横断的な関連が発生する。

機能間の干渉(フィーチャ間のインタラクション、相互作用)が発生し、再利用が困難な状態となることがある。

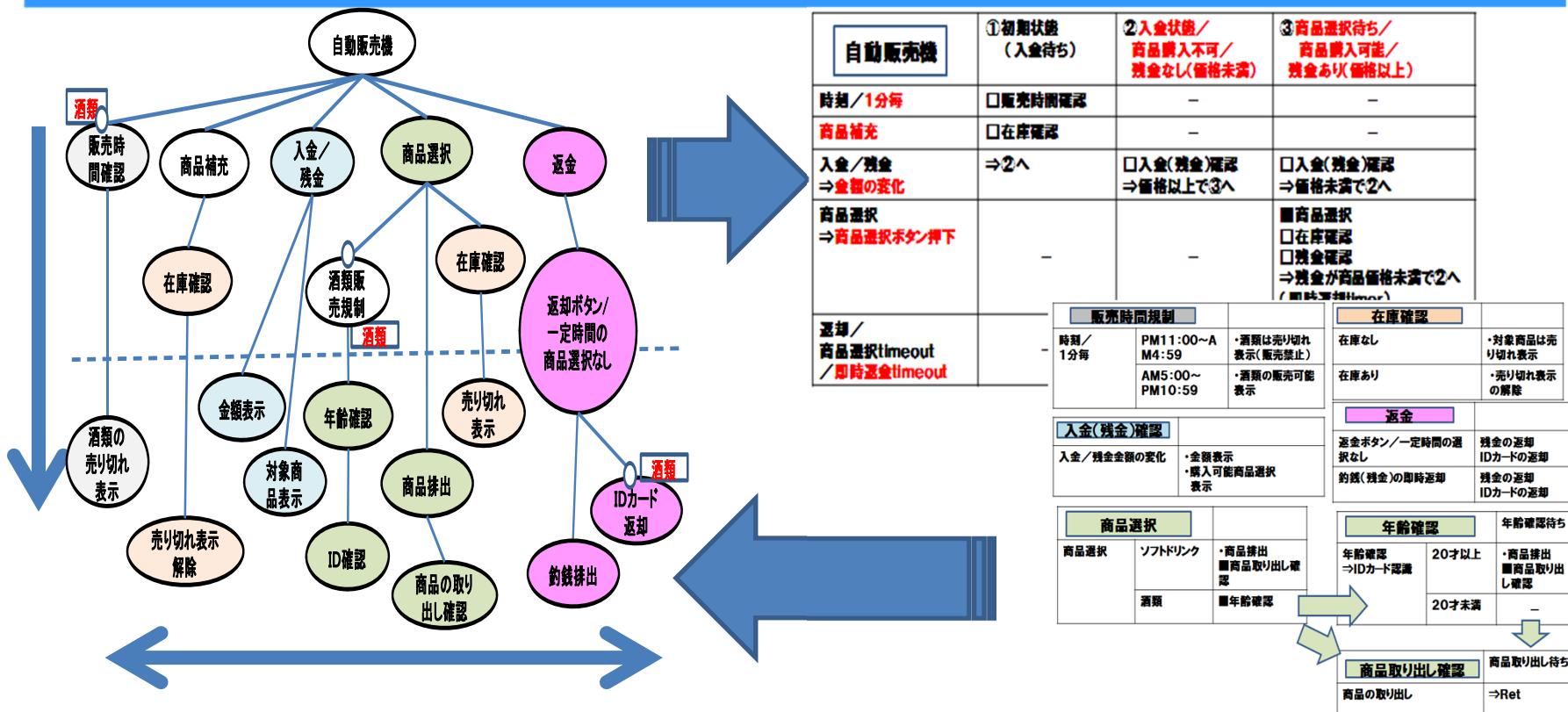


■リファクタリング

状態遷移表設計での要求や仕様の漏れ抜けをフィーチャモデルにフィードバックをかけて、実装構造とフィーチャモデルの対応関係を保つ、リファクタリングのプロセスが重要です。



■干渉の低減策



フィーチャのブレークダウン(縦方向の展開)は、アーキテクチャ/実装構造との対応をとり、フィーチャの拡散(横方向の展開)は、フィーチャに独立性を持たせます。

干渉を低減させるフィーチャモデルは、このフィーチャの展開と状態遷移表最適化の対応関係をリファクタリングにより、フィードバックしながら維持していくことで実現します。

自動販売機での実践例



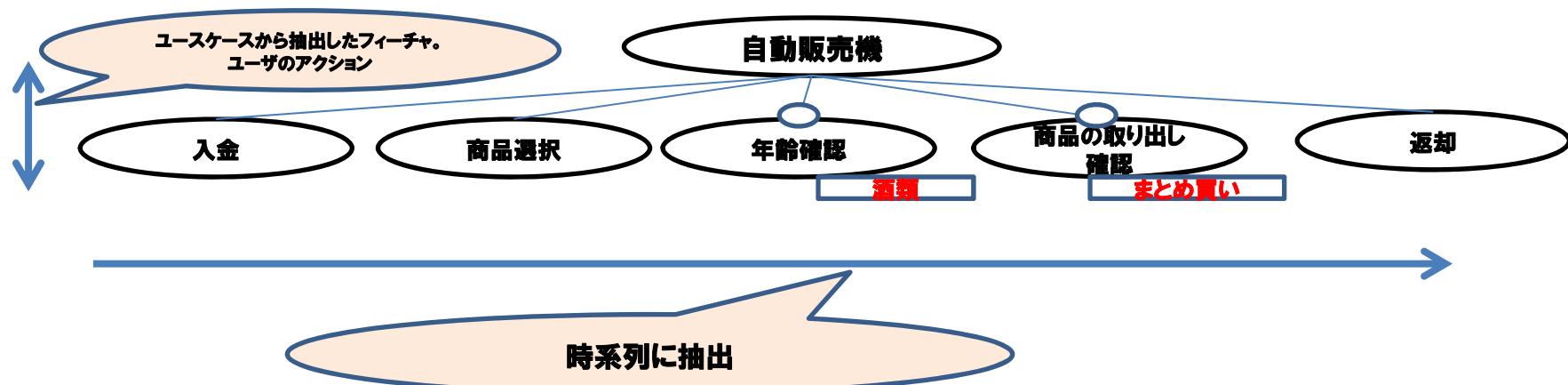
要求分析

缶製品の自動販売機の原始要求

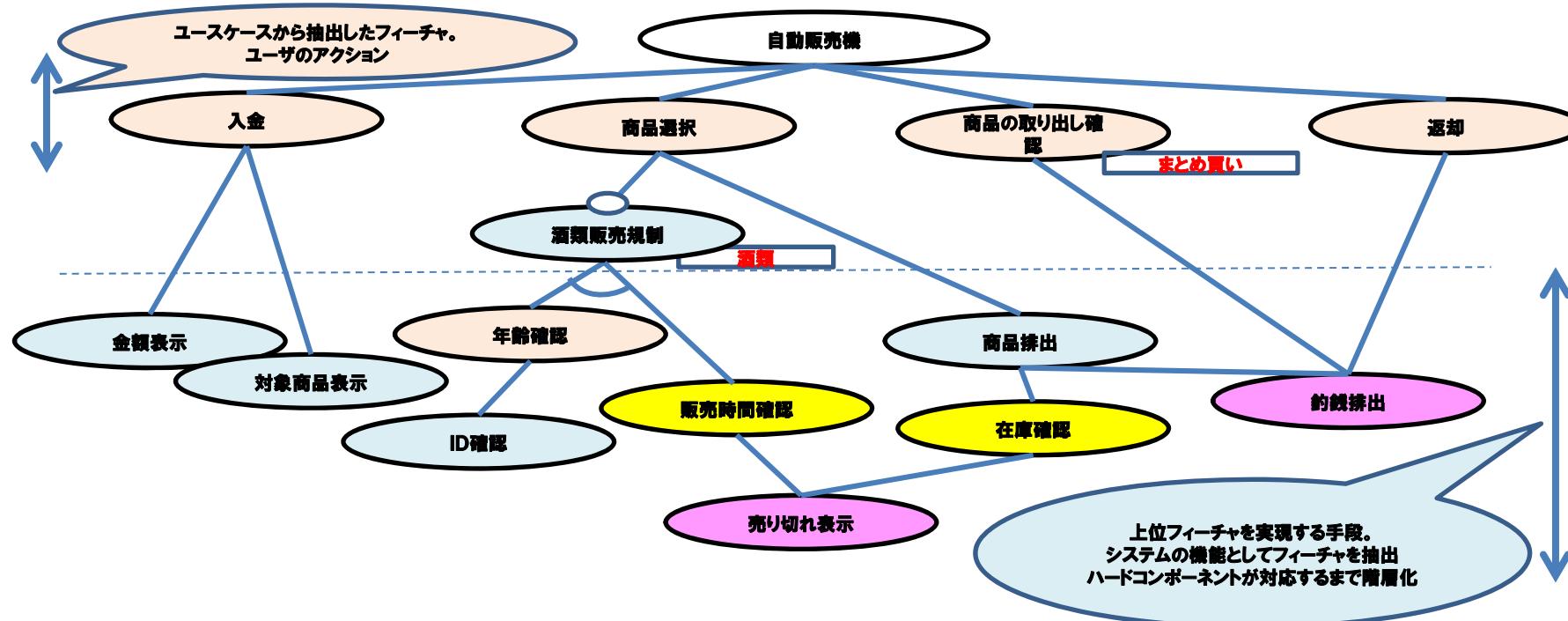
1. ジュースなどのソフトドリンク用販売機と酒類の販売機がある。
2. ソフトドリンクと酒類を混在させることができる。
3. 酒類の販売には以下の規制がある。
 - ・日本向けには販売時間の制約(pm 11時～am 5時は販売禁止)がある
 - ・欧州向けには、年齢確認(20才以上)が必要
 - ・規制のない地域もある
4. まとめ買いが可能である。ただし、商品の取り出しが終わったことを確認したのち商品を選択させる。
5. 在庫の確認により、売り切れ表示／商品の選択を不可とする。

以上の要求から、ソフトウェアプロダクトラインの系列製品としては、以下の2点とした。

- ・酒類の販売の有無
- ・まとめ買いの有無

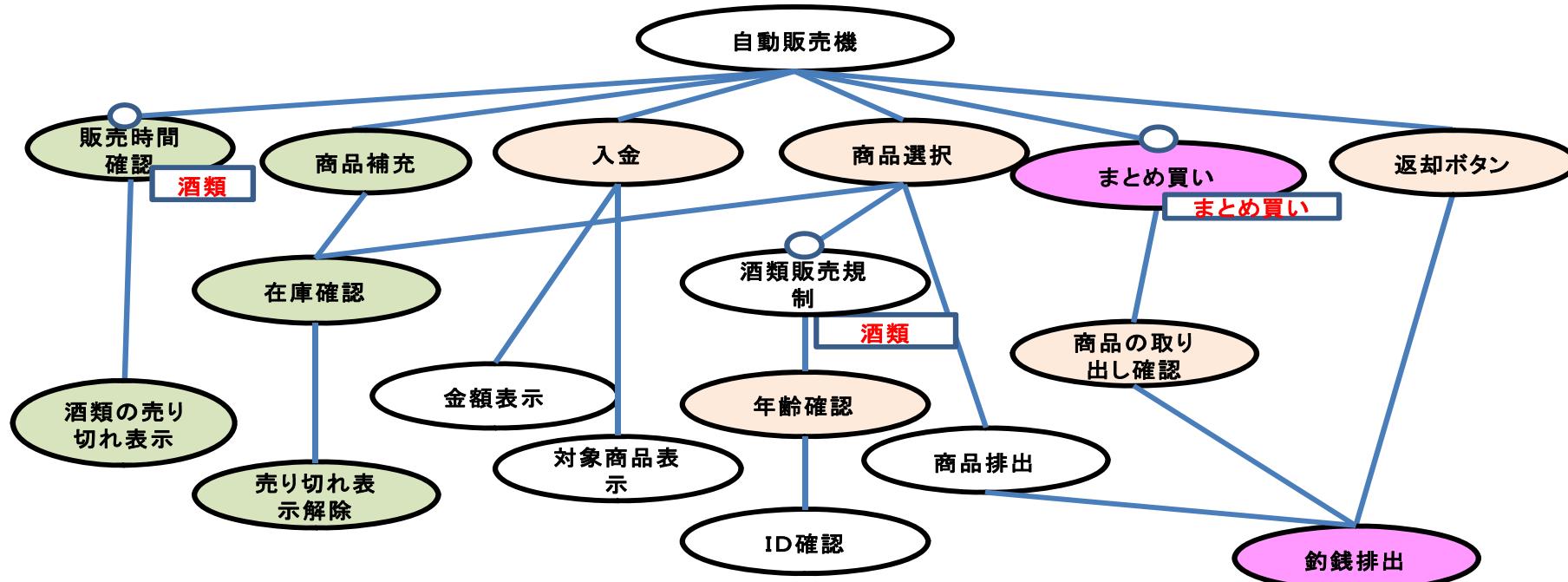


フィーチャ指向分析



フィーチャの分析とフィーチャの再構成

- ・ 酒類の販売規制表示を売り切れ表示で代用する。⇒売り切れ表示が干渉する
- ・ まとめ買いの有無により、釣銭排出のフィーチャが干渉する。
- ・ 酒類の販売時間による規制は、商品選択にかかわらず必要なので、販売時間の確認はサービスレベルのフィーチャとする。
- ・ 在庫確認も、自動販売機の起動時にも必要なので、サービスレベルのフィーチャとする。
また、売り切れ商品の補充時に、**売り切れ商品を解除する機能**も必要。
- ・ 自動販売機のアクターとして、商品の購入ユーザだけではなく、**商品の補充・交換／メンテナンス要員**も存在する。
⇒要件／ユースケースの検討漏れ。

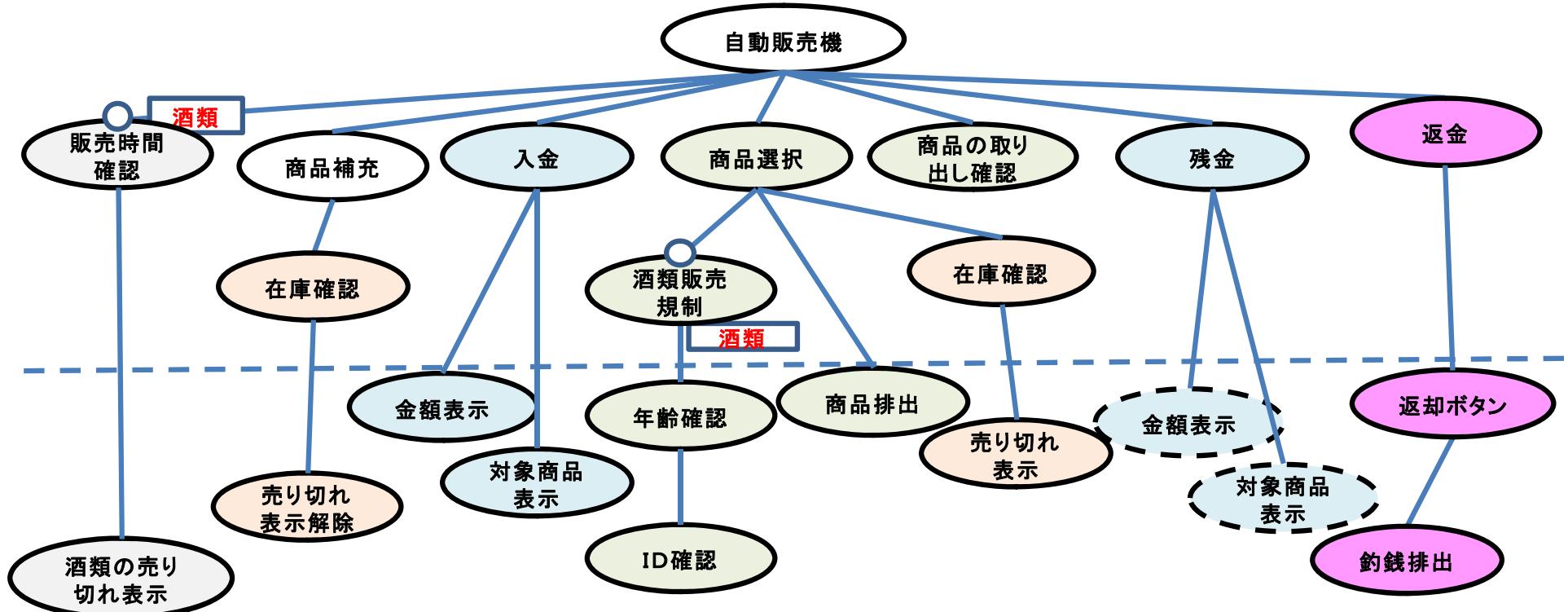


- 釣銭排出フィーチャの振る舞いの整理。
 - 返却ボタン押下で、釣銭返却
 - 一定時間の商品選択なしで、釣銭返却
 - まとめ買い時、残金が商品価格未満で、釣銭返却。
 - まとめ買ainし時、商品排出で、即時釣銭返却

まとめ買いの定義

まとめ買いは、商品購入後の残金が商品価格以上の場合に、連続して商品を購入するユーザの行為。
つまり、システムが持つ状態ではない。まとめ買いの有無はオプションでもない。

まとめ買いの場合は、残金が商品価格未満、返却ボタンの押下、一定時間の商品選択なしで釣銭返却。



【干渉の検討】

- ・酒類販売時間規制での売り切れ表示と商品購入後の在庫確認の売り切れ表示、商品補充後の売り切れ表示解除
- ・入金時と残金時の金額表示／対象商品購入可能表示
⇒入金は外部イベント、残金は内部イベント



シナリオ作成

■ ①ジュースの購入

1. 入金する。
2. 入金額が表示される
3. 購入可能商品の選択ランプが点灯する。
4. 商品を選択する。
5. 対象商品が排出される。
6. 商品を取り出しこよる。
7. 残金が商品額以上の場合、再度購入可能商品の選択ランプが点灯する。
8. 一定時間の商品選択なし、または返却ボタンの押下にて
釣銭が返却される

• ③購入のキャンセル

1. 入金する。
2. 入金額が表示される
3. 購入可能商品の選択ランプが点灯する。
4. 返却ボタンの押下にて釣銭が返却される

• ②酒類の購入

1. 入金する。
2. 入金額が表示される
3. 購入可能商品の選択ランプが点灯する。
4. 商品を選択する。
5. 対象商品が酒類の場合、年齢確認を要求する。
6. 20才以上の年齢が確認できたら、商品を排出する。
7. 商品を取り出しこよる。
8. 残金が商品額以上の場合、再度購入可能商品の選択ランプが点灯する。
9. 一定時間の商品選択なし、または返却ボタンの押下にて釣銭が返却される

• ④商品の補充

1. 商品を補充する。
2. 対象商品の売り切れ表示を解除する。



一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association

© Japan Embedded Systems Technology Association 2013

シーン抽出



- ①ジュースの購入
- 1. 入金する。
- 2. 入金額が表示される
- 3. **購入可能商品**の選択ランプが点灯する。
- 4. 商品を選択する。
- 5. 対象商品が排出される。
- 6. 商品を取り出しうる。
- 7. **残金が商品額以上**の場合、再度購入可能商品の選択ランプが点灯する。
- 8. **一定時間の商品選択なし**、または**返却ボタンの押下にて**釣銭が返却される

- ③購入のキャンセル
- 1. 入金する。
- 2. 入金額が表示される
- 3. **購入可能商品**の選択ランプが点灯する。
- 4. 返却ボタンの押下にて釣銭が返却される

- ②酒類の購入
- 0. 時刻による販売規制がある場合、**am5:00～pm10:59**であること。
- 1. 入金する。
- 2. 入金額が表示される
- 3. **購入可能商品**の選択ランプが点灯する。
- 4. 商品を選択する。
- 5. **対象商品が酒類**の場合、年齢確認を要求する。
- 6. **20才以上の年齢**が確認できたら、商品を排出する。
- 7. 商品を取り出しうる。
- 8. **残金が商品額以上**の場合、再度**購入可能商品**の選択ランプが点灯する。
- 9. **一定時間の商品選択なし**、または**返却ボタンの押下にて**釣銭が返却される

- ④商品の補充
- 1. 商品を補充する。
- 2. 対象商品の売り切れ表示を解除する。

トリガ条件

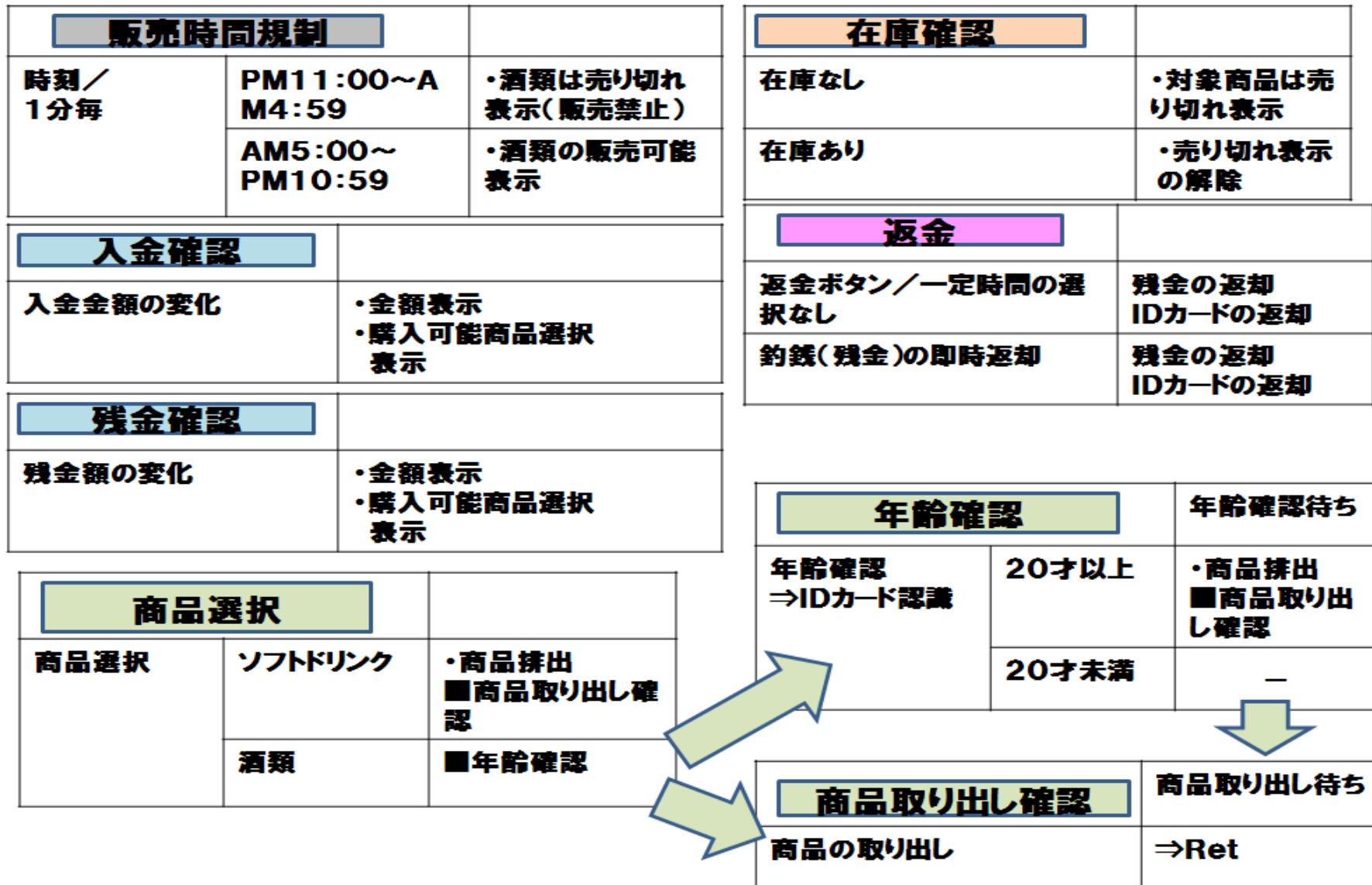
入金、商品選択(ソフトドリンク／酒類)、商品取り出し、返却ボタン／一定時間の選択なし、年齢確認(20歳以上／未満)
時刻確認(PM11時～AM4:59／AM5:00～PM10:59)
商品の補充

継続条件

入金(残金)が商品額以上／未満



状態遷移表作成





自動販売機	①初期状態 (入金待ち状態)	②残金が価格未満 (商品購入不可状態)	③残金が価格以上 (商品購入可能状態)	④返金待ち
時刻	<input type="checkbox"/> 販売時間確認	—	—	—
商品補充	<input type="checkbox"/> 在庫確認	—	—	—
入金	⇒②へ	<input type="checkbox"/> 入金確認 ⇒価格以上で③	<input type="checkbox"/> 入金確認	<input type="checkbox"/> 入金確認 ⇒価格未満で②へ
商品選択	—	—	<input checked="" type="checkbox"/> 商品選択 <input type="checkbox"/> 在庫確認 <input type="checkbox"/> 残金確認 ⇒残金が商品価格未満で即時返却timer、 ④へ	—
返却ボタン押下／ 商品選択timeout／ 即時返却timeout	—	<input type="checkbox"/> 返金 ⇒①へ	<input type="checkbox"/> 返金 ⇒①へ	<input type="checkbox"/> 返金 ⇒①へ

- ・入金確認と残金確認は、まとめることが可能。
- ・④返金待ち状態と、②残金が価格未満(商品購入不可状態)は同一状態。
- ・返却ボタン押下と商品選択timeoutは同一のイベントなので、まとめることが可能。

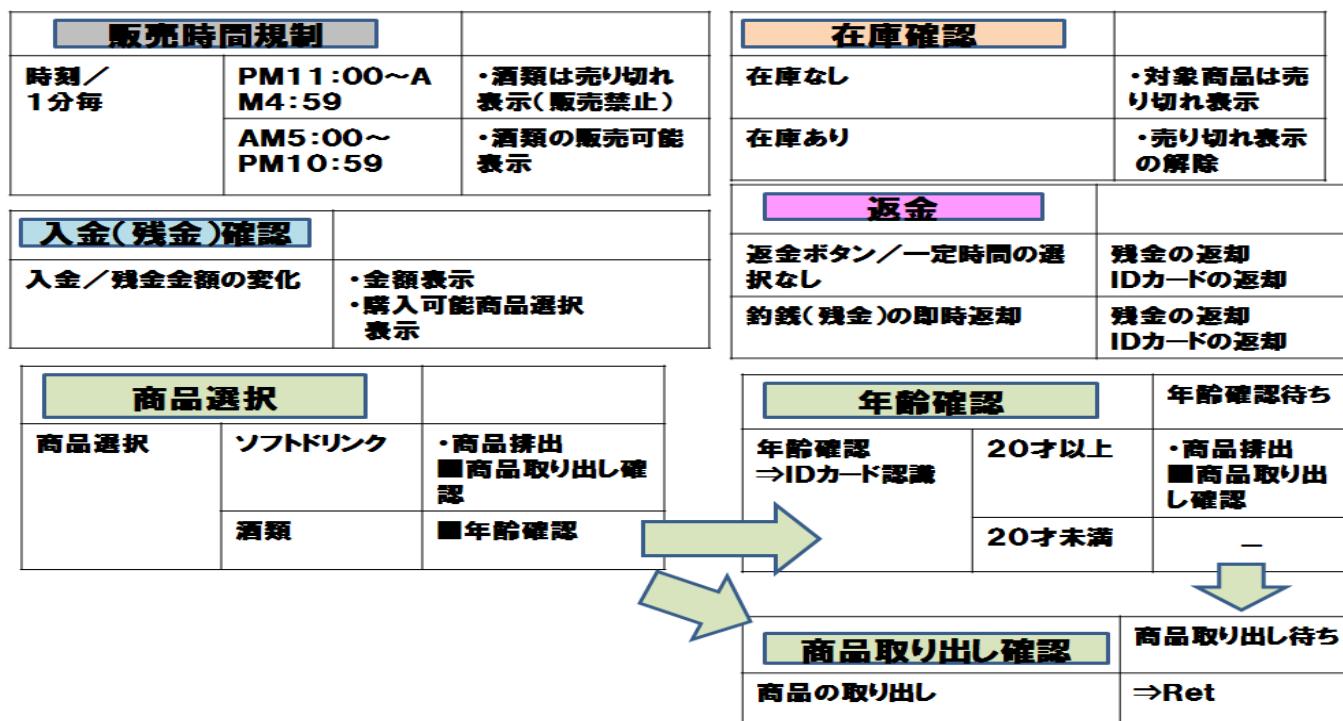
【制約条件】

- ①商品選択時、酒類の場合は毎回年齢確認することになるが、IDカードを挿入したまでの購入という条件にした。
- ②商品を取り出すまでは、入金も釣銭返却も受け付けないことになるが、商品の取り出し忘れを防止する付加機能として採用した。

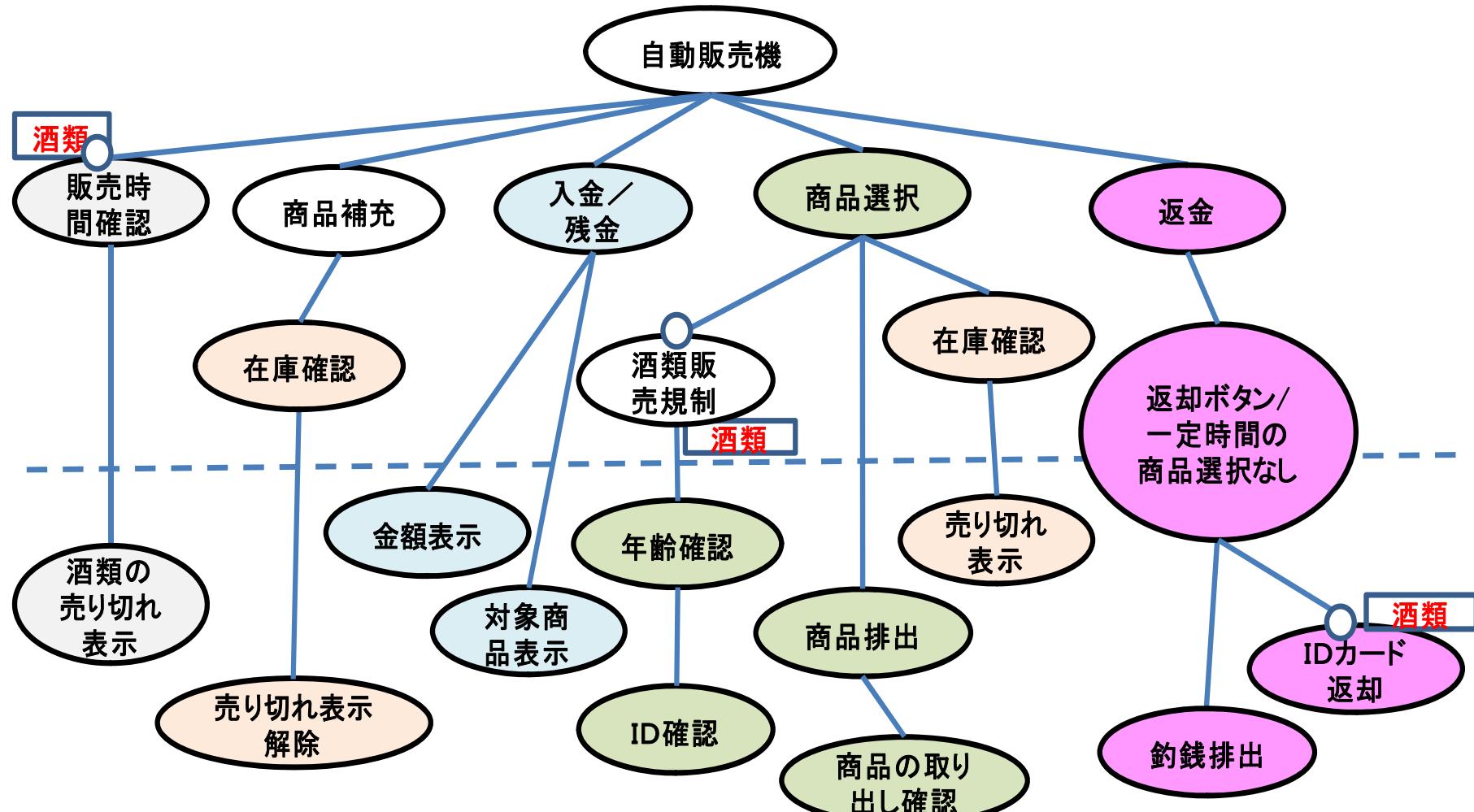
状態遷移表の最適化



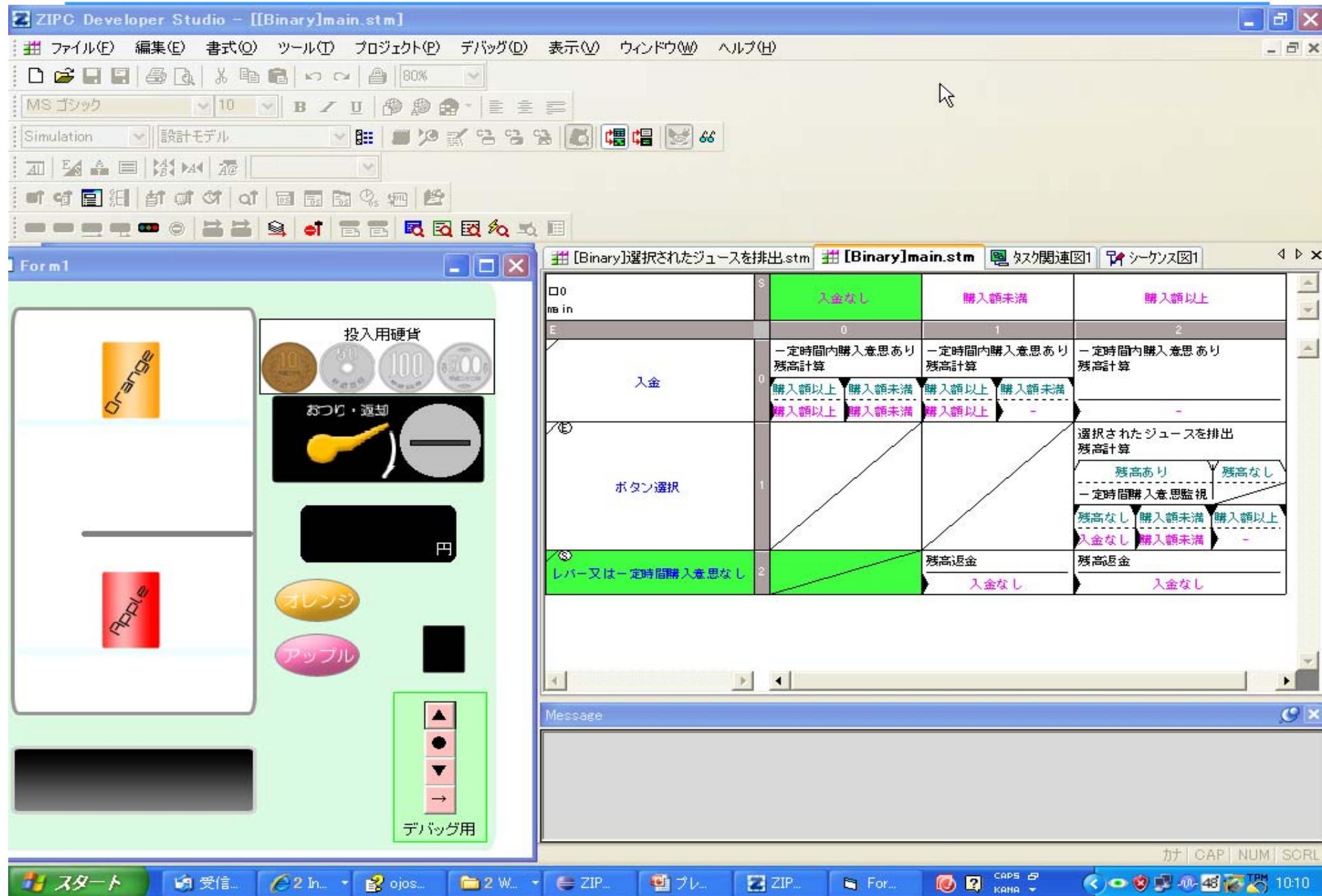
自動販売機	①初期状態 (入金待ち)	②入金状態/ 商品購入不可/ 残金なし(価格未満)	③商品選択待ち/ 商品購入可能/ 残金あり(価格以上)
時刻/1分毎	<input type="checkbox"/> 販売時間確認	—	—
商品補充	<input type="checkbox"/> 在庫確認	—	—
入金/残金 ⇒金額の変化	⇒②へ	<input type="checkbox"/> 入金(残金)確認 ⇒価格以上で③へ	<input type="checkbox"/> 入金(残金)確認 ⇒価格未満で②へ
商品選択 ⇒商品選択ボタン押下	—	—	<input checked="" type="checkbox"/> 商品選択 <input type="checkbox"/> 在庫確認 <input type="checkbox"/> 残金確認 ⇒残金が商品価格未満で②へ (即時返却timer)
返却/ 商品選択timeout /即時返金timeout	—	<input type="checkbox"/> 返金 ⇒①へ	<input type="checkbox"/> 返金 ⇒①へ



リファクタリング



自動販売機ZIPCシミュレーションデモ





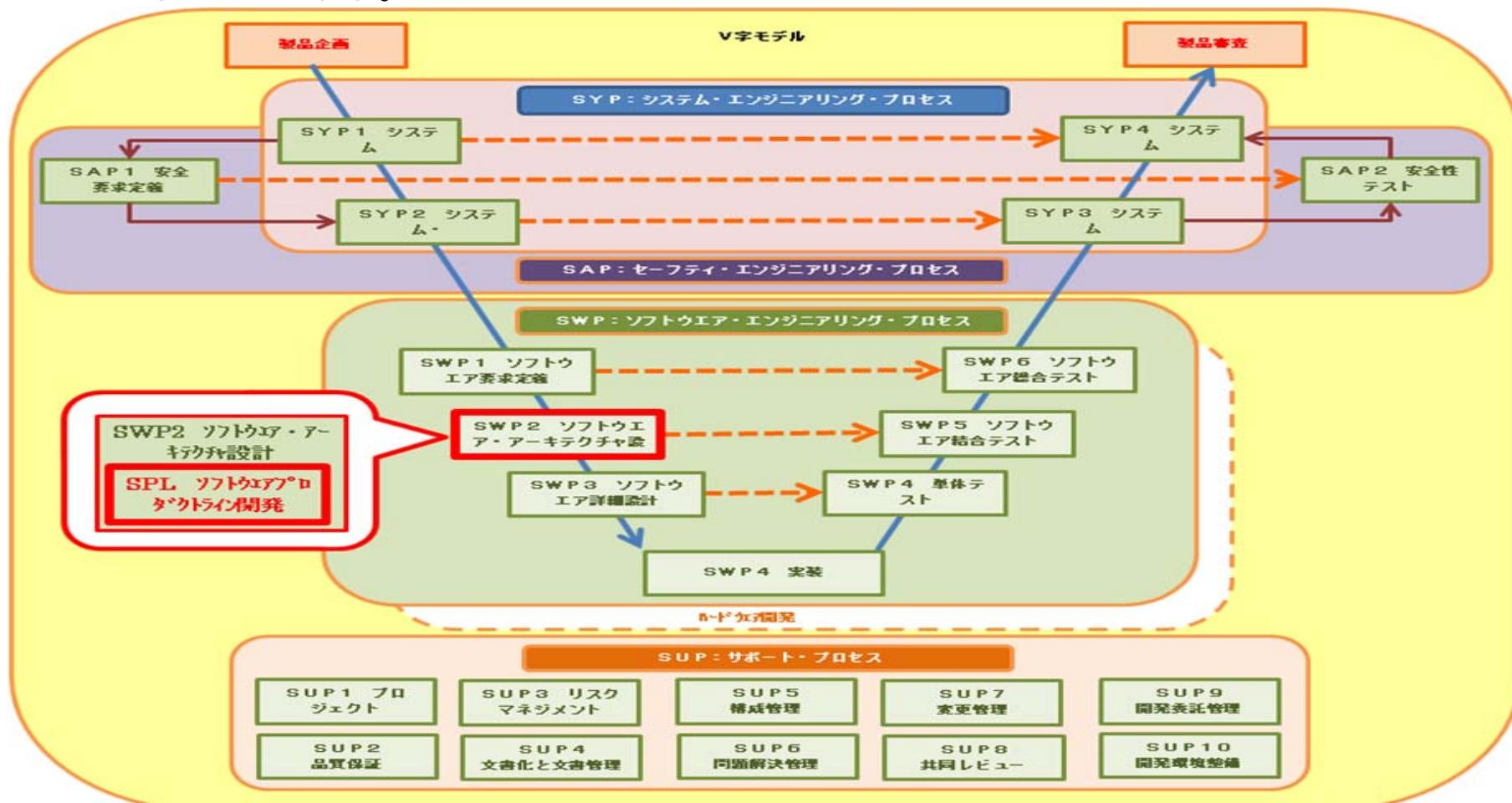
3章 SPLE開発プロセス [プロセス定義編]

3.1 本ガイドのプロセス定義範囲

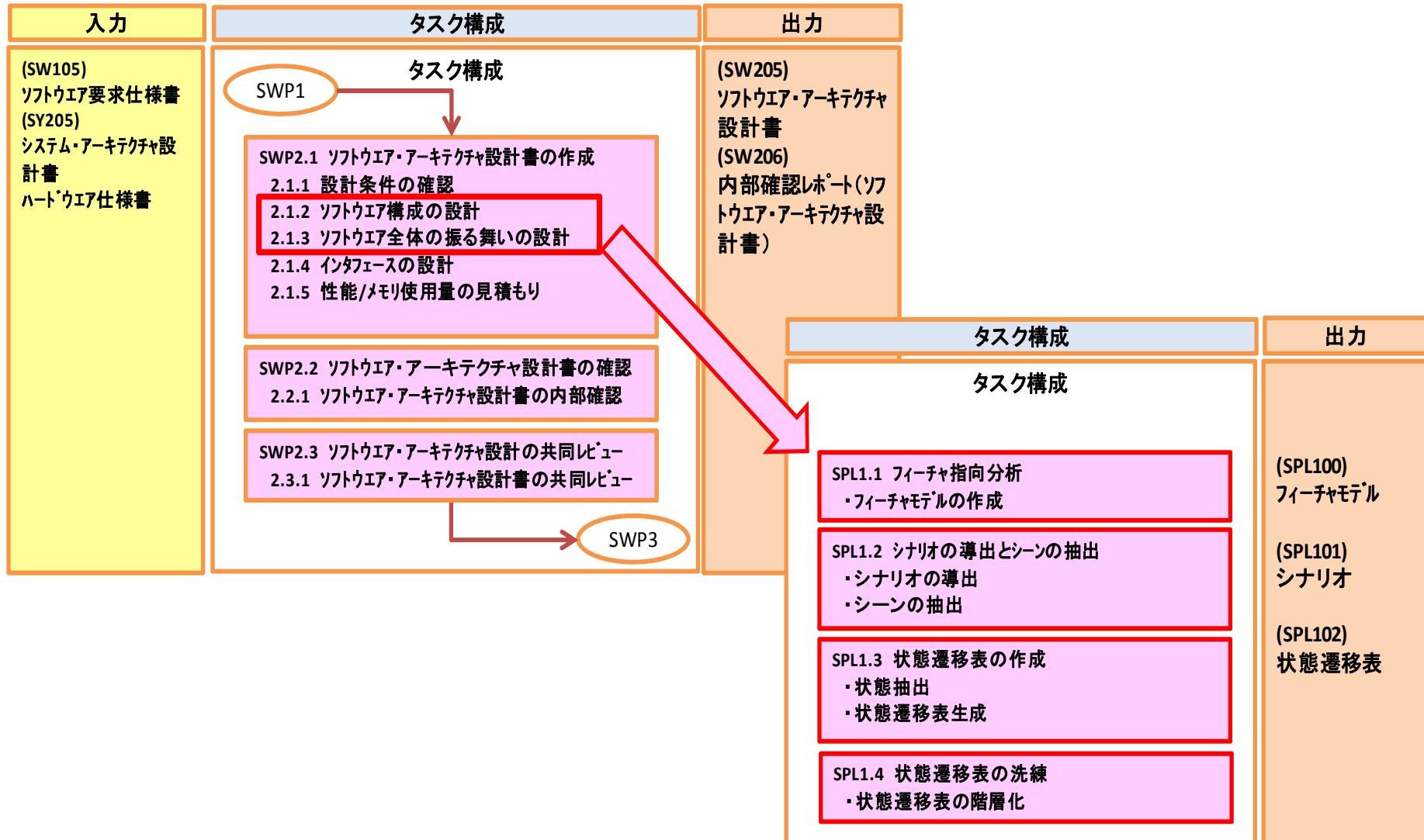
3.2 SWP2のタスク構成の変更点



- このSPL(E(ソフトウェア・プロダクトライン開発))のプロセスは、IPA/SECのESPR(組み込みソフトウェア向け開発プロセスガイド)ver. 2. 0に対して、SWP2「ソフトウェア・アーキテクチャ設計」のSWP2. 1. 2「ソフトウェア構成の設計」、SWP2. 1. 3「ソフトウェア全体の振る舞いの設計」を置き換えるものである。



3. 2 SWP2のタスク構成の変更点







3. 2013年度の新規取り組みテーマの紹介



■ソフト開発の課題と傾向

⇒システム障害やトラブルの対応にとまどう、システムの改修時に膨大な調査がかかる。こんな経験はないだろうか。これらは現行のシステムが“ブラックボックス化”しているのが大きな原因である。

そのうえ、度重なるプログラム変更や機能追加などでシステムのブラックボックス化は進むばかり。「もはや仕様書をそのつど現行システムに合わせて修正するのには限界がある」

システムから仕様をひも解く「リバース・エンジニアリング」より

<http://itpro.nikkeibp.co.jp/article/COLUMN/20070903/280974/>



■リバースモデリング

そこで今、にわかに注目を集めているのが「リバース・エンジニアリング」だ。

リバース・エンジニアリング(以下、リバース)とは、現行のシステム(ソースコードやデータベース)を解析し、その仕様を明らかにする技術。



「肥大化したシステムの分割や、プログラム間の依存関係の見直し、リファクタリングの検討などの分野で効果が大きい」。複雑化したシステムの仕様をリバースによって導き出し、保守性や拡張性などの向上につなげる。

■【仮説】

「フラグがある所に、状態はある。!!」



■ 2013年度のテーマ

「リバースモデリングにおける状態遷移表の適用」

⇒レガシーコードから、状態を抽出し状態遷移表モデルを導出する手法の研究。





2012年度 状態遷移設計研究会成果報告
「状態遷移表設計におけるS P L E実践ガイド」の紹介

2013/5/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



JASA技術本部成果発表会

「いつ使うの？今までしょ！」
いまから使える OpenEL[®] 1.0

2013年5月22日
プラットフォーム研究会
古俣 学



1. プラットフォーム研究会の紹介
2. OpenEL®とは
3. ロボット産業の動向
4. 平成24年度の活動計画
5. 平成24年度の活動結果
6. 考察・課題
7. 平成25年度の活動計画
8. 今後のPF研究会のロードマップ
9. OMGにおけるOpenEL®のロードマップ
10. まとめ

1. プラットフォーム研究会の紹介



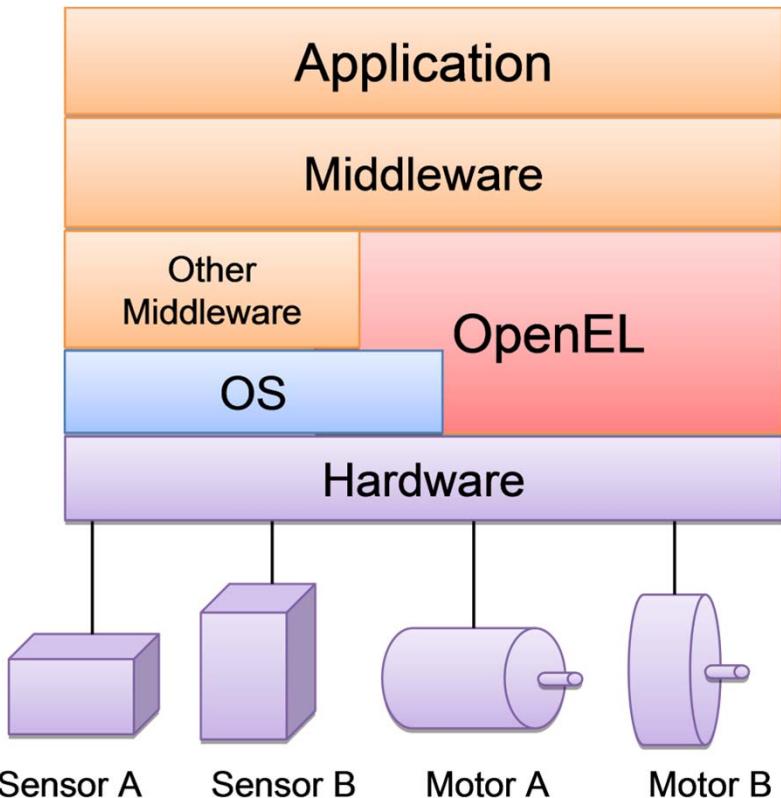
- 設立: 2000年
- 目的: プラットフォームの事例、特性等の評価、利活用の指針策定
- アドバイザー
 - 独立行政法人産業技術総合研究所 イノベーション推進本部 イノベーション推進企画部 総括企画
主幹 神徳徹雄
 - 首都大学東京 准教授 システムデザイン学部
博士(工学) 武居直行
 - 名古屋工業大学 教授 機能工学専攻 日本機械学会フェロー
博士(工学) 佐野明人
 - 福井工業大学 教授 機械工学科
博士(工学) 古荘純次
- 産学官連携コーディネーター
(地独) 東京都立産業技術研究センター
- 参加企業・団体
 - アップウインドテクノロジー・インコーポレイテッド
 - 図研エルミック株式会社
 - 株式会社コア
 - 株式会社エヌデーデー
 - 株式会社セントラル情報センター
 - (地独) 東京都立産業技術研究センター
 - 東芝システムテクノロジー株式会社
 - 株式会社ビツツ
 - 株式会社フィット・デザイン・システム
 - 株式会社イーシーエス
 - 日本システム開発株式会社
 - イーソル株式会社
 - 株式会社パトリオット
 - キヤノン株式会社
 - 大宮技研合同会社
 - オリエンタルモーター株式会社

2. OpenEL®とは(1/2)



■ OpenEL (Open Embedded Library)

- ・ ロボットや制御システムなどのハードウェアに近いソフトウェアの実装仕様を標準化
- ・ ソフトウェアの移植性、再利用性、生産性を向上
- ・ 組込みシステム向けのオープンなプラットフォーム



2. OpenEL®とは(2/2)



■ 誕生の背景

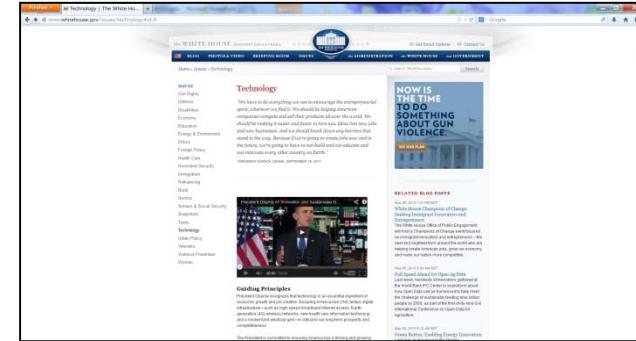
- ・ プラットフォーム研究会では拡大が予想されるロボット市場に着目
- ・ ロボット用ソフトウェアのプラットフォームは？
- ・ ロボット用ソフトウェアにはRTミドルウェアやROS等がある
- ・ しかし、ハードウェアに近いレイヤーは標準化されていないため苦労することが多い
- ・ ハードウェアに近いレイヤーを標準化するためにOpenELを作成

3. ロボット産業の動向



■ オバマ政権でロボット強化

- 中核技術の投資に加え、取り組みの一環として7000万ドルの資金を提供
- <http://www.whitehouse.gov/issues/technology#id-8>



■ 成長戦略を担うロボット産業

- 産業用ロボット市場は世界一
- 約1兆円規模の市場
- <http://president.jp/articles/-/8837>



■ 大学での技術開発

- 商業店舗でサービスロボットの試行
- <http://www.sankeibiz.jp/compliance/news/130403/cpd1304030501005-n3.htm>



4. 平成24年度の活動計画



■ 仕様策定

- OpenEL version 1.0

■ 知的財産権の確保

- 「OpenEL」の商標登録
- ロゴマークのデザイン
- ドメイン名の取得 - openel(.com/.net/.org/.or.jp/.jp等)

■ 普及・啓発活動

- ET Westや各種セミナーやイベントでの講演
- 雑誌等への執筆
- 他団体との連携

5. 平成24年度の活動結果(1/10)



■ 開催実績(会議11回、合宿1回)

- 5月24日 第1回会議
 - 仕様策定、勉強会(講師: キヤツツ: 穴田啓樹氏)他
- 6月21日 第2回会議
 - 仕様策定、ロボット技術に関する調査項目の洗い出し他
- 7月26日 第3回会議
 - 仕様策定、勉強会(講師: コア: 堀部幸祐氏)他
- 8月30日 第4回会議
 - 仕様策定、ロボット技術に関する調査研究報告他
- 9月27日 第5回会議
 - 仕様策定、JASA/ETセミナーのアンケート結果の分析他
- 10月25日 第6回会議
 - 仕様策定、勉強会(講師: 都立産技研: 金田光範氏)他

5. 平成24年度の活動結果(2/10)



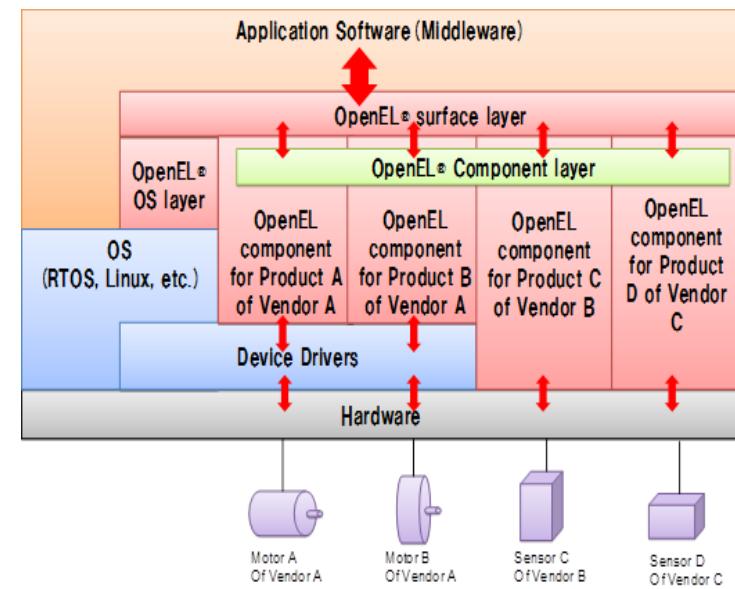
■ 開催実績

- 11月22日 第7回会議
 - 仕様策定、ET 2012のセミナー等の結果報告他
- 12月20日 第8回会議
 - 仕様策定、OMG技術会議の報告、平成25年度事業計画及び予算の検討他
- 1月24日 第9回会議
 - 仕様策定、勉強会(講師:フィットデザインシステム:辻氏)他
- 2月21日 第10回会議
 - 仕様策定、「OpenEL試作機 ステッピングドライバー」のデモ(オリエンタルモーター:板橋正也氏)他
- 3月28日 第11回会議
 - 仕様策定、OMG技術会議の報告、調査研究報告他
- 合宿の開催(4月26日～27日)
 - 平成23年度活動のまとめ
 - 平成24年度事業計画の詳細検討
 - 勉強会(RTミドルウェア、ROS、制御系装置におけるソフトウェア開発等)



■ 仕様策定

- OpenEL version1.0 を策定した
 - 新たにSurface(サーフェース)レイヤとComponent (コンポーネント)レイヤを追加
 - ✓ これによりハードウェアを抽象化を実現
 - サーフェイスレイヤとは
 - ✓ OpenEL利用者向けに提供するAPIを定義している
 - ✓ 定義されたAPIのみを使用することによって各デバイスハードウェアの差異を意識することなくソフトウェアの構築が可能
 - コンポーネントレイヤとは
 - ✓ 各デバイスハードウェア毎の処理になつており、サーフェイスレイヤのAPIから実際のデバイス操作への橋渡しを行う



5. 平成24年度の活動結果(4/10)



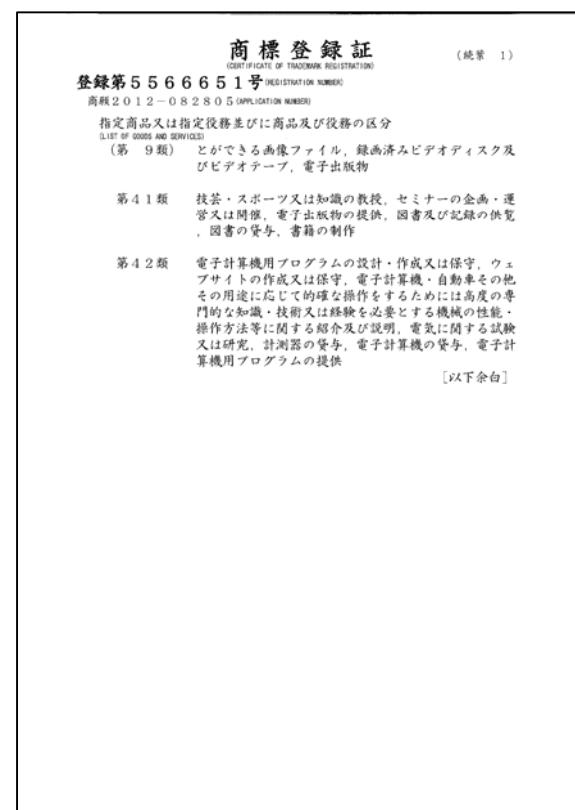
- モータAPIを追加
 - 速度制御
 - `elMotorSetSpeed_I32` / `elMotorSetSpeed_F64`
 - `elMotorGetSpeed_I32` / `elMotorGetSpeed_F64`
 - 位置制御
 - `elMotorSetPosition_I32` / `elMotorSetPosition_F64`
 - `elMotorGetPosition_I32` / `elMotorGetPosition_F64`
 - トルク制御
 - `elMotorSetTorque_I32` / `elMotorSetTorque_F64`
 - `elMotorGetTorque_I32` / `elMotorGetTorque_F64`
- OpenELのホームページに5月22日に公開
 - <http://www.jasa.or.jp/openel/>

5. 平成24年度の活動結果(5/10)



■ 知的財産権の確保

- OpenELを商標登録した
 - 商標登録日は2013年3月15日 第5566651号



5. 平成24年度の活動結果(6/10)



■ 知的財産権の確保

- ・ ロゴマークをデザインした



5. 平成24年度の活動結果(7/10)



■ 知的財産権の確保

- ドメイン名の取得
 - ドメイン名の取得はおこなわなかった
 - そのかわりOpenELのホームページを作成した
- <http://www.jasa.or.jp/openel/>

Main Page/ja

OpenEL (Open Embedded Library) is an open platform to standardize the specifications of the software implementation of robotics and control systems.

Main Page

OpenEL is an open platform to standardize the specifications of the software implementation of robotics and control systems.

Current Problems in Embedded System Development

Current problems in the development of embedded systems have been considerable effort required. For example, running on the LED or just to operate the motor on different hardware, there may spend many days. Because an application program interface to control the output of the sensors and motors, were each uniquely defined by the device manufacturer, has been implemented since.

Therefore, JASA propose to unify these interfaces which were different for each device manufacturer. JASA focus on robotics and control systems, has started drafting specifications OpenEL for Robots. In OpenEL for Robots, by the basic portion of the software platform for robotics and control systems, and aims to enable applications running on different hardware too soon. This increases the portability and reusability of the software, resulting in improved quality, lower costs and lead to improved productivity is expected to improve convenience for users and developers.

Why OpenEL?

There are lots of problems in ROS and RT Middleware. For example, Limitation of Software Development Environment.

5. 平成24年度の活動結果(8/10)



■ 普及・啓発活動(講演6回、メディア掲載3回)

- ET West (6/15)で講演を行った
 - 「拡大するサービスロボット市場を狙え！～OpenELで簡単ロボット開発～」
- Interface 2012年10月号(8/25発売)に寄稿した
 - OpenELの紹介
 - LEGO MINDSTORM NXTへのOpenELの適用例
 - FM3基板を使用したライントレースロボットへのOpenELの適用例
- ルネサスエンジニアフォーラム2012(9/8)で講演・展示を行った



5. 平成24年度の活動結果(9/10)



- JASA/日経BP共同開催ETセミナー(9/13)を開催した
- 日経エレクトロニクス10月15日号で紹介された
- CEATEC(10/5)でOpenELの講演を行った
 - 日本再生の切り札はロボットで～ロボット産業創出狙う仕掛け人が登場
- ET2012のJASA技術本部セミナー(11/14)でOpenELの講演を行った
- ET2012でロボットトラック(11/16)を開催した
- 財団法人日本数学検定協会のインタビューを受けた
 - <http://www.su-gaku.biz/interview/2012-12-17-nakamura/index.html>

5. 平成24年度の活動結果(10/10)



■ OpenELの国際標準化

- 世界的な標準化団体であるOMG(Object Management Group)の技術会議に参加し、世界各国に向けてOpenELの紹介および標準化の提案を行った

OMGのRobotics Information Dayにて発表とデモ



- 2012年12月11日(火) 8:30–18:00
- 開催場所: 米国バーリングーム市
- 参加者: 36名前後
 - 日本 - 産総研、東京大学、筑波大学、電気通信大学
 - 東芝、本田技術研究所、チェンジビジョン
 - JARA(国際電気通信基礎技術研究所)
 - JASA(アップウインドテクノロジー)
 - 韓国 - 韓国電子通信研究院(ETRI)、セジョン(世宗)大学
 - Future Robot
 - 米国 - サウスウェスト研究所(ROS-Industrial)他
 - ドイツ - Honda Research Institute Europe GmbH
 - フランス他
- 産総研による参加報告
 - http://www.openrtm.org/openrtm/ja/news/ja_robots_info_day_2012_report





- 2013年3月19日(火)
- 参加者: 10名
 - ・ ウクライナ、ドイツ、日本
- JASAからはOpenEL[®]1.0の仕様策定作業の進捗について発表
- JASAからの「ハードウェア抽象化レイヤーWGの設立」提案は満場一致で可決
- JASAがOMGに正式に入会後、5月に情報提供依頼書(RFI)を発行し、新規にWGメンバーを募集
- WGの議長には、PF研究会委員長の中村が就任予定



6. 考察・課題



- モータやセンサの制御における複雑な要求(制御方法、単位系、変数の型、OSの有無等)が明らかとなった
- OpenEL version 1.0仕様の策定作業はかなり難航したが、関係者の協力により仕様を策定することが出来た
- 記事の執筆や日経BP社とのJASA/ETセミナーの共催により、OpenELの普及・啓発活動に拍車がかかった
- OMGへの入会(ドメイン会員)およびHAL WGの設立により、ワールドワイドでOpenELの国際標準化を進める環境が整った
- OpenEL version 1.0仕様の検証(verification)と妥当性確認(validation)のために、実証実験を実施する必要がある
- OpenELの実用化を推進するためには、普及・啓発活動に加えて、RTミドルウェアやROSとの連携が必要である

7. 平成25年度の活動計画



■ 仕様策定

- 各種産業用ロボット、サービスロボットを対象としたOpenEL version 1.0の実証実験の実施
- OpenEL version 2.0の策定
- OpenELとRTミドルウェアおよびROSの連携
- OpenELの国際標準化活動

■ 普及・啓発活動

- 週刊BCN 2013年04月15日号 vol.1477
「RT標準化を推進して組込み市場を活性化」
 - http://biz.bcnranking.jp/article/interview/face/1304/130418_133129.html
- 国際ロボット・カンファレンス 2013を日経BP社と共催
- ET2013ロボットトラック

国際ロボット・カンファレンス 2013



- JASA(セミナーWGおよびPF研究会)、日経BP共催
- 日時: 6/25(火) 10:00-17:00
- 場所: ソラシティ カンファレンスセンター(御茶ノ水) ホール
- テーマ: 行動支援ロボット、高付加価値ロボット
- 講演者(予定)

- 米iRobot、米Double Robotics
- 韓Future Robot、カナダ大使館
- トヨタ自動車(株)組立生技部
- トヨタ自動車(株)パートナーロボット部
- デンソーウェーブ、産総研
- 首都大学東京、名古屋工業大学
- 大阪工業大学、富士重工業
- サイバーダイン、オリンパス
- JASAプラットフォーム研究会:
「日本発の国際標準を狙う次世代のロボット開発プラットフォームOpenEL」



8. 今後のPF研究会のロードマップ



- 2014年3月まで
 - ・ 各種産業用ロボット、サービスロボットを対象としてVersion 1.0の実証実験を実施し、Version 2.0に反映
 - ・ RTミドルウェアやROSとの連携
 - ・ 日本国内での普及促進
- 2015年3月まで
 - ・ 各種産業用ロボット、サービスロボットを対象としてVersion 2.0の実証実験を実施し、Version 3.0に反映
 - ・ 世界各国での普及
- 2016年3月まで
 - ・ OMG(Object Management Group)での標準化(最短で2015年6月)
- 2017年3月まで
 - ・ ISOでの標準化WGの設立

9. OMGにおけるOpenEL®のロードマップ(1/2)



2013(H25)年 3月 : ハードウェア抽象化レイヤー WG(OpenEL® WG)を設立

2013(H25)年 6月 : 情報提供要請(RFI)を提出

2013(H25)年 9月 : レスポンスへの対応

2013(H25)年12月 : レスポンスのFIX

 JASAおよびHAL WGメンバーでドラフトを作成

2014(H26)年 3月 : 標準提案公募(RFP)ドラフト版提出

 JASAからOpenEL® Version 2.0仕様をOMGに提出

2014(H26)年 6月 : 標準提案公募(RFP)発行

2014(H26)年 9月 : 提案登録(LOI)締切

2014(H26)年12月 : 一次提案締切

9. OMGにおけるOpenEL®のロードマップ(2/2)



2015(H27)年 3月 : 二次提案締切

2015(H27)年 6月 : 標準提案の採択(OMG内部公開)

2015(H27)年 9月 : FTF(Finalization Task Force)による公文書作成

2015(H27)年12月 : FTFによる公文書作成

2016(H28)年 3月 : FTFによる公文書作成

2016(H28)年 6月 : 標準仕様の発行(一般公開)

ISO標準化WGを設立

ISO標準化WGへ標準仕様を提出



- ロボット技術の研究・開発に世界各国で**多額の予算**が投入
- 産業用ロボット市場において日本のシェアは世界一であるが、サービスロボット市場は産業用ロボットとは全く異なる新しい市場
- 「**サービスロボット大国日本**」の実現のためには、ロボット向けソフトウェアの国際規格や技術認証における**日本のリーダーシップ**が必須
- JASAでは、産総研、都立産技研、首都大学東京等と連携し、次世代のロボット開発プラットフォームである**OpenEL**を策定し、**国際標準化**を推進



ご清聴ありがとうございました。

さあ、いきますぐOpenELを使いましょう！

<http://www.jasa.or.jp/openel/>

「いつ使うの？いまでしょ！」
いまから使える OpenEL® 1.0

2013/5/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会（以下、JASA）が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



JASA ハードウェア委員会 活動報告

平成25年5月22日
JASA技術本部 成果発表会



活動趣旨

1. 現状認識

最適な組込みハードがあってこそ、高機能な組込みソフトが真価を発揮できる

2. 研究会活動

“これから組込みシステム開発に求められるハード技術とは何か”を考える

- ・技術者に求められるハードウェア技術要素とは？
- ・現在の技術者に不足している技術要素とは？

3. ゴール

ハードウェア・クライシスを招かないための提案をしたい

- ・ハードウェア技術要素マップとしてとりまとめる



主な活動内容

1. 「組込みシステム」の技術調査

現在、5つのテーマについてグループごとに調査中

WG1:組込みシステムの時代による変化(発展性)

WG2:組込みシステム構成におけるハードウェアの位置付けの変化

WG3:組込みシステム技術者に重要さを増す従来ハードウェア・スキルとは

WG4:組込みシステム技術者に求められる新しいハードウェア・スキルとは

WG5:ポスト組込みシステムとは何か

2. 企業視察

上記1の技術調査を目的に関東近隣企業の企業視察を行う

3. 研修会

委員会内外の有識者の講演による研修会を開催する

4. 日本プラグフェスト運営支援

5. 外部団体(JPCA、学校等)との交流



主な活動内容

研修会 第31回 JASA/ETセミナー 災害用ロボットがもたらす日本の未来とその可能性



●「国際救助隊サンダーバード構想」
京都大学 工学研究科 教授
工学博士 松野文俊 氏

●「災害用ロボットの活躍と今後の課題」
千葉工業大学 未来ロボット技術
研究センター 副所長
工学博士 小柳 栄次 氏



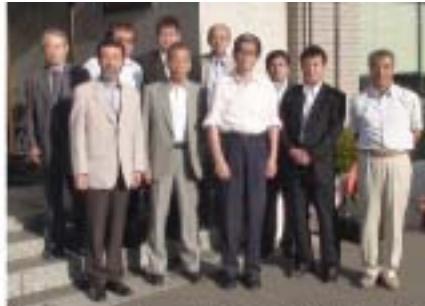
主な活動内容

企業視察

株式会社東和電機製作所(函館市)



東和電機製作所本社（函館）



JASAハートウェア研究会の調査チームと
浜出社長（前列中央）



浜出丸



工場見学で LED 照明器具を説明する浜出社長

Bulletin JASA 2010 Vol.36

SPECIAL: Embedded Technology 2010 開幕
ET2010 JASA会員出展企業紹介

SPECIAL: JASA国際だより ベトナム視察ミッション参加報告

技術寄稿:組込み製品責任者に捧げるソフトウェア開発管理のコツ

技術寄稿: いまさら聞けない 組込み入門講座(その5)「組込みソフトウェアの基礎の基礎」

会員企業訪問:全自動イカ釣り機を実現、世界シェア70%を誇るトップメーカー
株式会社東和電機製作所



中間報告

組込みシステムにおいて重要さを増す 従来のハードウェア技術とは

2013年5月22日
ハードウェア委員会 WG3
小澤拓治



- ・本レポートの目的
- ・環境の変化と組込み応用商品の変化
- ・製品開発スタイルの変化と組込み技術の変化
- ・マイコン基板にもアナログ回路が必要
- ・まとめ



本レポートの目的

ハードウェアの高速化, 高性能化, 高密度化によって, 組込み開発に必要な技術も変化している. 専用LSIチップ・セットの出現により, 液晶テレビ開発に必要な技術は, 回路設計からチップやパネルなどの組み合わせ技術に置き換えられている. このような流れのなかで従来のハードウェア技術は不要なものになってしまうのか. 製品開発に必須な要素として残る技術はあるのか. これらについて検証したい.

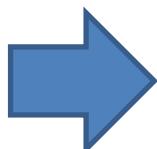


環境の変化と組込み応用商品の変化

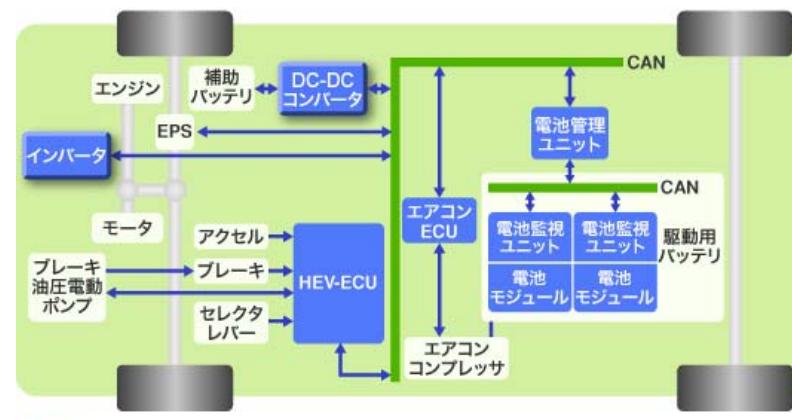
－省エネ要求やゆたかさ追求による商品の変化

クルマの世界では、EVやハイブリッドの普及により組込み技術は付帯装備から機能そのものを支える技術になっている

従来：ナビなどの電装品＝付帯装備



現在：EVのモータ制御など＝クルマの機能そのもの



*1

*2



環境の変化と組込み応用商品の変化

－省エネ要求やゆたかさ追求による商品の変化

組込みとは無縁だった家電製品にマイコンが搭載される

従来:マイコンとは無縁の家電製品



* 3



* 4



* 5



現在:マイコンによりインテリジェント化した家電製品



* 6



* 7



* 8

組込みシステムにおいて重要さを増す従来のハードウェア技術とは



環境の変化と組込み応用商品の変化

—省エネ要求やゆたかさ追求による商品の変化

現代の製品に共通の技術キーワード

=

マイコンとプログラムによる
モータやヒータなどのパワー・デバイス制御

パワー・エレクトロニクスはいまや組込みにとって
かかせない重要な技術



製品開発スタイルの変化と組込み技術の変化 – 機能を開発するのではなく、買ってきて機能を組み合せる時代

ディスクリートの回路設計からIP活用設計、そして今や
回路の入ったチップを購入する時代。

ディスクリートの回路設計



* 9

IP活用設計



* 10

専用チップ・セットを使った商品化



* 11



* 12



製品開発スタイルの変化と組込み技術の変化 —機能を開発するのではなく、買ってきた機能を組み合せる時代

液晶テレビを例にすれば…

RFフロントエンド + デジタルTV用SoC + 液晶パネル
これらの要素部品を買ってつなげればテレビはできる

しかし！

機能は実現できるがそれだけでは商品にならない！！

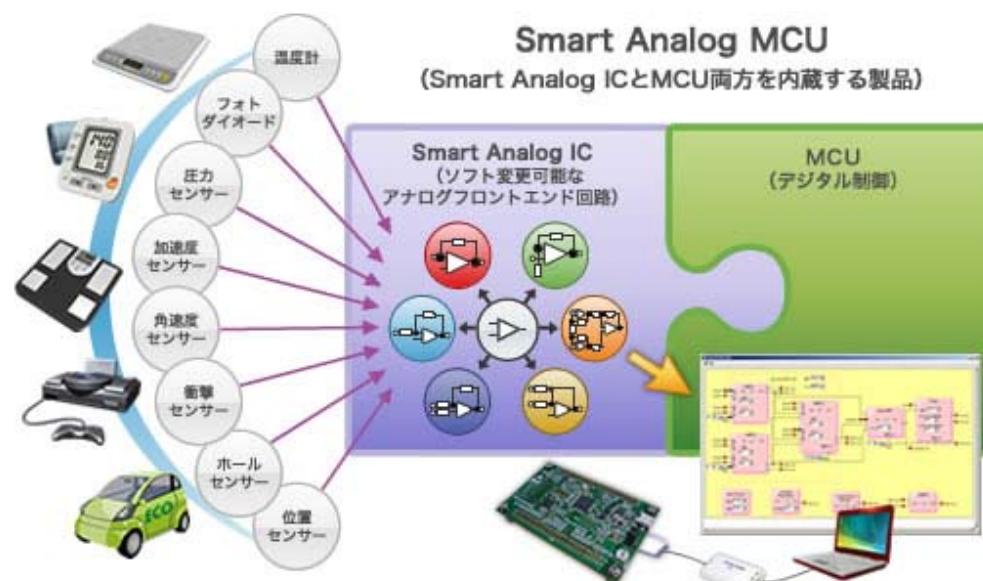
例えば

HDMIを正常に動作させるには部品をつなげるだけでは
だめ。基板のパターン設計など、高周波のアナログ技術が
必要



マイコン基板にもアナログ回路が必要 —A/DやD/A搭載マイコンでもアナログ回路が必要

各種センサとA/Dの間にはアンプやフィルタ回路が必要
フィルタやアンプはもっとも基本的なアナログ回路技術



* 13



まとめ

- ・ **環境の変化と組込み応用商品の変化**

今後の組込み技術の発展にはマイコン技術だけではなく、パワー・エレクトロニクス技術が必要不可欠

- ・ **製品開発スタイルの変化と組込み技術の変化**

部品を組み合わせることにより機能は実現できても商品は作れない。GHz帯の信号を扱う機器を安定して動作させるためには高周波回路技術が必要不可欠

- ・ **マイコン基板にもアナログ回路が必要**

A/DやD/A内蔵マイコンであっても、センサやスピーカなどを直接接続できるとは限らない。多くの場合においてフィルタや增幅回路などの基本的なアナログ回路が必要になる



結論

現代において、組込み技術応用分野は多岐にわたり、その開発手法も日々変化し続けている。

この変化によってマイコン関連技術とは別に、新たに脚光を浴びるようになった従来技術が存在する。

組込み技術が、従来の電子という枠ではなく、パワーエレやメカトロなどの従来技術と融合することで、新しいパラダイムの商品が生まれることを期待する。

終わり



参考:引用文献

- *1 <http://web.bayfm.jp/navi/>
- *2 http://www.semicon.toshiba.co.jp/application/automotive/ecology/power_train/evs/
- *3 <http://www.sony.co.jp/SonyInfo/CorporateInfo/History/SonyHistory/index.html>
- *4 <http://chihoko777.exblog.jp/9921439/>
- *5 <http://perker121118.jugem.jp/?eid=168>
- *6 <http://www.tiger.jp/products/ricecooker/>
- *7 <http://www.dyson.co.jp/Fans/BrowseTheRange.aspx?filters=desk&filters=tower&filters=pedestal>
- *8 http://www.mutoheng.com/~drafter/drafter/index_drafter.html
- *9 <http://www.dmprof.com/company/story/>
- *10 <http://www.chipworks.com/blog/technologyblog/2013/01/11/smart-tvs-at-ces-2013-operating-systems-and-processors/>
- *11 <http://www.chipworks.com/blog/technologyblog/2013/01/11/smart-tvs-at-ces-2013-operating-systems-and-processors/>
- *12 http://japan.renesas.com/products/smart_analog/index.jsp



「中間報告 組込みシステムにおいて重要さを増す従来のハードウェア技術とは」

2013/05/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



組込み技術者に求められる 新しいハードウェアスキルとは

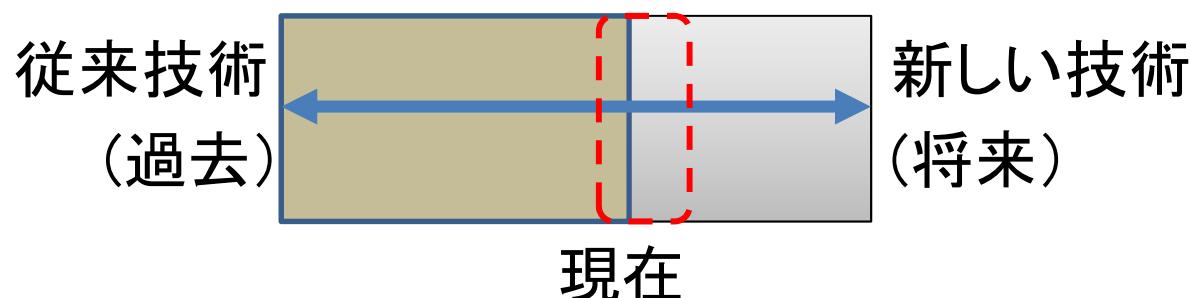
2013年5月22日
ハードウェア委員会 WG4
木下光明



■はじめに

組込み技術の範囲とは、エレクトロニクス全般に渡り広範囲となるため、必要なスキルの絞込みは難しいが、昨今の事象やこれからロードマップをたどり、組込み関連企業における技術者にとっての視点で、必要な要素から検討した。

新しいとする範囲を、下図のイメージとした。
(既存の技術や手法の最新、および発展部分)



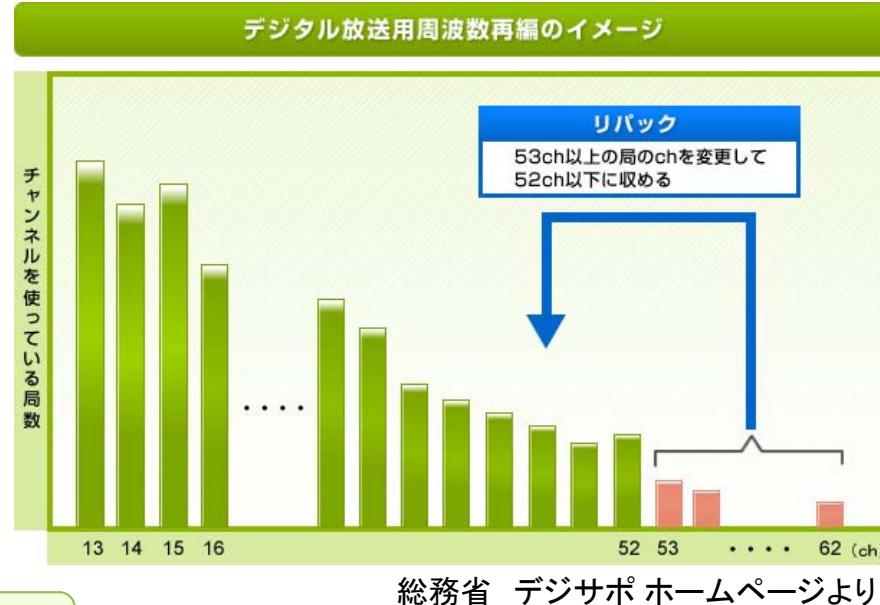


■ここ数年で起きた事象

地上デジタル化(2011年7月24日)

周波数再編(リパック)により
無線通信機の増加

- ・携帯電話
- ・災害
- ・移動体マルチメディア
- ・高度交通システム(ITS)



無線通信／非接触通信の普及

WiFiや赤外線通信
Felicaなどの非接触ICカード



WiMAX、DC-HSDPA、LTEがサービスイン

数十Mbpsの無線通信がどこでも利用可能に



■ここ数年で起きた事象

3D表示可能なモニタ、テレビの登場

奥行き感のある映像表現が可能に
グラスレスでも3Dに



(Nintendo 3DS)

AR(Augmented Reality:拡張現実)技術の発達

携帯電話やスマートフォンやゲームなどが
画像処理可能なまでにパフォーマンスが向上
現実環境における作業支援がその応用



IPV4アドレスの枯渇(2011年4月)

予想以上に早く枯渇
IPV6に徐々に移行 (アドレス32ビット→128ビット)

GPGPU (General-purpose computing on graphics processing unit)

グラフィック専用チップを演算用に転用



■ここ数年で起きた事象

ROM／RAM／FlashROM記憶装置の容量増

およそ1年で2倍の割合で増えてきている



ハードディスク等の補助記憶装置の容量増

TB(テラバイト)単位の製品も低価格化して普及
(3.5inch1台で4TB程度)

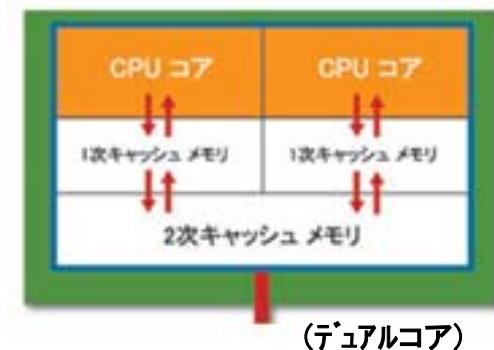


ARMプロセッサの台頭

汎用CPU、SoCの両領域に浸透
ローエンドからハイエンドまで多数のラインナップ

CPUのマルチコア化

動作クロック、單一コア当たりの処理能力が
上げどまり、コアの並列化で処理能力強化





■ ここ数年で起きた事象

いわゆる「ビッグデータ」の定量的価値(例)

50億台の携帯電話が使用(2010年)

300億のコンテンツが毎月Facebook上で共有

IT費用の5%増加で、年間40%増のデータ創出

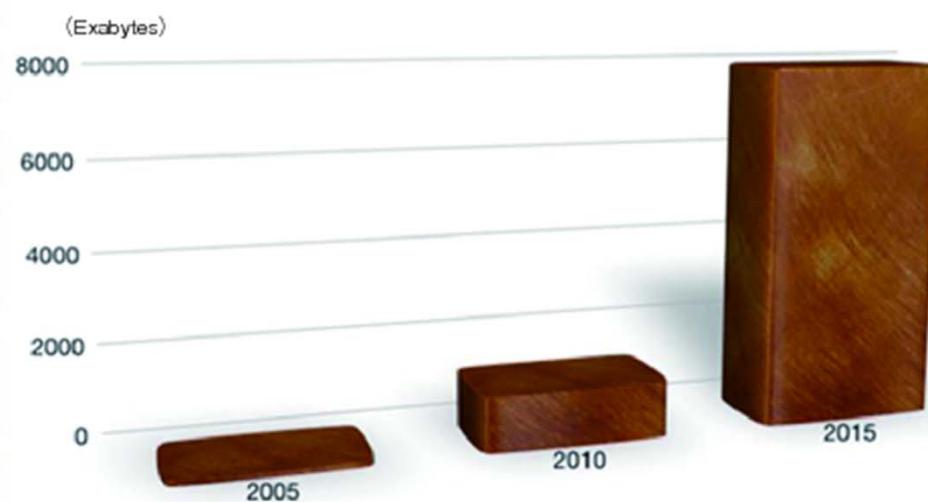
米国のヘルスケアでは年間3000億ドルの価値
創出が期待(スペインの年間ヘルスケアコストの2倍)

EUの公共セクターでは年間2500億ユーロの価
値創出が期待(ギリシアのGDPを超える)

個人の位置情報データを活用することで年間
6000億ドルの消費者価値創出が期待

小売の営業利益に60%改善の見込み

10年間のデジタルデータの成長



(出典)情報通信審議会ICT 基本戦略ボード資料
「2011 Digital Universe Study:Extracting Value from Chaos」
(IDC/2011.6)「, Big data:The next frontier for
innovation,competition, and
productivity」(McKinsey Global Institute/2011.5)により作成)

総務省 平成24年版 情報通信白書より

Kilo → Mega → Giga → Tera → Peta その後がExabyte = $1,000^6=10^{18}$ バイト

社会で扱う情報量は2025年には現在の約200倍になるとの試算もある。



■現在から近未来技術マップ

情報通信

半導体分野

SoC(ドリームチップ)、
LSTP(超低電圧デバイス)、高速不揮発メモリ

ネットワーク分野

無線 超高速無線アクセス、マルチホップ、
変調技術、高効率および広帯域アンプ

ユーザビリティ
分野

有線 波形制御、光増幅、訂正技術

知覚インターフェース、表現インターフェース
インタラクション技術、セキュリティ

ロボット分野

次世代産業用意ロボット
サービスロボット、フィールドロボット

MEMS分野

光MEMS、RF-MEMS、センサMEMS、バイオMEMS

設計製造加工
分野

バーチャルマニュファクチャリング
人・ロボット協調生産、ミニマル加工技術

二次電池分野

エネルギー密度指向型、出力密度指向型、
寿命指向型

エネルギー分野

太陽、風力、バイオマス等

システム・
新製造

エネルギー

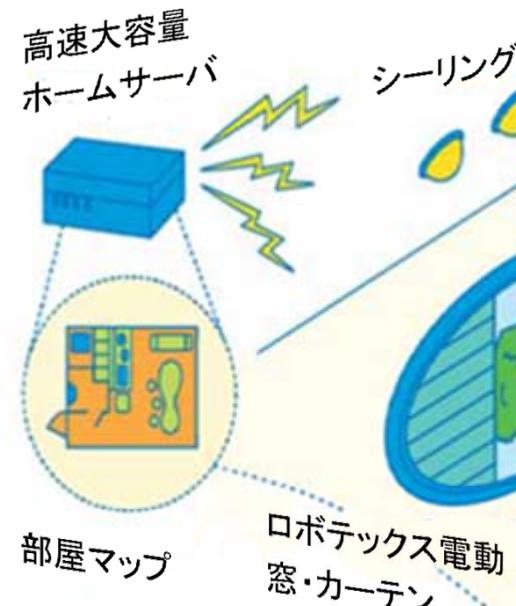
経産省 技術戦略マップより抜粋



■2025年想定される社会イメージ(1)

★ IT・ユビキタスで
暮らしが変わる！？

● 働く家具たち



マルチモニタ



見守りロボット

マニュピレータ付
ロボテックス車椅子

匂い管理用
空気洗浄器

・新しいアプリケーションでは
高度化して、複合的要素が伴う

NEDO技術戦略マップ
パンフレット抜粋

● お部屋が丸ごとロボット

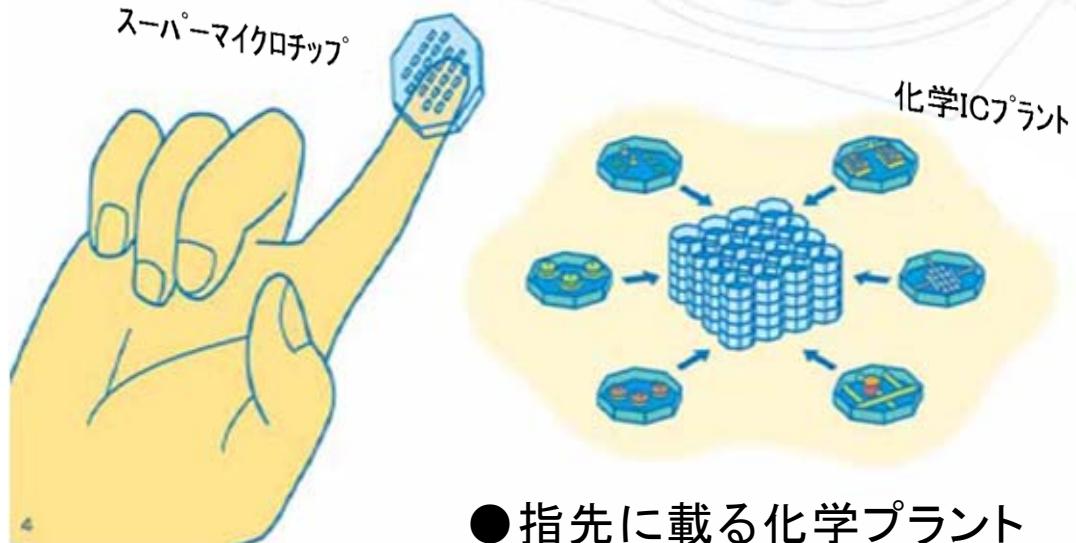


■2025年想定される社会イメージ(2)

★工場とモノづくり



●遠隔操作、マイクロサイズの物づくり



●指先に載る化学プラント

さらなる小型化、高速化は進む



■ ハードウェア開発サイクルの基本と展開

今までに行われてきた開発の流れを考察すると、

その手順は一連のサイクルから成り立っている。

新たに生まれた手法や技術により、そのサイクルが同じ

時間経過であっても、何倍もの機能や性能を実現する

ことが出来るようになってきた。

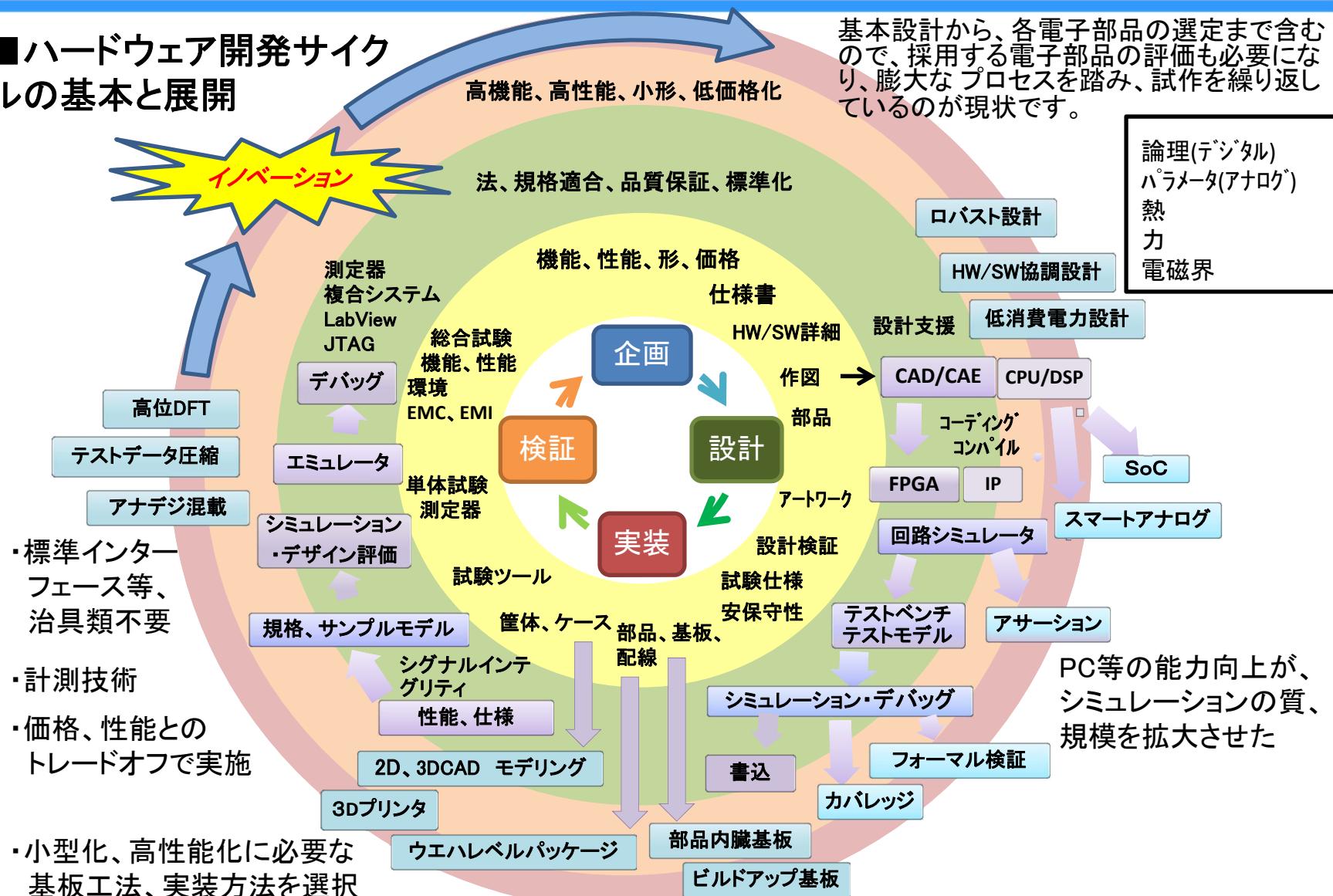
未来計画のロードマップを目指したとき、

組込みハードウェアにおける開発には何が必要か

組込み技術者に求められる新しいハードウェアスキルとは



■ハードウェア開発サイクルの基本と展開





■開発ステップでの新しい項目

プログラマブルデバイス

- スマートアナログ

設計(SoC設計、FPGA設計)

- モジュール間通信技術
- マルチプロセッサ
- リコンフィギュラブルロジック
- デバイスマデリング技術
- 回路シミュレーション技術
- 低消費電力設計
- アナデジ混載
- IPベース設計
- ライブラリ設計

デバイス(LSTP)

- センサーチップ
- 高速不揮発性メモリ

実装、部品

- 部品内臓基板
- ウェハレベルパッケージ
- シリコンパワーデバイス

ケース、筐体

- 2D、3Dモデリング
- 3Dプリンタ

テスト技術

- 高位DFT(Dsigne for test)
- テストデータ圧縮
- アナデジ混載

評価・解析技術

- 信頼性評価
- 製品の信頼性保証技術
- 不良原因の特定、故障解析



■組込み機器に求められる新しい技術

さらなるプロセッサ能力向上のために

- ・マルチコア化、 GPGPU
- ・光接続や高速信号での接続

さらなる小型化のために

- ・内蔵化など構造上の小型化
- ・極小部品の実装技術

さらなる省電力化のために

- ・電源制御の細分化方法
- ・バッテリーの長時間駆動方法
- ・エネルギーハーベストの構築方法

インターネット接続が必須に

- ・無線/有線LAN、通信網

さらなるリアリティ映像に向かって

- ・超高精細動画(4K × 2K)
- ・各種パネル、カメラ

なんでもワイヤレスに向かって

- ・無線通信規格、電波伝搬特性
- ・アンテナ技術、RF(アナログ)
- ・ワイヤレスネットワーク
- ・ワイヤレス給電
- ・テラヘルツ

スマート革命に備える

- ・スマートフォンなどのM2M通信
- ・ビッグデータの活用



■新しい技術に必要なスキルとは

- ・ 単体及び全体をシミュレーションにより設計確認できる
　テストモデル、テストデータ等の作成と結果の評価
- ・ CPU、SoC、FPGAおよび周辺の配置・配線・実装の判断
- ・ IPの組込み、DSPの組込み、高速化回路の実現とデバッグ
- ・ ワイヤレス化に伴うARIBなどの無線通信の規格または法律
- ・ 高速化における光インターフェース、SerDes等の実装方法
- ・ センサ接続、センサデータ処理方法、電源供給/制御



■開発技術者が必要となる一般スキル

- ・ プロジェクト実行計画、また完成度を高める為の手法を具体化
→ 企画、設計から完了の計画立案、または計画の理解
- ・ IT関連ベンダーや、開発機器(ツール)の 最新情報収集
→ 評価し習得する(習得スピードも重要)
- ・ イノベーション力や、仕様の提案力
→ 製品化とりまとめなど
- ・ 他国語での理解、判断
→ データシート、仕様書
- ・ PCからタブレットやスマートフォンなどへ派生する環境対応
→ 新しい使用方法、必要なアプリの把握



まとめ

- 小型化、高速化、省電力化、ワイヤレス化、スマート化は、重要な位置づけで益々進化を続けるテーマとなる
- 高度化・広範囲化に渡る知識を求められてきているが、電気・電子の基本技術を理解した上であることが重要
- プロジェクト上の役割を理解すること、要求事項(完成形)を理解して、機能、仕様、規格などを網羅し確認できる
- 技術面をはじめ、情報収集と整理、展開、活用

スマートフォンの登場、情報家電市場の拡大など製品機能の進化により、組込システム分野は変革期を迎えています。新しい技術は高度化していき、要求スキルも高まります。興味を持って探究できる様にする、組織的に教育環境を高めることなどに工夫していく必要がある



「組込み技術者に求められる新しいハードウェアスキルとは」

2013/5/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



組込み仮想化技術の メトリックスと性能評価に ついて

2013年5月22日
一般社団法人組込みシステム技術協会
OSS活用WG 組込み仮想化G
(株)アックス 竹岡尚三



組込み仮想化Gに ついて

WGメンバー



アプリケーション/ユーザ

ヤマハ

パナソニック

デンソー

NDD

(株)KSK

半導体/ハードウェア基盤

ルネサス・エレクトロニクス

富士通研究所

OS/ハイパバイザ・ベンダ

アックス

RedBend

Linx (独 RealtimeSystems社代理店)

※ユーザ企業と組込み仮想化技術の主なプレーヤの多くが参加



組込み仮想化の必要性

- 仮想化技術は求められている
- ハードウェア資源が豊富になった
 - 組込みでは、特に、CPU演算器に対して、IOがネック
⇒ CPUパワーは余っている
 - HDDレコーダや携帯電話では、メモリ(DRAM)は、最小単位が512MB程度に
- ライセンス問題から自由に
 - OSSライセンス(GPLなど)と、プロプライエタリものとの絶縁
- セキュリティ性を高める
 - 一台のスマートフォンに
 - 怪しいアプリケーションのための仮想機械
 - 信頼できるアプリケーションのための仮想機械
- エンタープライズでは、仮想化は実際に多数使われている

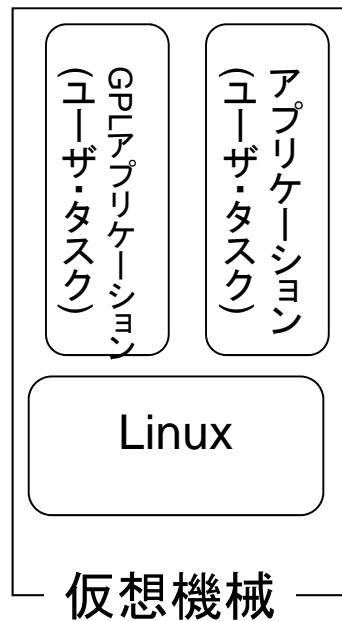
ライセンス問題の解消,GPLとの絶縁



GPLライセンスのソフトウェアとプロプライエタリなソフトウェアを別々の仮想機械に入れて、絶縁する。絶対に安全

徐々に、GPL側から、プロプライエタリなソフトウェアに書き換えて移行が可能

GPL パーティション



プロプライエタリ・パーティション
(GPLから守る)



ハイパバイザ

ハードウェア
(実機械)

高セキュリティ・スマートフォン

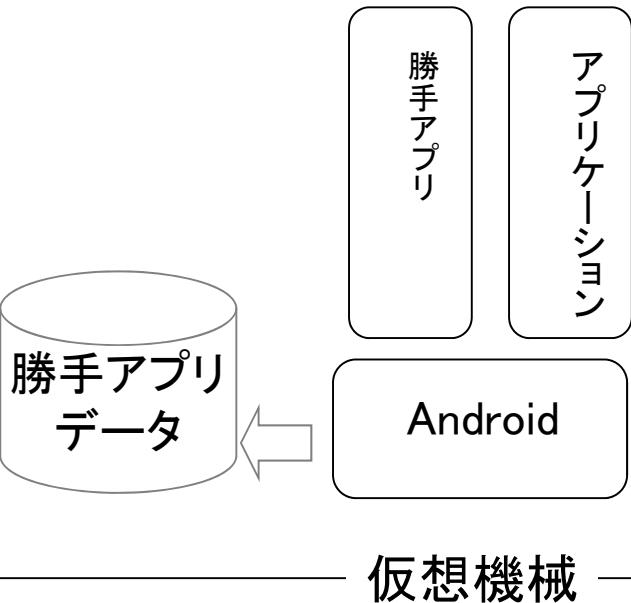


一つのスマートフォン(ハードウェア)に、

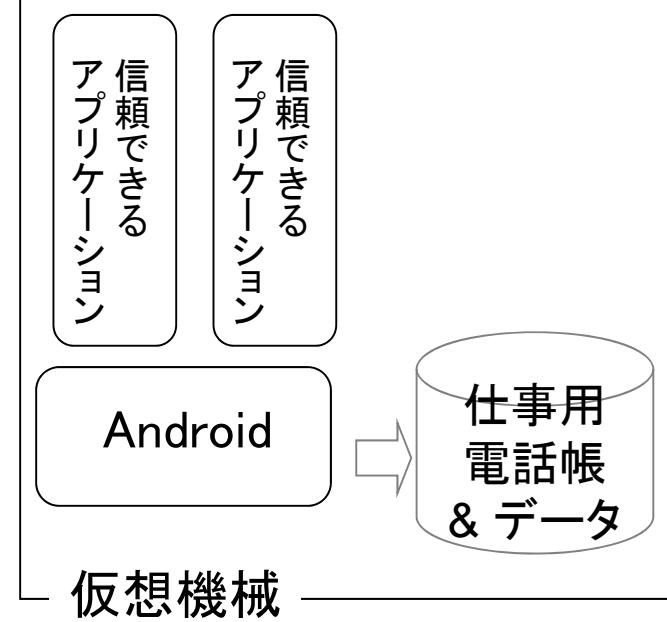
- ・勝手アプリ(信頼できないアプリケーション)を入れる仮想機械(OS)
- ・セキュアな(仕事用の電話帳、データを入れる)仮想機械

を収容

勝手アプリ・パーティション



セキュア・パーティション



ハイパバイザ

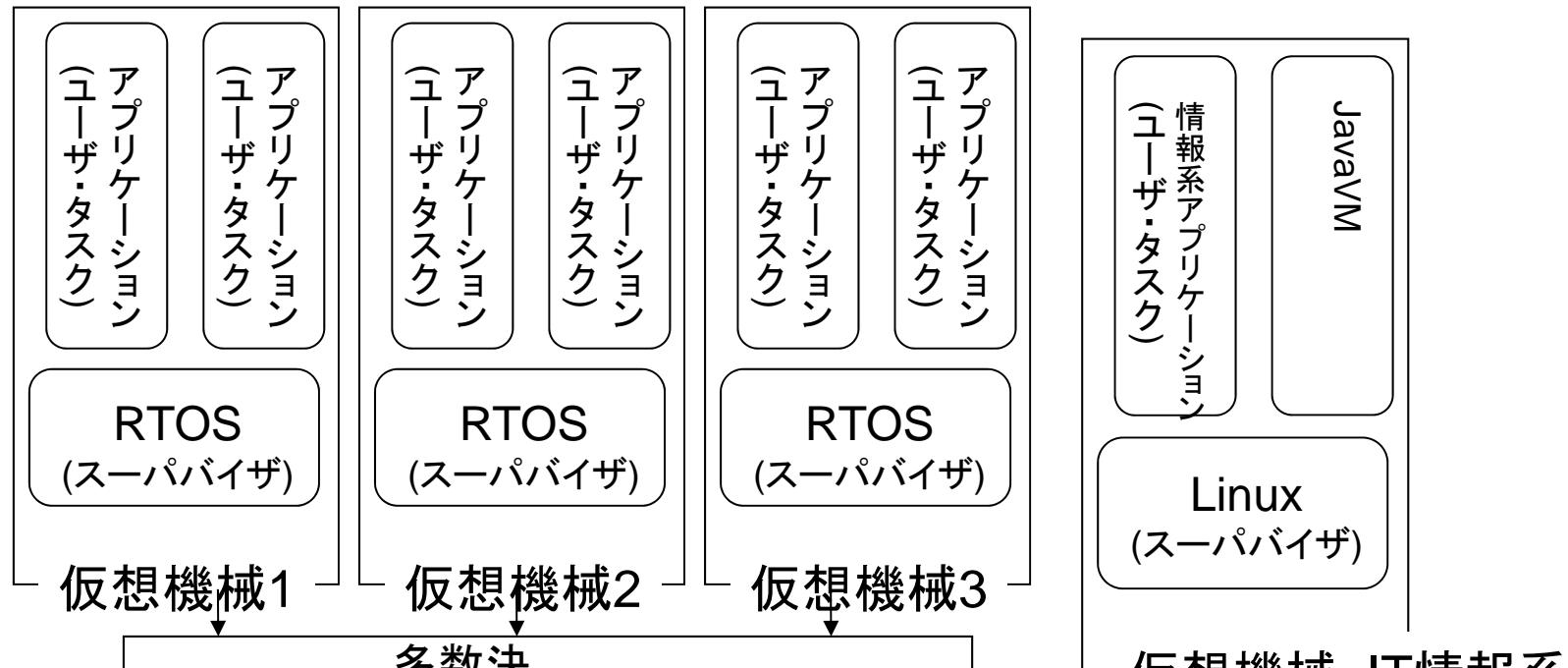
ハードウェア
(実機械)

機能安全,オートモーティブ



高信頼

IT情報系OSがフォールトしても、実時間系は影響を受けない。逆も同様



ハードウェア
(実機械)

組込み仮想化の必要性



- 資源を区画化(パーティショニング)して、仮想機械を動かす
 - メモリの絶縁
 - 個人情報保護。高セキュア
 - CPU時間の配分
 - 実時間性を保って、IT系OSを動かす
- 組込みのための仮想化とは?
 - 次ページ

組込み仮想化とエンタープライズ仮想化の違い



- エンタープライズ、データセンター、クラウド
 - スケジューリングはいい加減でも気にならない
 - 外部とのネットワーク接続がネック(律速)
 - CPU, メモリは余っている

⇒ 仮想化。一台のハードウェアに多数の顧客を収容

 - 絶縁(空間パーティショニング)がしっかりしていれば良し
- 組込み
 - 時間の配分、区画化が重要。実時間性。応答性の確保
 - CPU演算器の速さに対してIOが非常に遅い

⇒ CPU内では、CPUパワーは余っている

 - メモリがあるなら、仮想化
 - 空間の絶縁は大事

組込み仮想化の2010～12年の現状



- 言葉が乱れている
 - 定義はあるのだが、自分勝手な言葉をしゃべっている連中が居る
 - 自称「仮想化」が存在したり…
 - それぞれの技術が異なっていることも多い
 - 技術が進化中なので仕方ない面もある
- ユーザが、各製品を比較しにくい
 - 言葉が乱れている
 - 皆(主にソフトウェア提供者)が勝手になにかをアピールしている
 - 比較のための要素が分からぬ。評価の基準がわからぬ
- ユーザへのプロモーションが難しい
 - 仮想化の適用すべき場面が分からぬ
- デバイス、割り込みの仮想化の標準/評価基準が無い

組込み仮想化WGの目標



ユーザが分かりにくく、プロモーションが難しい



仮想化WGでなんとかしよう



マーケット創世 & ユーザへの役立ち

WGの具体的目標



- ▲ 仮想化技術の比較を容易にする必要性、有用性を認識させる
　　そのために
 - ▲ 用語集を作る
 - ▲ メトリック(尺度、分類の基準)を作る
 - ▲ 啓蒙活動を行う
 - ▲ 仮想化技術を分類
 - ▲ 各社製品の比較表を、JASAなりに作成
 - ▲ JASA版用語辞書を作る
 - ▲ それらをもとに、広く日本の共通認識を広める
- マーケットの創世



組込み仮想化WG これまでの成果

2010/10月～2011年3月の成果



- 啓蒙用資料が必要
 - まず、メトリック(尺度、分類の基準)を作る
- メトリック(尺度、分類の基準)を作る
 - 1)比較表を作る。各項目がメトリックである
 - 2)メーカー、ベンダーの記入できるシートを作る。各項目がメトリック
 - 3)定性的な評価基準ができた
- 2010年度、上記の比較表を作った
 - フィードバックを得て隨時更新中

1)仮想化技術の比較表のメトリック



分類

仮想化タイプ: 空間パーティショニング, 時間パーティショニング

- 諸元 --

サポートCPU

ホストOS

サポート(ゲスト)OS, ゲストOS提供者

--商品情報--

ライセンス形態, 価格, URL, 場所, 主要マーケット, 日本代理店, 備考,

- 機能 --

仮想ドライバ取り揃え, 動的マイグレーション,

仮想マシン間通信

HVサービス, メモリ要求, アイソレーション階層, 時間保証パーティション有無

--性能--

最短応答時間, 最悪応答時間, 平均応答時間, 最大仮想マシン数, セキュリティ・レベル, 機能安全レベル

2)仮想化の実現記述シート



- 仮想化の実現記述シート
- 各ハードウェアごとシートを作成
- そのハードウェアで実現されているゲストOSについて詳細に記載
- シート中の各項目
 - 仮想化資源
 - ハードウェア
 - 仮想化ソフトウェア(Hypervisor)
 - オペレーティングシステム(Supervisor)
 - デバイスドライバ

2011年度の成果



- ベンチマーク・プログラムを収集し、組み込み仮想化システムで実行
2社の仮想化システムで、実際の性能を測定した
 - RedBend社”VLX”、アックス社”蛍”
 - 興味深い結果が得られた
 - 定量的なメトリックの一部ができた
- 仮想化システム比較表
 - 更新
 - 比較要素のリファイン

ベンチマークで測りたいもの(1)



● 仮想化によるCPU負荷の増大

- ハイパバイザの走行時間 (CPU負荷増大)
- 仮想化によりアプリケーションの実行時間が増えるであろう (ハイパバイザコールのオーバヘッド)
- システム全体のスループットが測れる
 - ピーク性能、平均性能は測れる
- メモリの使用量も見ておきたい(VM単位なので、大きな粒度になろうか?)

● レイテンシ (測るのにハイパバイザに機構が必要か)

- 仮想化でのレイテンシとは?
- VM起床遅延のこととするか?
- VMの理想的起床時間となにか?

ベンチマークで測りたいもの(2)



※以下は、2011年度には、計測できなかった。2012年度に実施中

- デッドラインが守れているか

- 時間にに関する最悪値を知りたい
- アプリケーション・タスクは起床時間を守れるか

- 割込み遅延, 割込みの取りこぼし率

- 最悪値を知りたい
- ハード操作が容易なら測定は比較的容易だろう

ベンチマークの具体的な要件(1)



- 古典的ベンチマークを、以下のシステムで一通り行う

- 素のシステム
- 仮想化済み、VMは1つ
- 仮想化済み、VMは2つ

- 単純なベンチマークが仮想化によりどれくらい遅くなるか？

- それで「仮想化オーバヘッド」というようなものが明らかになるだろ
- VMスケジューリングが発生するであろうから
- 素 vs 仮想化のシステムの性能差=仮想化オーバヘッド (?)

- ベンチマークと一緒に動かす負荷の組み合わせが大事

- 同じプログラムを動かすと、妙な安定が起こる可能性あり
- 実アプリケーションより、いい性能が出る可能性あり。
- 周期があるとすれば、それが互いに素になるようにすべき
- 最小公倍数を大きく or そもそも周期が無くなるように、ランダムさを入れるべきか

ベンチマーク・プログラム



- 古典的ベンチマークを一通り収集
- 総合的なベンチマーク
 - Byte benchmark
 - LMbench
 - gcc/コンパイラ
- ファイルIO, 総合的(?)なベンチマーク,
 - grep
 - find
- ネットワーク, 総合的な(?)ベンチマーク
 - netperf
- ミクロなベンチマーク
 - dhrystone
 - bcopy

※残念ながら、組込み制御アプリケーションは無し

ベンチマーク結果



● AXE製ハイパバイザ「蛍」について実施

- 実施ハードウェアなど
- Hardware: Armadillo-500 FX
- Processor: ARMv6-compatible processor rev 4 (v6I)
- BogoMIPS: 532.48
- Linux version 2.6.26-at14 (実機械、仮想機械などすべてで)

● VMへのCPU割り当てポリシー

- VMが2つの時
 - VM1:VM2 = 8:2 でCPUを割り当てるように設定

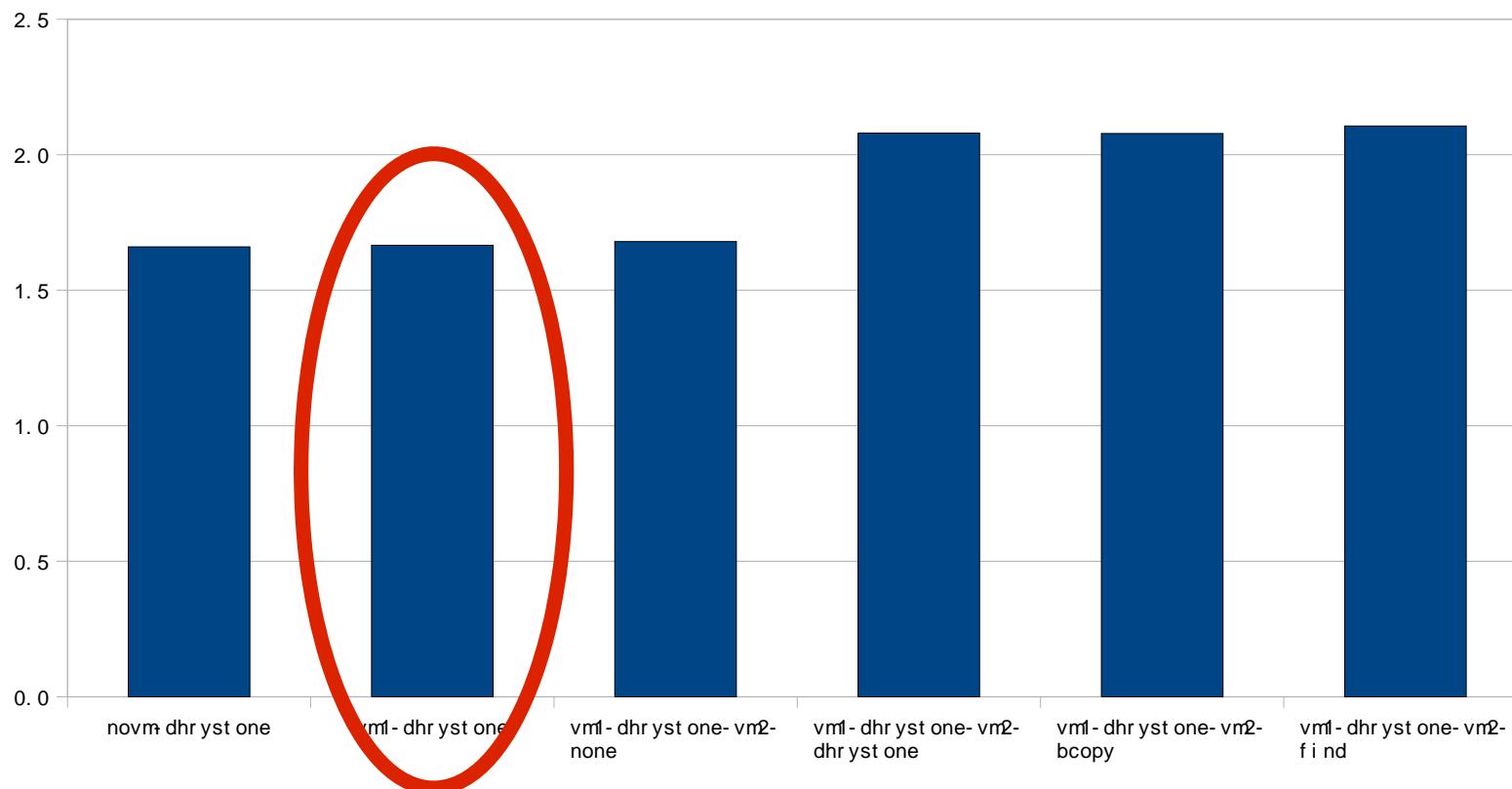
ベンチマーク結果



● Dhrystone

- 計算が小さい時は、仮想化(VM1つ)した方が性能が高い

Microseconds for one run through Dhrystone



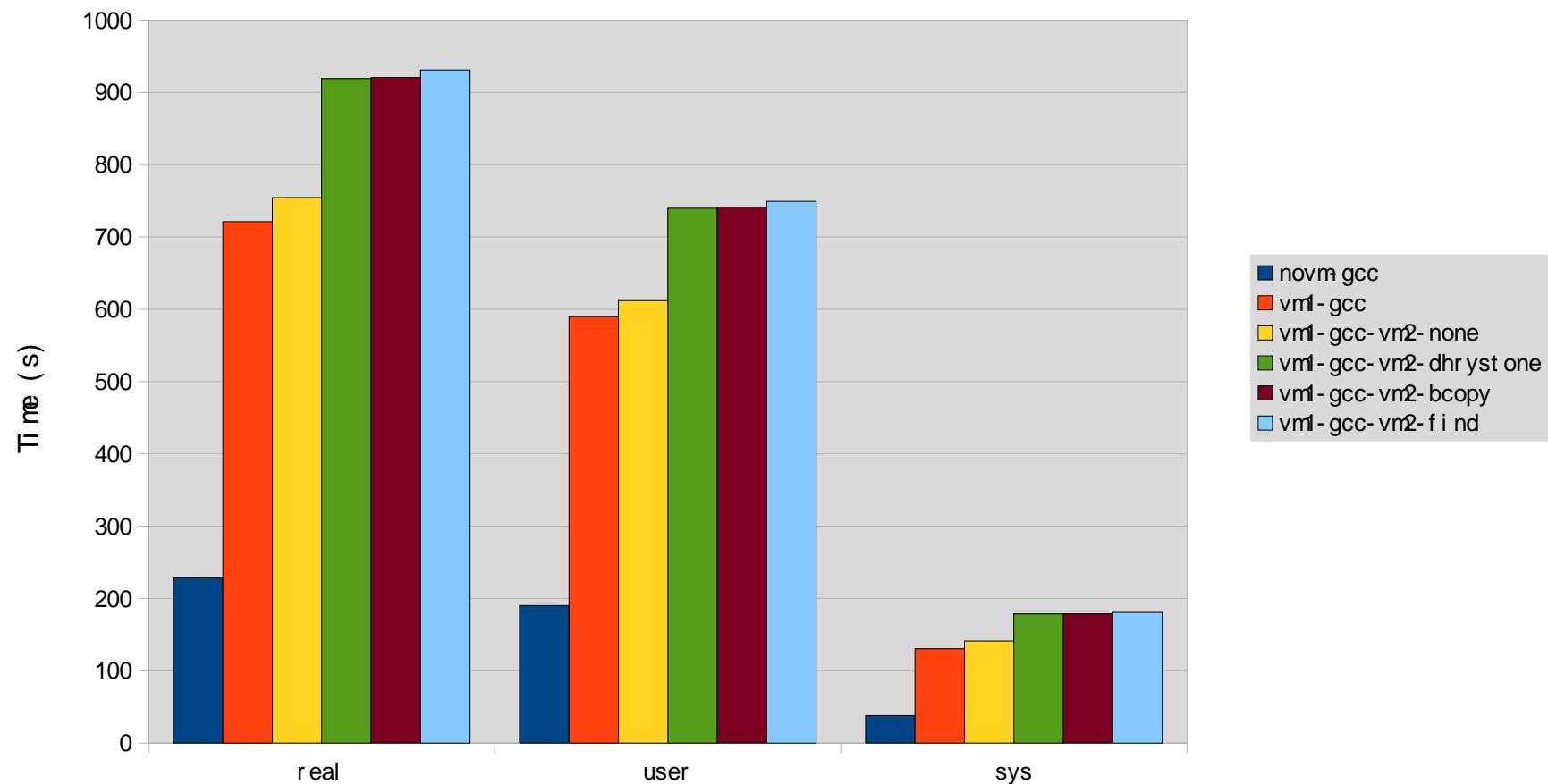
ベンチマーク結果



gcc : CPUをよく使い、メモリもよく使用、ファイル操作もあり

一般に予想されるとおりの結果

gcc (a)





2012年度 中間成果

2012年の目標



- 実時間に関する、定量的なメトリックを作る
 - 実時間に関するベンチマーク・プログラムを作成収集
 - ベンチマークの実施
 - WG参加メンバー各社で実施したい
- ユースケースを収集
 - どういうシステムで、仮想化が有効か
 - 事例を収集、提示できるようにする
- これまで作成したメトリックの見直し

2012年の目標 実時間に関するベンチマーク



- 実時間に関するベンチマーク・プログラム
- 実際の使用に近い状況を作る
- ユースケースを想定
 - ユースケースに沿ったベンチマーク・プログラムを作成
 - 実使用とかけ離れた計測結果は、不要
- 実際のユースケースも考えた



実時間その1 周期起床

1. 仮想化して、どれだけジッタが増加するか?

周期起床を計測。ジッタをヒストグラムで見たい

1) 周期は、50mSec

周期: 1Secも余裕があれば測定

2) 周期 50uSec

- 仮想化されたシステムでサービス可能か?
- 確率的にサービス可能か否か?
 - 確率的にサービス可能とは、時々ダメだがそれなりに使える

計測は長時間に渡って行うべし

- 数十時間はできることが望ましい
- まれに発生するジッタが存在する

※ここでいう「ジッタ」とは、理想的な時刻に対して、起床時刻が前後にぶれることを言う



実時間その2 割り込み応答

2. 割り込み応答時間を測りたい

- 素の Linux
- 仮想化済みLinux(VMM+Linux)

両者での割り込み応答時間を測る

- 1) 通常割り込み処理ルーチンの入り口で、計測
- 2) Linuxの割り込みディスパッチ・ルーチン(do_IRQ)の入り口で測る

この計測で得たいデータ

- 割り込み応答時間の増加
- 割り込み応答時間のばらつき
 - 割り込み応答時間の最大と最小の差が大きいと良くない(ジッタ(揺らぎ)があると考えられる)



組込み仮想化技術のメトリックスと性能評価について

2013/05/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。

実装品質強化委員会 WG2

2012年度活動報告・2013年度方針

2013年5月22日
東洋大学 野中 誠

実装品質強化委員会 これまでの活動範囲

- 組込みシステムの開発現場の人達が「使う気になる」
ガイドライン策定のためのベース作りを行う
 - ツール利用状況アンケートの分析 (WG1)
 - 実製品における非機能要求実装方法の調査(WG2)



競争優位の要因となる品質要素に的を絞った
ガイドライン策定へつなげていく

戦略的メッセージ、2012年度活動内容

『組込みシステムの競争優位要因となる品質要素を、エンジニアリングによっていかに実現するか』

- ・競争優位の要因となる／なった品質要素は何か
 - ・その品質要素は、どのように要求定義されたのか
 - ・その品質要素は、どのように仕様化されたのか
 - ・その品質要素は、どのように実装されたのか
 - ・その品質要素の達成度合いは、どのように評価されたのか
-
- ヒアリングによる情報収集（2012年度）
 - ・三つの製品について、複数部門からヒアリングを実施
 - 対象製品：カーナビ、デジタルミキサー、プリンター
 - ヒアリング相手：製品開発、組込みソフト開発、マーケティング

2013年度の活動：ガイドライン作成

- **INPUT：ヒアリング記録**

- 顧客満足に関わる組込みシステムの品質要素は何か？
- それらの品質要素に対する組込みソフトの貢献は何か？
- それらの品質要素に対するソフトウェア品質要求は何か？
- それらの品質要素のソフトウェアによる作り込みと評価の方法は？

- **PROCESS：委員会での検討（合宿！）**

- 顧客視点の品質要素と、技術視点の活動（実装／評価）を結びつける
 - 品質機能展開（QFD）等の技法を用いる
- ヒアリング記録に基づきつつ、委員の豊富な実務経験を加味する

- **OUTPUT：ガイドライン**

- 競争優位の品質要素を、エンジニアリングとして実装する方法の指針



アジャイル適用の課題と、 解決にむけての考察・実践

2013年5月22日
アジャイル研究会
島田 尚弥

目次



1. 研究会発足背景と活動内容
2. アジャイル開発とは
3. アジャイルの試験導入と評価
4. 今後の活動について

ウォーターフォールモデル



- 1980年代、米国防総省(DOD)はウォーターフォールモデルと文書駆動型を核とした開発を標準化し、推奨した。(DOD-STD-2167)
 - しかし、2167に準拠した開発の75%が失敗に終わる。
 - 2000年に反復型開発を推奨する指導文書DOD5000.2を発表。
-
- 前工程が完全である事が前提の開発プロセスだが、実際の現場ではこの前提が成り立たない事が多い。

手戻り工数増加・納期遅れ・品質低下



■ 研究会発足背景

- ・ アジャイル開発を組み込みソフト開発に適用すること業界が抱える課題を解決することはできないか？

■ 研究会目的

- ・ アジャイルソフトウェア開発手法の導入ガイドラインの策定、または、開発プロセスデザインのガイドラインの策定

■ 2012年度活動実績

- ・ 2012年4月アジャイル研究会発足
- ・ アジャイルソフトウェア開発手法の調査
- ・ 実践と評価

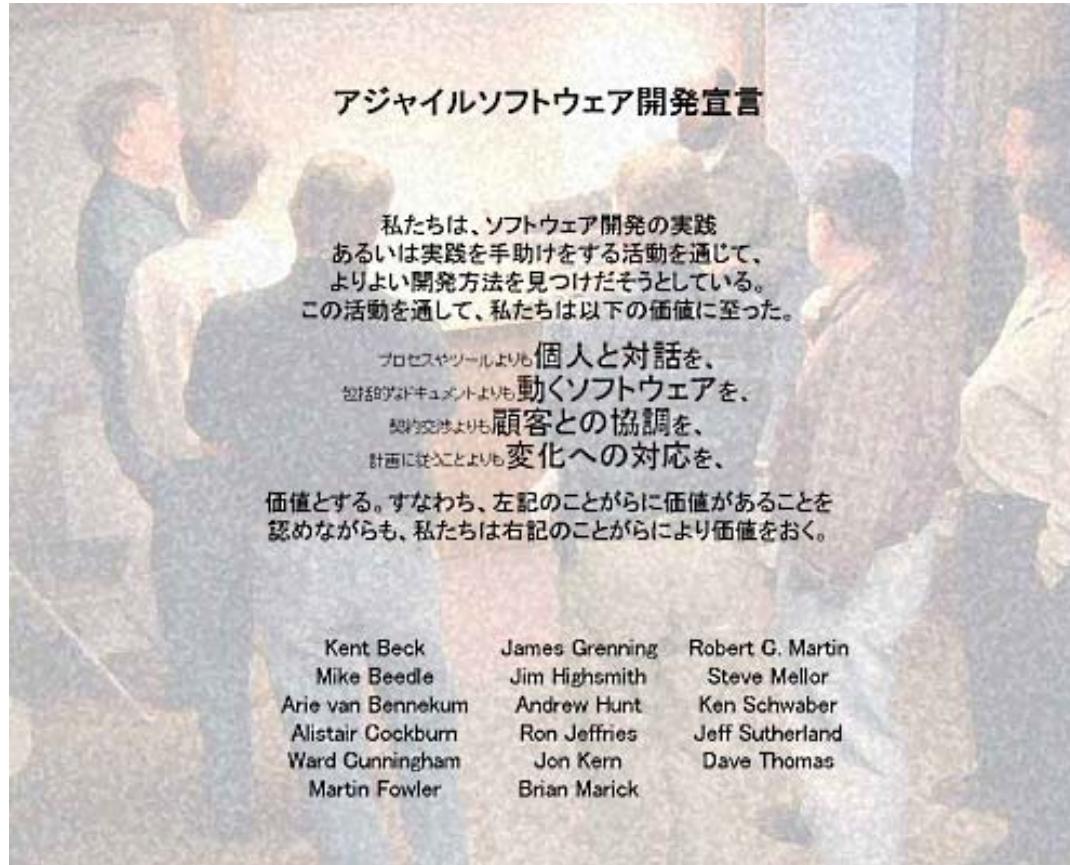


アジャイルソフトウェア開発

アジャイルソフトウェア開発とは



- Agile:[形容詞] 機敏な、俊敏な
- 迅速かつ適応的にソフトウェア開発を行う軽量な開発手法群の総称



アジャイル原則



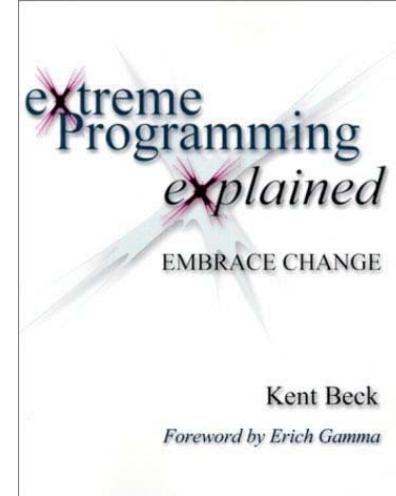
- 顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供します。
- 要求の変更はたとえ開発の後期であっても歓迎します。
- 変化を味方につけることによって、お客様の競争力を引き上げます。
- 動くソフトウェアを、2-3週間から2-3ヶ月というできるだけ短い時間間隔でリリースします。
- ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働くなければなりません。
- 意欲に満ちた人々を集めてプロジェクトを構成します。
- 環境と支援を与え仕事が無事終わるまで彼らを信頼します。
- 情報を伝える最も効率的で効果的な方法はFaceToFaceで話をすることです。
- シンプルさ(無駄なく作れる量を最大限にすること)が本質です。
- チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。

顧客満足を最優先し、より早く価値あるソフトウェアを提供するため、常に最適な開発プロセスに改善し続ける事。

主なアジャイル開発手法



- XP(ExtremeProgramming)
- Scrum
- クリスタルファミリー
- FDD(Feature-Driven-Development)
- MDA(Model Driven Architecture)



従来のプロセスは「開発が進むにつれて、変更コストが増大する」という前提のうえで、「開発が進んでも変更コストが増大しない」ように工夫された手法。

要求・仕様は変更されるものであり、それを許容する



アジャイル導入を阻む壁

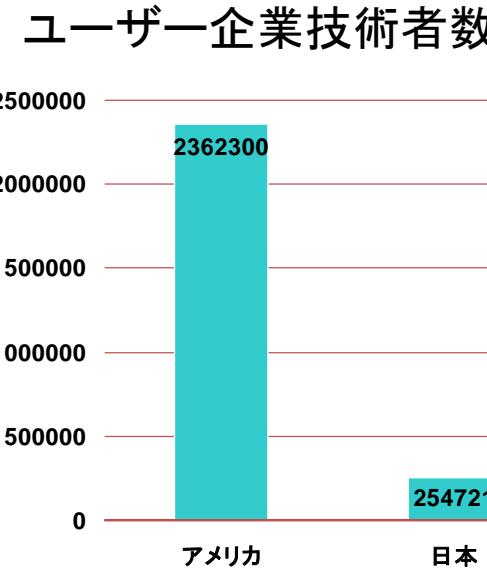
なぜ日本での普及が遅れているのか



- IT企業(特にWeb、ゲーム業界)では、徐々に普及し始めているが、まだまだ普及率は低い



※IPA グローバル化を支えるIT人材確保・育成施策に関する調査報告書

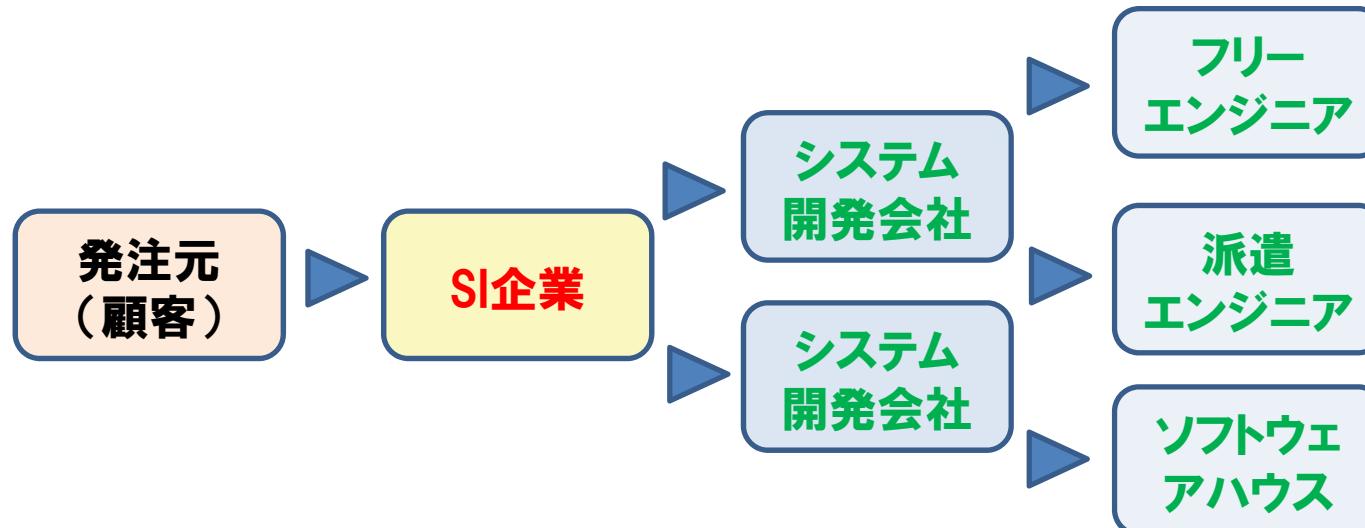


国内情報サービス産業の売上高の約5割は「受託ソフト開発」

受託開発に最適なウォーターフォール



- 要求定義の工数見積にしたがって一括請負契約を結ぶことができる
- 承認プロセスが明確であるため責任の所在が明確
- 開発者の担う役割が固定的かつ部分的で、低スキルの技術者でも参画しやすい



合理化 および 部分最適

アジャイルへの不十分な理解



ドキュメントは書
かない？

ドキュメントは書く
ソフト>ドキュメント

いつでも仕様変
更可能？

無制限ではない

ペアプロで
生産性上がる？

下がる
トータルでは上が
る事がある

計画は立てな
い？

むしろ
アジャイルの方が
計画的

WFより短納期
で開発できる？

「おわる」の
定義次第



アジャイル導入の課題と実践

対象プロジェクト



- 開発メンバー: 7人 (PL+開発者)
- 開発プロセス: ウォーターフォール
- 対象製品: センサーデータを用いたクラウドシステム (自社他部門からの受託開発)
 - Webアプリケーション開発、DB開発、組み込みソフト開発
- 抱える課題:
 - 短納期によりテストがおろそかになり、品質低下
 - 要求の追加・変更による手戻りコスト増
 - メンバーの入れ替えによる引き継ぎコスト増

アジャイル開発手法の評価のため試験導入

スクラム導入における課題



導入時の課題

手法	概要	課題
定期リリース	固定化した短期間(1-2週間)でリリースを行う	要件や機能によって顧客の希望納期が異なり、リリース間隔の固定が難しい
メンバーの固定化	プロジェクトの最中はメンバーの変更は行わない	組織・業務の特性上、メンバー入れ替えが発生し、固定化するのが難しい
自己組織化	スクラムマスター(PL)は直接の指示は行わず気づきを与え、作業の割当も行わない	開発担当者全員が自発的に行動する必要があり、全員の意識を変えるには時間がかかる

対策



現場にあった形にカスタマイズ

手法	課題	対策
定期リリース	要件や機能によって顧客の希望納期が異なり、リリース間隔の固定が難しい	スプリントは可変とし、対象スプリントで対応する要件・機能のボリュームによって設定する
メンバーの固定化	組織・業務の特性上、メンバー入れ替えが発生し、固定化するのが難しい	1ストーリーに対して二人の担当者を割り当て、ペアワークという形で情報共有の円滑化を行う
自己組織化	開発担当者全員が自発的に行動する必要があり、全員の意識を変えるには時間がかかる	作業割り当てはスクラムマスターが行う



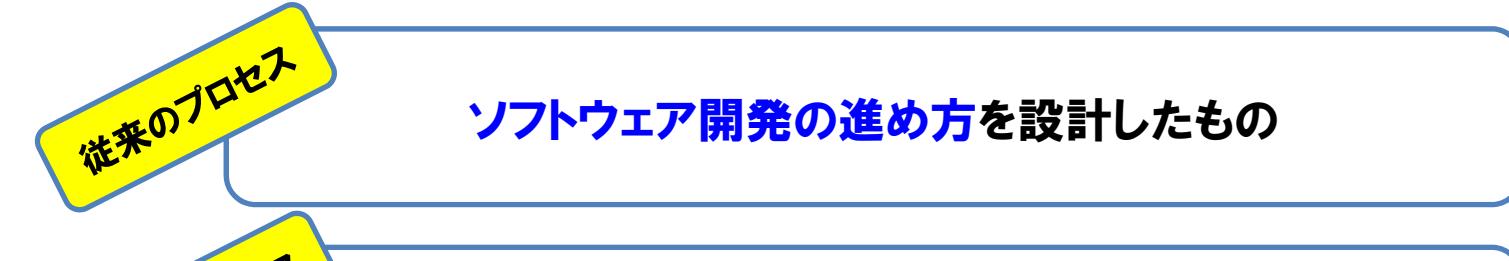
期待していた効果と得られた効果

期待効果	実際の効果
手戻りコストの低減	定期的にリリースし、顧客からフィードバックを得ることで手戻りコストが大幅減
生産性向上	従来の方法と変わらない。テストを何度も繰り返すため、むしろ生産性は低下
トラックナンバー向上 (メンバー離脱時のリスク)	「特定の人物だけが知っている」状況からの脱却に成功 (トラックナンバー向上)

おおむね予想どおりの結果



- 従来のプロセスとは根本的に概念が違う



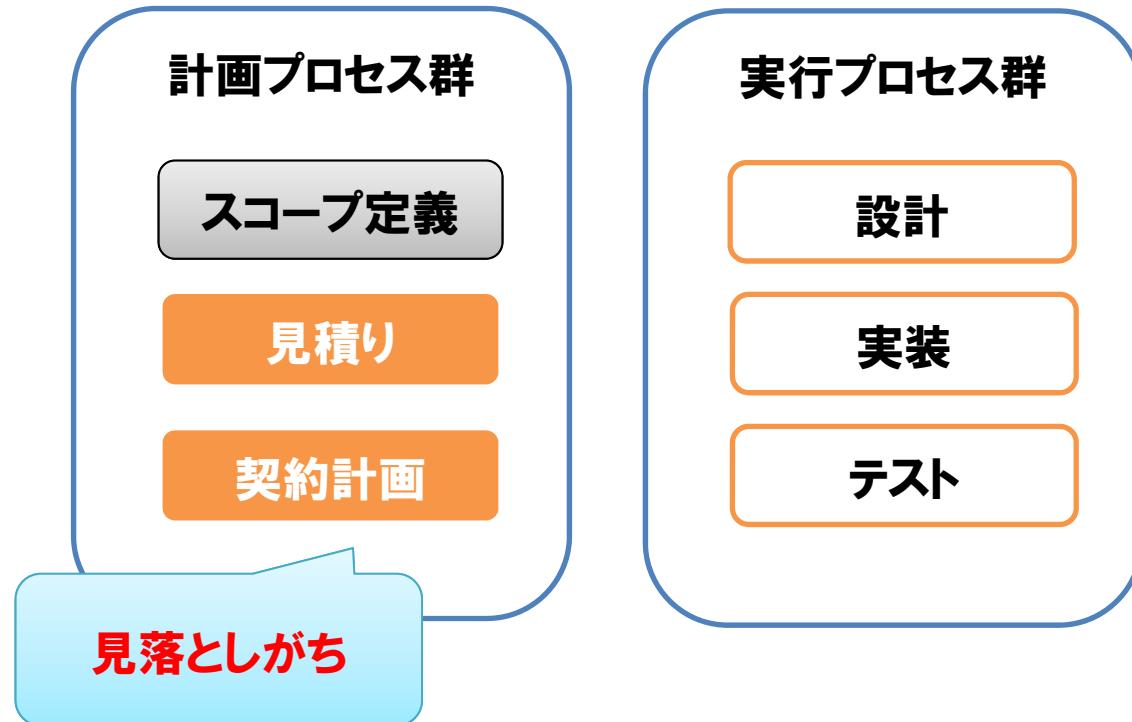
部分最適化ではなく全体最適化を行う必要がある

- 数ある手法を1つ1つ単体では語れない

部分導入する場合は、それぞれの相関関係を考慮



■ 見落としがちな計画プロセス



実行プロセスにあわせて計画プロセスも見直す
必要がある



組み込みソフト開発における プロセスデザインガイドラインの策定

■課題の抽出

組み込みソフト開発の現場で抱える課題の抽出

■考案プロセスの実践・評価

課題解決に向けて考案したプロセスを実際の開発現場で実践し、導入効果を評価

■ガイドライン策定

実践・評価を元にガイドラインを策定

■フィードバックからブラッシュアップ

ET2013にてガイドラインの発表。外部からのフィードバックを受けブラッシュアップをはかる

Thank you for your kind attention



研究会員

株式会社 イーシーエス:青田健太郎 島田尚弥

株式会社 ヴィツツ:脇田周爾 深見修一朗 小林一夫

株式会社 コア中部カンパニー:佐藤崇

株式会社 サンテック:水野基司 片山貴晴

株式会社 パッファロー:石井秀輝

東海ソフト 株式会社:秋谷勤

東洋電機 株式会社:熊谷秀雄

萩原電気 株式会社:佐藤博昭 加藤澄夫 広瀬勝規 水谷紘也

JASA中部 アジャイル研究会

www.jasa.or.jp/chubu

「アジャイル適用の課題と、解決に向けての考察・実践」

2013/5/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。



安全性向上委員会(2)

要求の仕様化に関する “要求仕様”と手法

～必ずしも“厳密さ”ではなく
“意図”したものが実現できる要求定義を求めた活動の報告～

2013年5月22日
安全性向上委員会
株式会社レンタコーチ
中村 洋
nakamura@rentaco.jp

内容



1. 活動計画と活動実績
2. 2012年度成果報告
 1. 要求の仕様化に関する要求と課題
 2. 要求の仕様化を支援するプロセス
 3. 仕様記述実験
3. 目指す方向と今後の予定



■ 委員会構成

- ・ 昨年度までの情報セキュリティWGと製品安全WGを一つにまとめて活動する。
- ・ 新たにSSQを活動テーマとする。

■ 主要課題

- ・ 機能安全IEC 61508/ISO 26262関連調査研究(継続)
- ・ 情報セキュリティ関連調査研究(継続)
- ・ 要求の仕様化に関する調査研究(新規)

SSQ: セーフティ、セキュリティ、品質

要求の仕様化に関する調査研究



■ 目標とする成果物

- ・ 要求の仕様化に関する要求事項や課題
- ・ 要求の仕様化を支援するプロセスと手法やツール
- ・ 各層における発注・受託関係者に役立つ報告書

■ 活動方法

- ・ 機能安全や情報セキュリティ関連の規格を対象として、要求事項を洗い出す。
- ・ REBOKや要求工学などの専門書を調べて、要求の仕様化のプロセスと課題を整理する。
- ・ モデルベース開発で使われている手法やツールを調べる。
- ・ 実際に使われている形式手法とそのツールを調べる。
- ・ JASA会員を対象としてアンケートを取り、プロセス、手法やツール、課題に関する実態を調査する。



活動実績の概要

■ 委員会での討議

- ・ 月1回のペースで、9回
- ・ REBOK、ISO 26262、構造化分析等を調査し、半形式手法・形式手法を調査検討
- ・ 電気ポットを題材として仕様記述を実践

■ 合宿研修

- ・ 9月21-22日、3課題の集中検討

■ 対外発表

- ・ 6月14日、ET-WEST2012にて計画を紹介
- ・ 11月14日、ET2012にて活動状況を報告



要求の仕様化に関する要求と課題

ISO 26262: 安全要求の定義と管理



対象分野	要求事項
表記法	特徴の実現のため、自然言語と別表の手法を組み合わせる
特徴	曖昧さを避けるため、安全要求は分離されている 安全要求は、曖昧でなく、理解しやすく、不可分で、矛盾せず、実現可能で、検証可能である（要求の特性）
属性	識別番号、状態、ASIL
管理	安全要求は階層構造をとり、組織的に体系づけられ、完全性を持ち、外部一貫性を保ち、重複なく、保守容易である 安全要求は上下方向に追跡可能であり、設計、検証とも追跡可能 別表の手法を組み合わせて検証する 安全要求は構成管理に従う

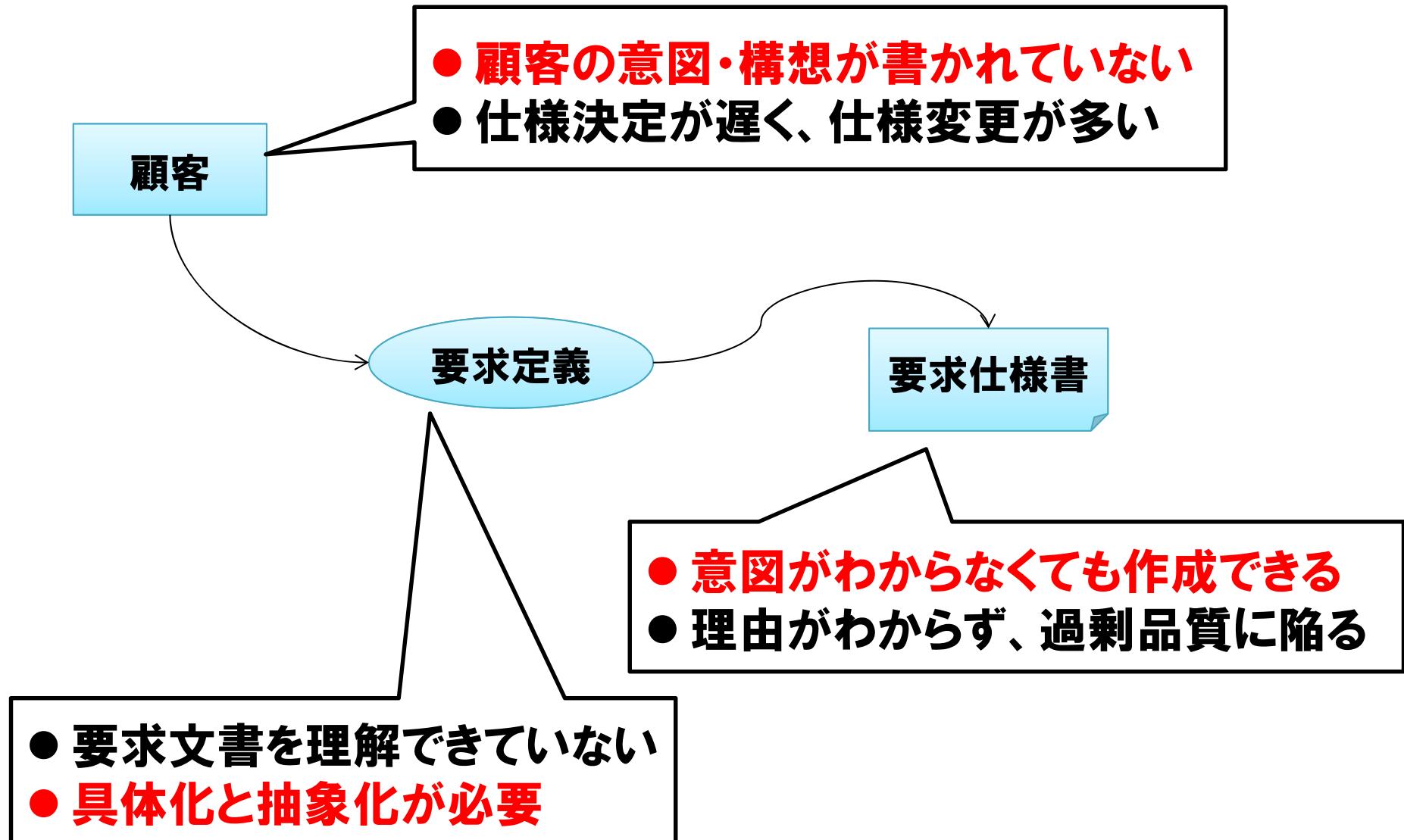
表記法	ASIL-B	ASIL-C	ASIL-D
形式的でない表記法	HR	R	R
半形式手法	R	HR	HR
形式手法	R	R	R

REBOK: 要求の特性



特性	意味
单一性	要求の対象が一つであること
完全性	要求に漏れや不完全な記述がないこと
一貫性	要求に矛盾がないこと
法令遵守	法律や規制に準拠していること
独立性	要求がそれ自体で完結し、他の要求に依存していないこと
追跡可能性	要求の源泉や設計など、前後の工程の成果物と関連付けることが可能であること
最新性	要求が最新の条件に基づいていること
実現可能性	要求がプロジェクトや環境などの特別な制約なしに、実現可能であること
無曖昧性	要求の中に曖昧さがないこと（たとえば、二つ以上の解釈がないこと）
必須性	要求が必須の内容（それが欠けると、要求の意味がなくなったり、不具合となるもの）を含むこと
検証可能性	要求が検証可能であること

開発現場における課題



一般社団法人
組込みシステム技術協会

Japan Embedded Systems Technology Association

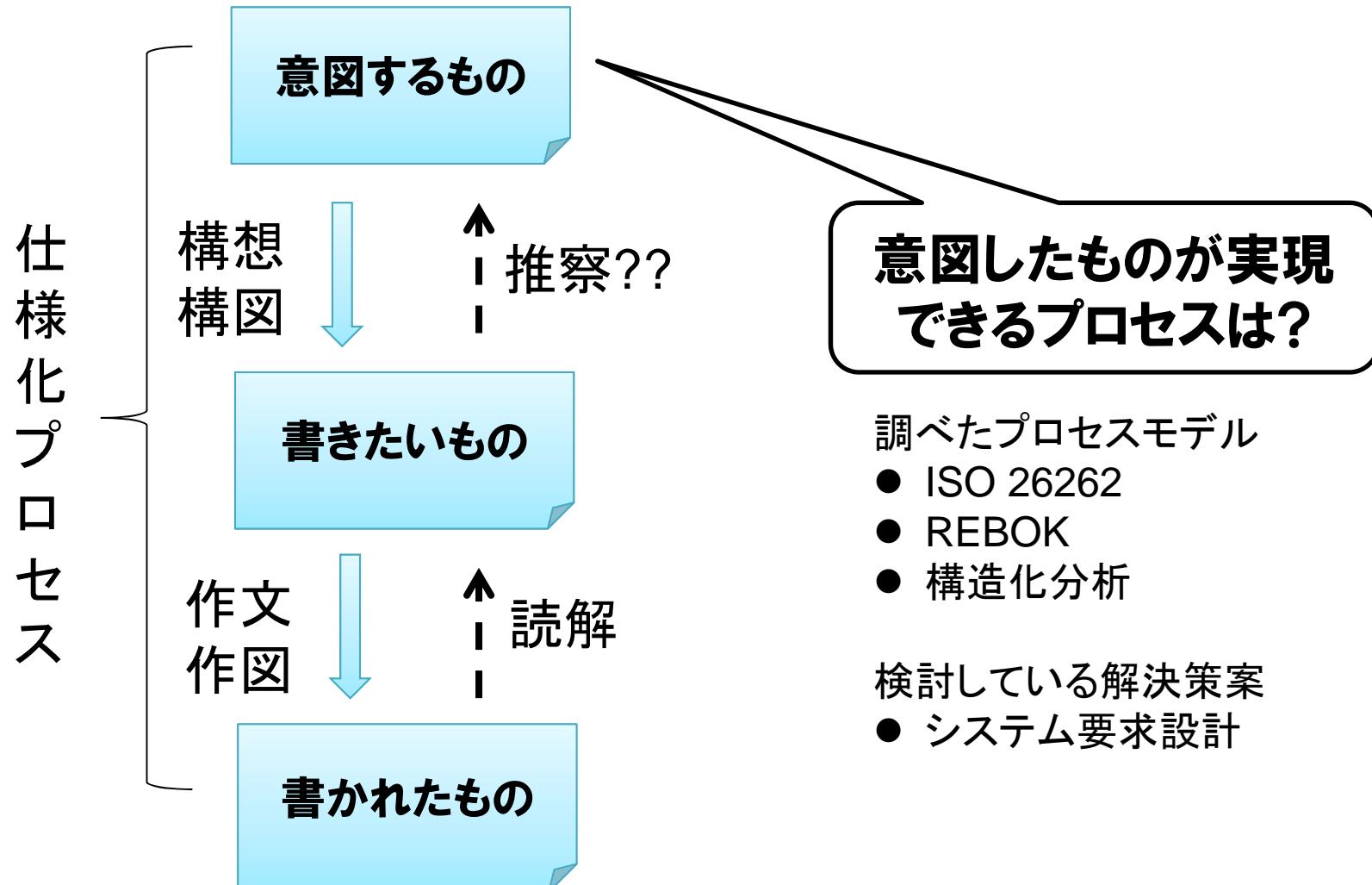
© Japan Embedded Systems Technology Association 2012-2013



意図したものが実現できるプロセス

要求の仕様化を支援するプロセス

要求の仕様化を支援するプロセス



ISO 26262: 主要プロセス



構想フェーズ

- アイテム定義
- 安全ライフサイクル開始
- ハザード分析・リスクアセスメント
- 機能安全構想

赤字表記のプロセス
が要求定義に関わる

システム開発フェーズ

- システム開発開始
- 技術安全要求定義
- システム設計
(ハード・ソフト開発)
- アイテム統合・テスト
- 安全妥当性確認
- 機能安全アセスメント
- 生産リリース

生産と運用フェーズ

- 生産
- 運用、サービス、廃棄

ハードウェア開発フェーズ

- ハードウェア開発開始
- **ハードウェア安全要求定義**
- ハードウェア設計
- ハードアーキテクチャ指標の評価
- 故障による安全目標違反の評価
- ハードウェア統合・テスト

ソフトウェア開発フェーズ

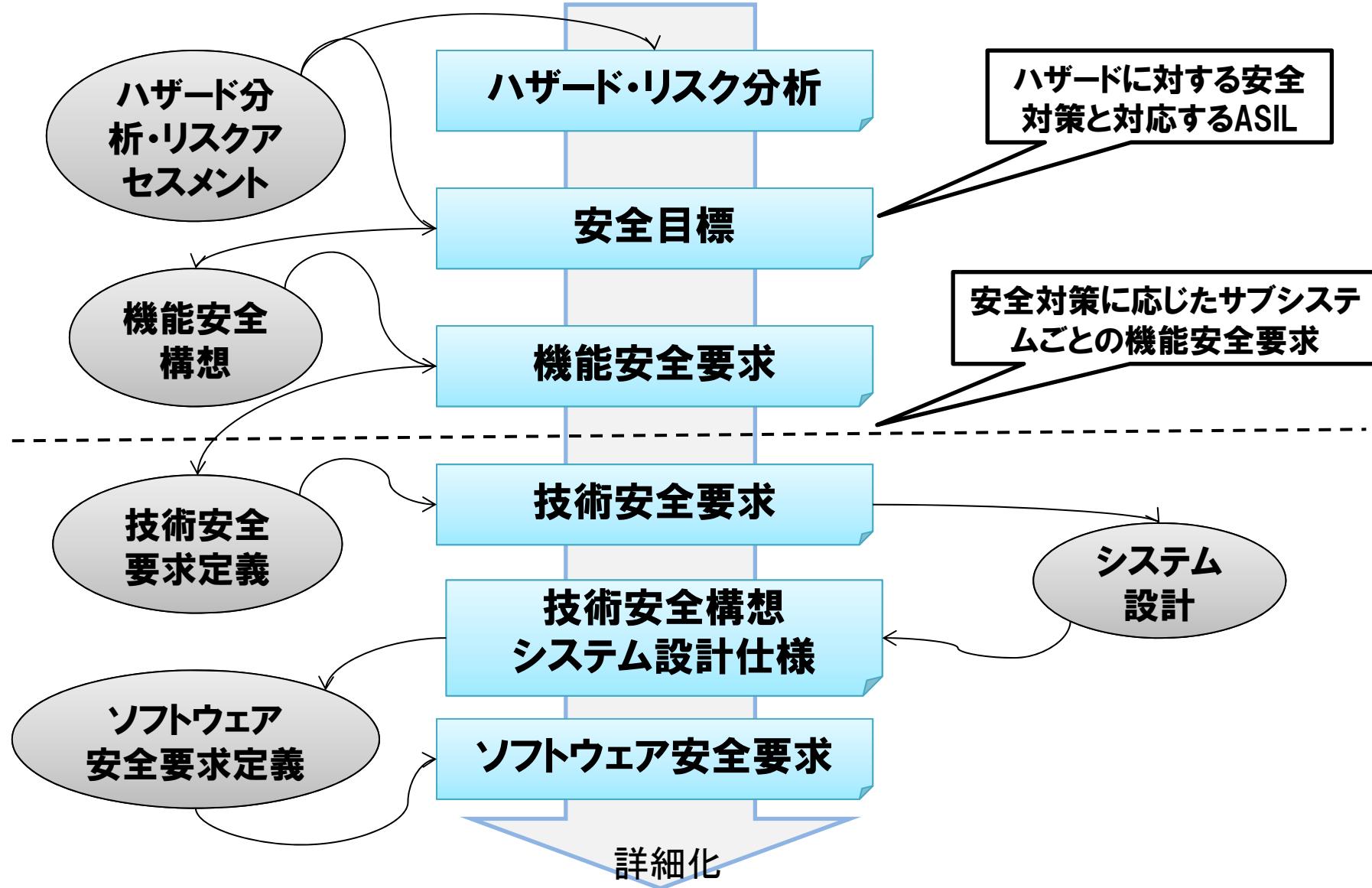
- ソフトウェア開発開始
- **ソフトウェア安全要求定義**
- ソフトウェアアーキテクチャ設計
- ソフトウェアユニット設計と実装
- ソフトウェアユニットテスト
- ソフトウェア統合・テスト
- ソフトウェア安全要求検証



一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association

© Japan Embedded Systems Technology Association 2012-2013

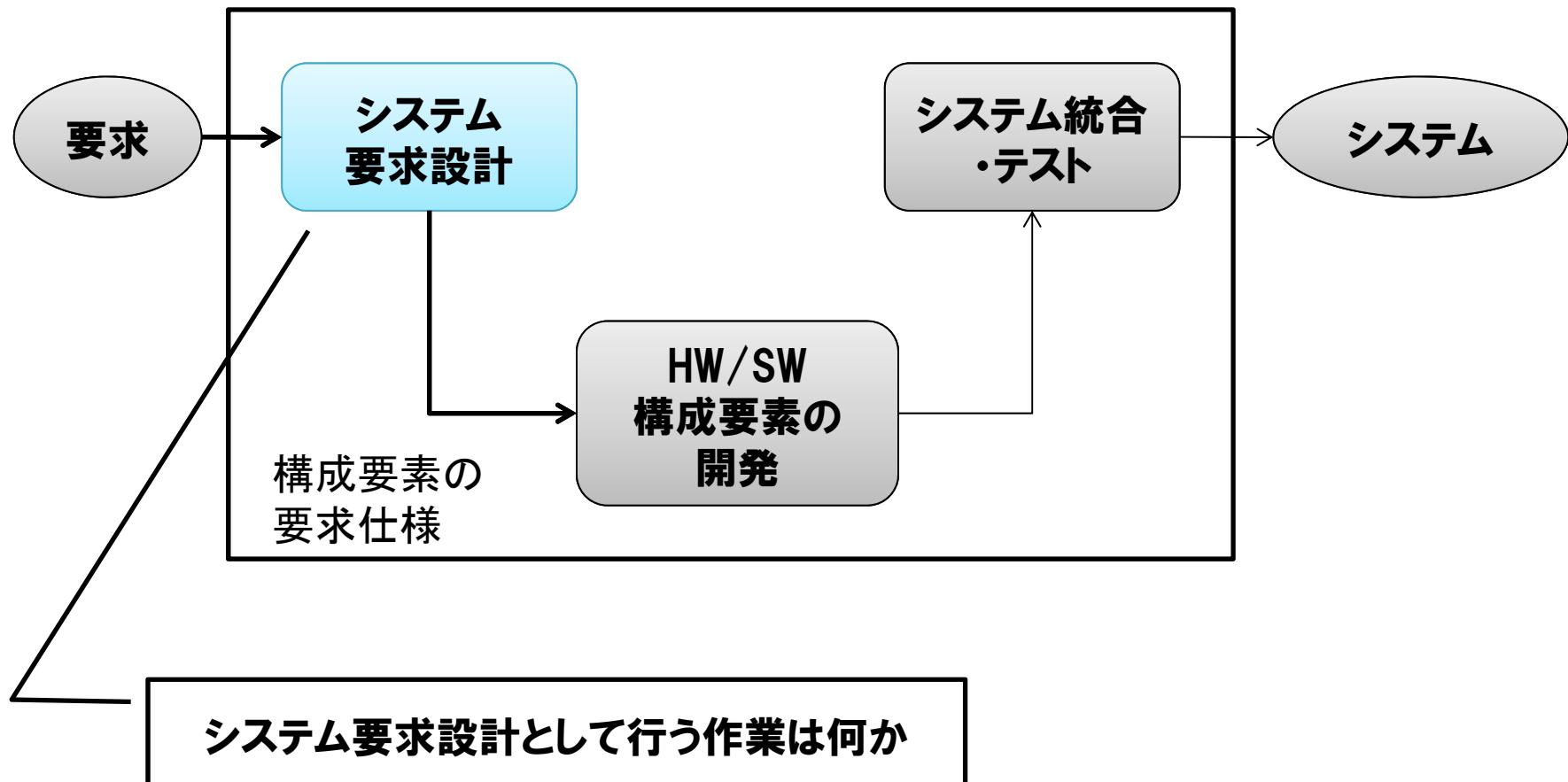
意図したものが実現できる要求定義手順



一つの解決策: システム要求設計



システム開発プロセス



システム要求設計の作業概要



手順	作業項目	表記法		
		SysML	VDM	SLP
要求分析	要求を獲得する	○		○
	システムとその境界を決める	○		
	システムの使われ方（機能）を定める	○		
	ユースケースの動作を表現する	○	○	
アーキテクチャ設計	システムを構成要素に分解する	○		
	部品の相互作用を定義する	○		
	部品の相互接続を定義する	○		
制約評価	システム特性に関する制約を獲得する	○		
	性能等を評価し、アーキテクチャを修正	○		
要求割当て	構成要素の要求仕様を定める	○		
	要求の追跡性を確立する	○		

SysML:システムモデリング言語、VDM:形式手法の一つ、SLP:要求仕様記述ツール



電気ポットを題材として、意図したものを記述する

仕様記述実験

電気ポットを題材として



電気ポットに関する商品企画部門からの要求

要求ID	要求事項
PR01	電気ポットの容量は2リットルとし、10度Cから沸騰するまでの時間は、15分以内とする。
PR02	電源コンセントをつなぐと、直ちに作動し、ヒータで加熱を始め、沸騰に達したら、90度Cに保温する。
PR03	再沸騰ボタンが押されたら、再沸騰を始める。
PR04	水が加えられ、温度が低下したら、再沸騰を始める。
PR05	保温中であれば、お湯を注ぐことができる。



この要求を基に、システム要求設計を行う

要求を獲得する:単一要求への分解



(:PR01:電気ポットの容量は2リットルとし、10度Cから沸騰するまでの時間は、15分以内とする。

Do<容量>を{2リットル}せよ

Do<10度から沸騰するまでの時間>を{15分以内}せよ

(:PR02:電源コンセントをつなぐと、直ちに作動し、ヒータで加熱を始め、沸騰に達したら、90度Cに保温する。

if <電気ポット>が{電源を接続された状態}

 Fn[作動] (1)

 Fn[加熱] (2)

else

 Do nothing

endif

if <電気ポット>が{沸騰状態}

 Fn[保温] (3)

else

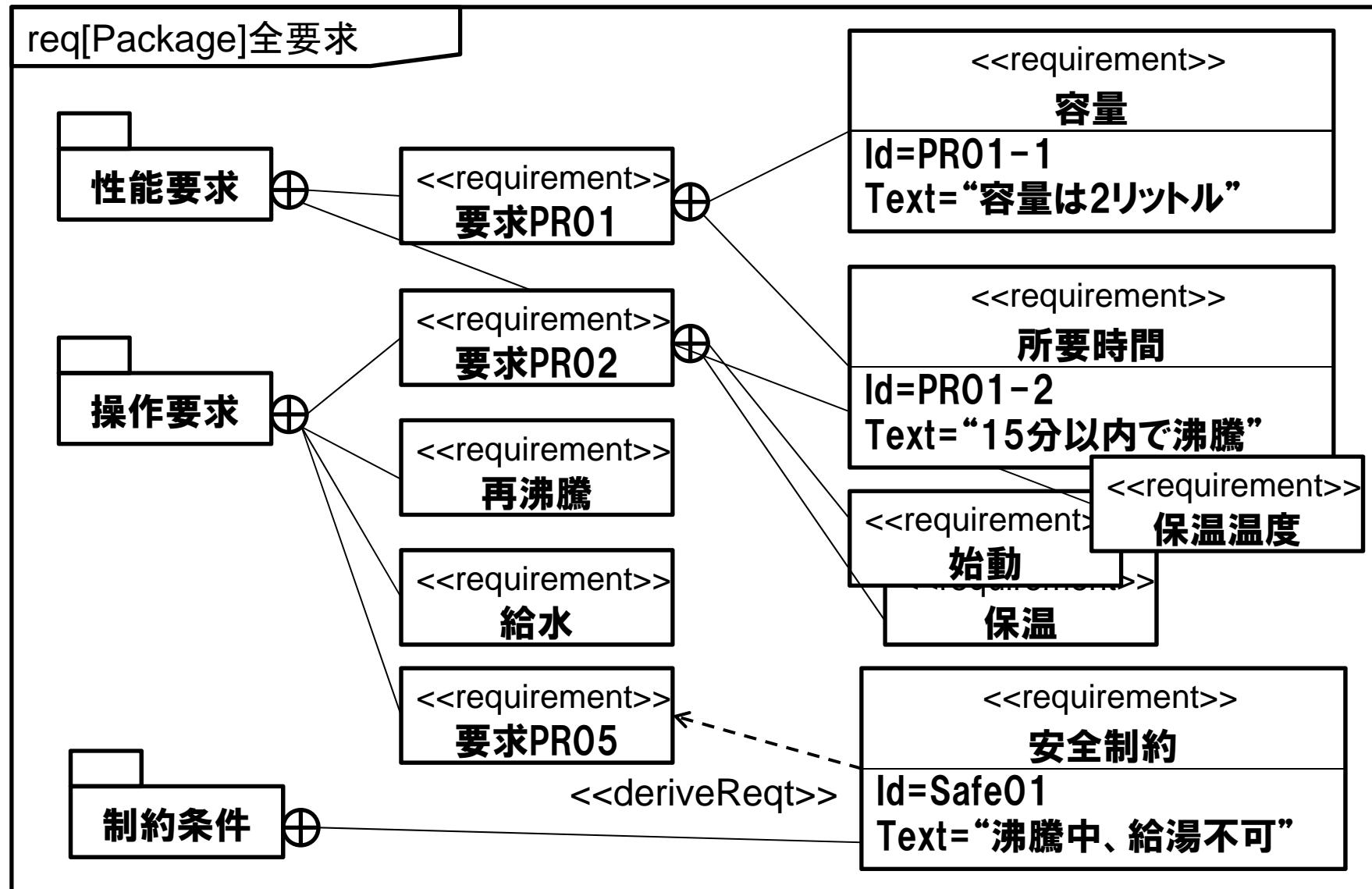
 Do nothing

endif

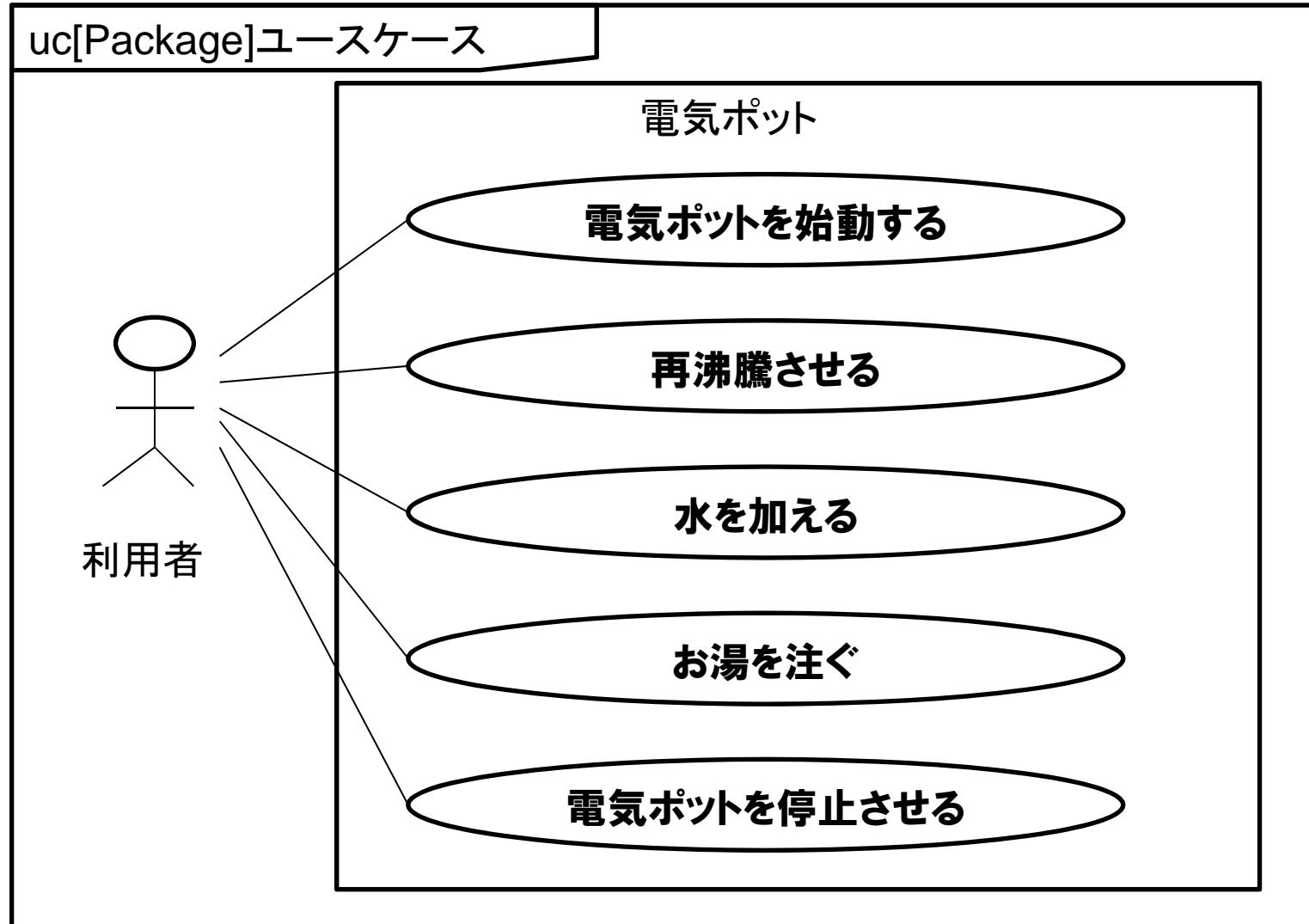
Do<保温温度>を{90度}せよ

(:以下、省略

要求を獲得する:要求の整理

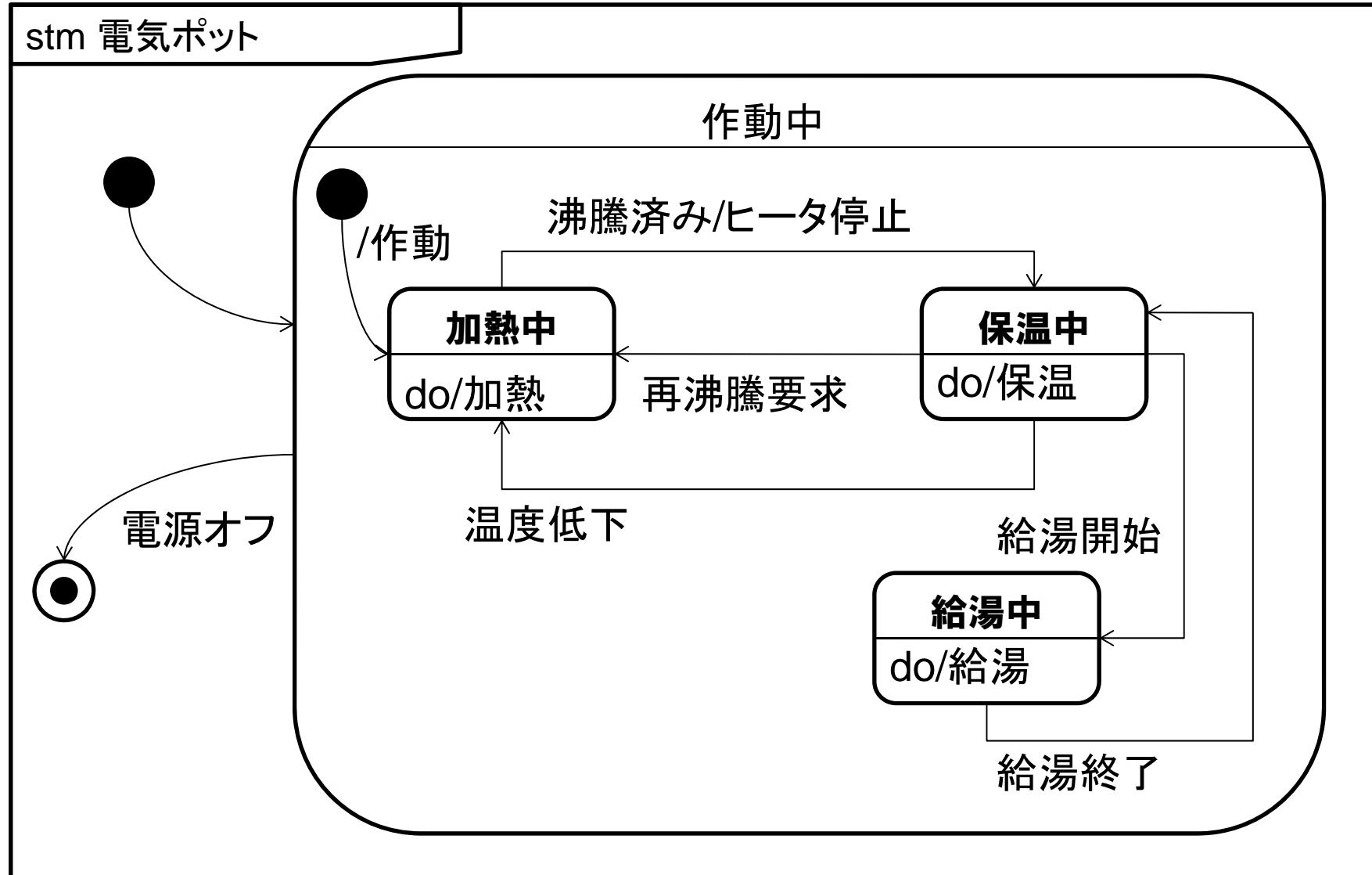


システムの使われ方(機能)を定める

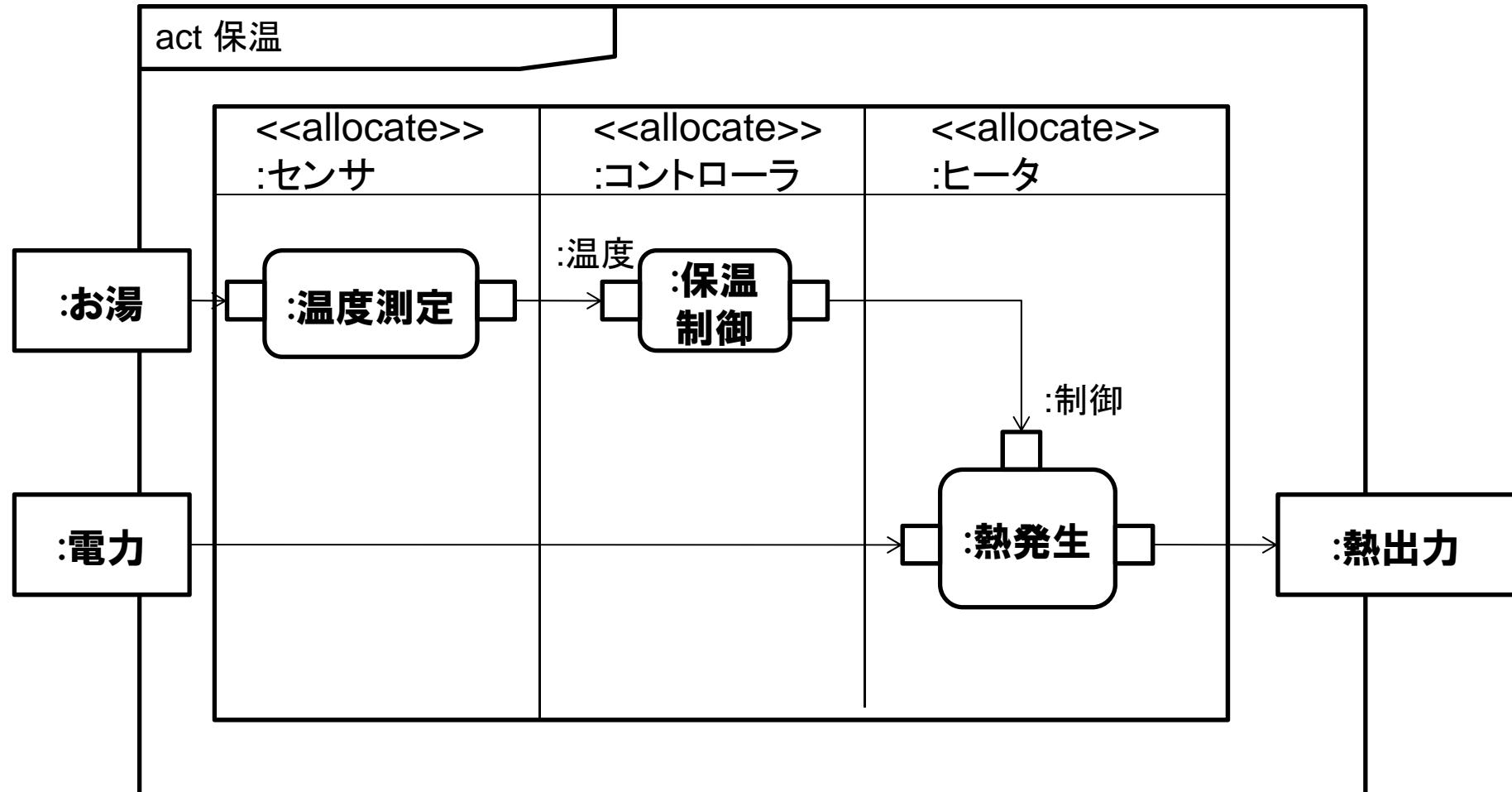


備考:操作に関する要求を基にしてユースケースを洗い出す。

ユースケースの動作を表現する



部品の相互作用を定義する



備考:処理要素であるアクションを部品に割当てる。



目指す方向と今後の予定

要求の仕様化プロセス



手順	作業項目		表記法とスキル			
	発注者	受託者	自然言語	非形式手法	半形式手法	形式手法
1	要求を記述		W	W		
2		要求分析	R&W	R	W	W
3		仕様書作成	W		W	W
4	仕様を確認		R		R	R
5	要求仕様を合意					

スキルの表記は、W: 書く能力、R: 読む能力

今後の予定



- 手法とツールに関する調査継続
 - ・ モデルベース開発等
- プロセスや、手法、ツールの実験継続
 - ・ 題材として、電気ポットを使用
 - ・ 情報セキュリティやセーフティに関する要求を追加
- 要求の仕様化を支援するプロセスや、手法、ツールの推奨案の検討
- 会員を対象とするアンケートの検討



- ISO 26262規格書、ISO
- 要求工学知識体系 (REBOK)、JISA
REBOK企画WG編、近代科学社
- 構造化分析とシステム仕様、Tom
DeMarco著、日経BP出版センター
- A Practical Guide to SysML、
S.Friedenthal他著、MK/OMG Press
- プログラム仕様記述論、荒木啓二郎他著、
オーム社

まとめ



- 安全性向上委員会はセーフティ、セキュリティに品質を追加したSSQを新たな活動テーマとした。
- その一環で、要求の仕様化に関する要求事項と課題、要求の仕様化を支援するプロセスや、手法、ツールを明らかにすることを目標とし、要求の仕様化に取り組んだ。
- 要求の特性を実現するためには、自然言語だけではなく、半形式手法等を用いて要求を記述する必要がある。
- 要求の仕様化は不可逆的なプロセスであり、意図したもののが再現できるように工夫する必要がある。
- 一つの解決策として、システム要求設計というプロセスを検討し、それを用いて仕様記述実験を試みた。
- 今後は、仕様記述実験を継続し、要求の仕様化プロセス推奨案、それを支援する手法、ツールを明確にしてゆきたい。

ご清聴ありがとうございました



要求の仕様化に関する“要求仕様”と手法

2013/5/22 発行

発行者 一般社団法人 組込みシステム技術協会
東京都中央区日本橋浜町1丁目8-1
TEL: 03(5821)7973 FAX: 03(5821)0444
URL: <http://www.jasa.or.jp>

本書の著作権は一般社団法人組込みシステム技術協会(以下、JASA)が有します。
JASAの許可無く、本書の複製、再配布、譲渡、展示はできません。
また本書の改変、翻案、翻訳の権利はJASAが占有します。
その他、JASAが定めた著作権規程に準じます。