



平成30年度 トップエスイーシンポジウム ポスター発表会

トップエスイーコース ソフトウェア開発実践演習

アドバンス・トップエスイーコース プロフェッショナルスタディ

2019年3月22日





トップエスイーコース
ソフトウェア開発実践演習

GRACE



ユースケース図をもとに詳細設計する

キヤノン株式会社
 ライフマティックス株式会社
 株式会社 東証システムサービス

森谷 郁文 moriya.ikufumi@mail.canon
 紺野 雅晃 konno@lifematics.co.jp
 久保 栄治 e-kubo@tssx.co.jp

開発における問題点

システム開発の現場では、**概念モデルやユースケース図／記述**をもとに、体系的、網羅的には**分析・設計を実施していないことが多い**。また、**設計の正しさを検証していないことが多い**。その結果、実装から設計への手戻りが発生し、ソフトウェア品質の低下をまねている。

手法・ツールの適用による解決

今回の演習では、現金残高管理システムを対象として、UMLを用いて、**概念モデルやユースケース図／記述の作成から設計までの一連の分析・設計作業を行った**。そして、**検証ツールであるLTSAを用いて、設計の正しさを検証した**。設計の段階で不具合を検出、解消することで、ソフトウェア品質の低下を抑える効果を確認した。

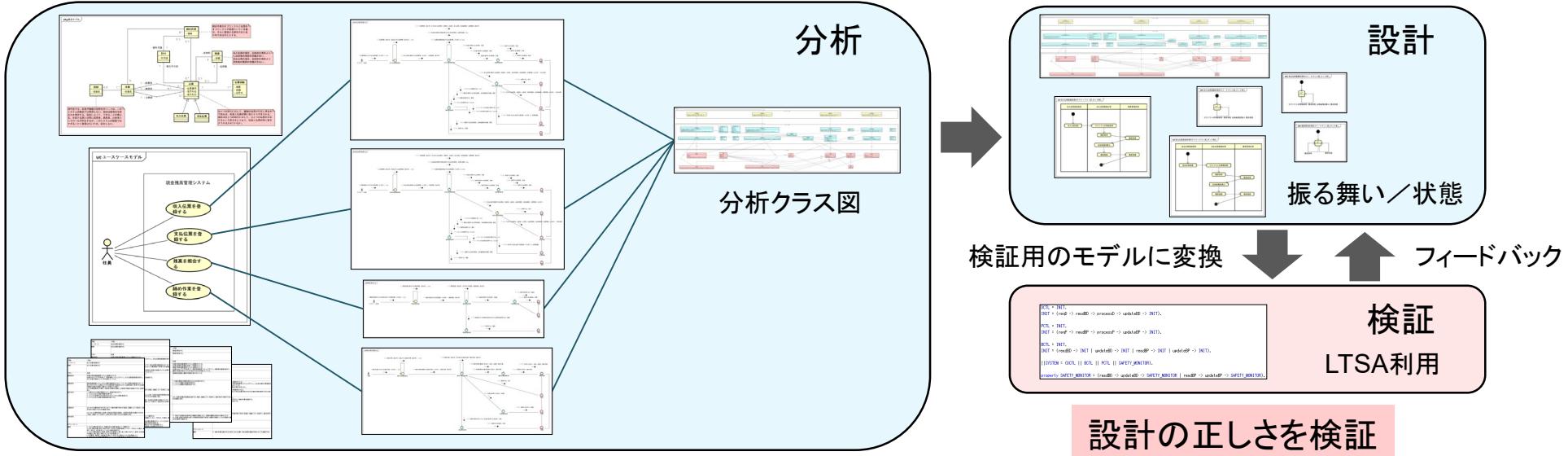
アプローチ

概念モデル
ユースケース図／記述

ロバストネス分析

分析・設計作業をシームレスに実施

設計クラス図



検証

考察

初期のモデル



Trace to property violation in SAFETY_MONITOR

不具合検出

見直し後のモデル



ロックの処理追加

No deadlocks/errors

不具合解消

(参考) 演習課題の工数

	分析・設計	検証
対応工数	12h×3人=36h	15h×3人=45h

検証は設計の1.25倍

- 確認できた効果
 - 設計段階から不具合がありそうな箇所の確認, 客観的な検出, 解消
 - 設計の正しさを説明できる
- 現場適用の課題
 - コスト 人材不足, ラーニングコスト
 - 適用の難しさ 開発支援ツール
- 現場への導入に関する検討
 - コストを下げ, 導入の敷居を下げる
- 手法に慣れた後, 現実の開発に適用
 - 検証にかかる対応工数と, 手戻りによる工数や障害発生時の工数等を比較して, 適用箇所を決める

機械学習システムの要求

～なぜ機械学習プロジェクトはPoCで終わるのか～

荒川 純也, 石谷 規彦, 坂本 竜太, 定塚 和久, 長柄 昌浩

機械学習プロジェクトの問題点

機械学習プロジェクトは下記に示す特性から、一般的に従来のソフトウェア開発と比べて難易度が高いと云われている。

- ・従来プロジェクトに比べて不確実性が高い
- ・データ品質による影響度が高い
- ・顧客側の期待値が高い&理解度が低い

問題解決に向けたアプローチ

文献や論文より収集した機械学習プロジェクトの事例から、プロジェクトの結果に影響を与える要因を抽出・整理した。

更に機械学習に従事する有識者を対象としたアンケートを実施することにより、PoC段階でこれら要因の確認有無とプロジェクト結果を調査し、成功に導く重要な要件を分析した。

アンケート結果から見えてきた重要な要件と成功に向けた考察

重要な要件	次フェーズに繋がらなかった理由(自由記述抜粋)	ソフトウェア工学		機械学習工学	
		適用可能な手法	有効性	必要性	必要な手法
ビジネス課題/ゴール	技術的可能性の検証が先行し、 ビジネススキームへの展開方向が明確でなかった。	ゴール指向要求分析 (GQM+Strategies)	◎	×	-
	ビジネスユースケースの価値の検討が不十分な状態で始めてしまい、最終的に実証したいことにズレが生じてしまった。	ビジネスモデルキャンパス	○	×	-
運用後のコスト/体制	PoCを実施することで 開発費用がかさみ 、提供段階で顧客の想定を超える金額となった。	見積手法	○	△	機械学習の不確実性を考慮した見積手法
	PoCを実施した結果、システム実装時の リソース確保が難しい ことが判明した。	ユースケース分析 見積手法 (デルファイ法等)	◎	×	-
データ収集コスト/品質	データが別部署にある などクライアント側の制約が原因でPoC自体の課題設定が小さくなってしまった。	ステークホルダ分析	△	○	データ観点到特化したステークホルダ分析
	PDCAサイクルを回す業務体制になっておらず、 データ収集に対して理解が得られなかった。		△	○	Machine Learning Canvasの利用
	正解データを準備するなど 顧客が作業負担を大きく感じた。	見積手法	△	○	データ収集負荷を考慮した見積手法

アンケート作成ポイント

アンケート作成時に考慮した分析ポイントと質問内容

<分析ポイント>

1. PoC実施状況
2. ロール別, 立場別
3. 要件の重要度
4. 次に繋がらなかった要因
5. 要件が明確になった時期

質問項目	質問内容
回答者の経験	機械学習プロジェクト経験
	PoC実施経験
	PoC実施結果
PoCについて	PoC実施期間
	PoC実施体制
	回答者の役割・立場
	PoCの目的
要件が明確になった時期	ビジネス観点 機械学習観点
PoCの結果	結果に影響を与えた要素 次へ繋がる重要な要素

活動成果

- ・機械学習プロジェクトの実施事例収集と調査
- ・プロジェクト結果に影響を与える要因を抽出
- ・PoCに関するアンケートを実施することで、文献や事例には載っていない“生の声”を収集
- ・文献調査結果とアンケート結果を比較し、調査結果の妥当性を検証
- ・機械学習プロジェクトを成功に導く3つの“重要な要件”を抽出し、適用可能な手法を考察

機械学習適用に向けた 要求分析・要求合意プロセスの提案

NECソリューション・イノベータ株式会社 岩崎 聖 キヤノン株式会社 飯田 利彦
 NTTテクノクロス株式会社 中村 紘爾 富士通株式会社 高落 要

機械学習を用いたシステム提案の課題

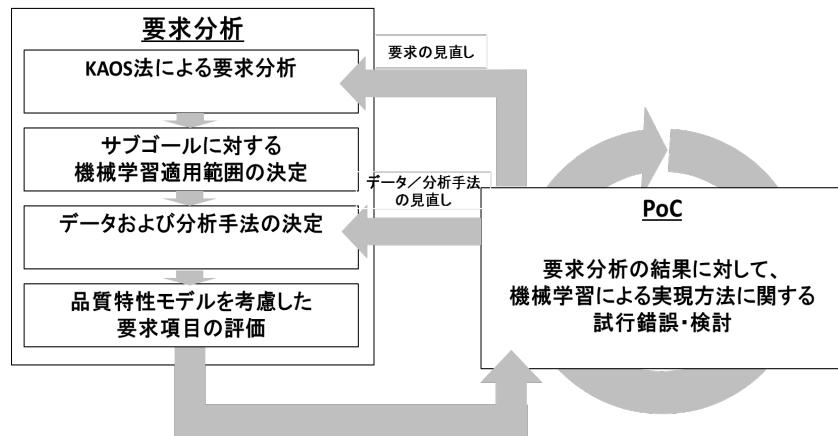
機械学習の技術の急速な発展により、活用するアイデアがあれば、機械学習をシステムに導入することができるようになった。機械学習を導入したシステムは難しい課題の解決が期待できる反面、通常のシステムとは異なる特有のリスクやコストがあることが知られており、経験のないものが導入を検討することは難しい。

要求分析・要求合意プロセスの提案

機械学習により解決可能な課題の抽出、導入可否を判断するためのリスク評価、システム構築までに必要な顧客との合意形成のプロセスを提案する。提案した手法に従って進めることによって、経験のないものであっても、機械学習を導入したシステムの検討を行うことができる。

提案する要求分析・要求合意プロセス

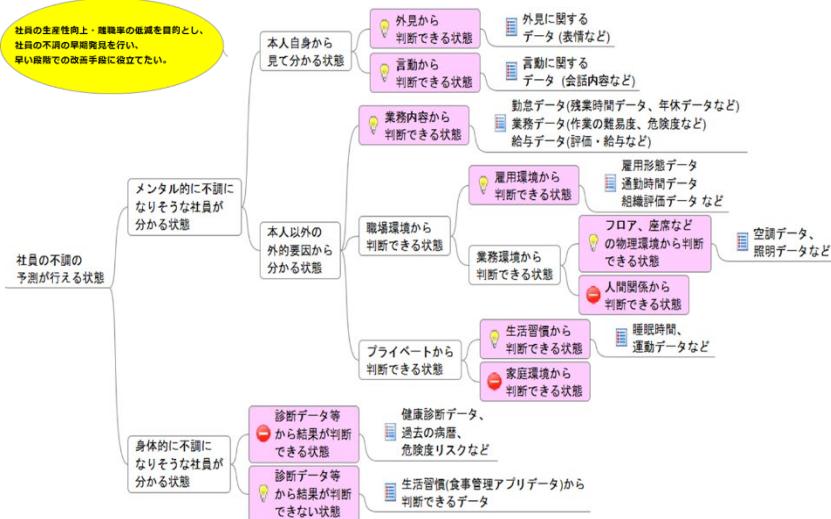
- KAOS法による要求分析:** KAOS法で達成方法を検討可能なサブゴールを抽出
- 機械学習適用範囲の決定:** 抽出したサブゴールに対して、**機械学習適用範囲**を決定する
- データおよび分析手法の決定:** 学習・適用に使用するデータとコストを評価する
- 品質特性モデルを考慮した要求項目の評価:** 顧客との合意状況をもとに、プロジェクトが品質特性を満たすかどうかを**システムの品質評価モデル**で評価する
- PoCによる検討の継続:** 4要素全ての評価がAになるシステムを得られるまで検討を継続



機械学習適用範囲の決定

■機械学習の適用範囲

・サブゴールを満たすための**「既知のデータ」**が存在し、**「未知の結果(推論※)」**が導き出せかどうか。



システムの品質評価モデル

要素	評価モデルの要素	品質副特性
1	各選択肢に十分な例題がある。	正確性
2	出力が不正解な場合でも、システム全体で対応可能	相互運用性
3	データが機械処理容易な形式	合目的性
4	定義した選択肢に対して、網羅異性を確認可能	

評価	定義
A	各選択肢に十分な例題がある。
B	出力が正確でない場合、利用者を含めたシステム全体で対応可能
C	データが機械処理容易な形式
D	定義した選択肢に対して網羅性を確認可能

4要素を4段階で評価
 ↓
全てAになるまで議論、分析を続ける

機械学習応用システムの要求 ～自動運転をテーマとして～

株式会社エクスマーション 前田佑希子

日本電気株式会社 岩本賢芳
 日本電気株式会社 伊藤賢人

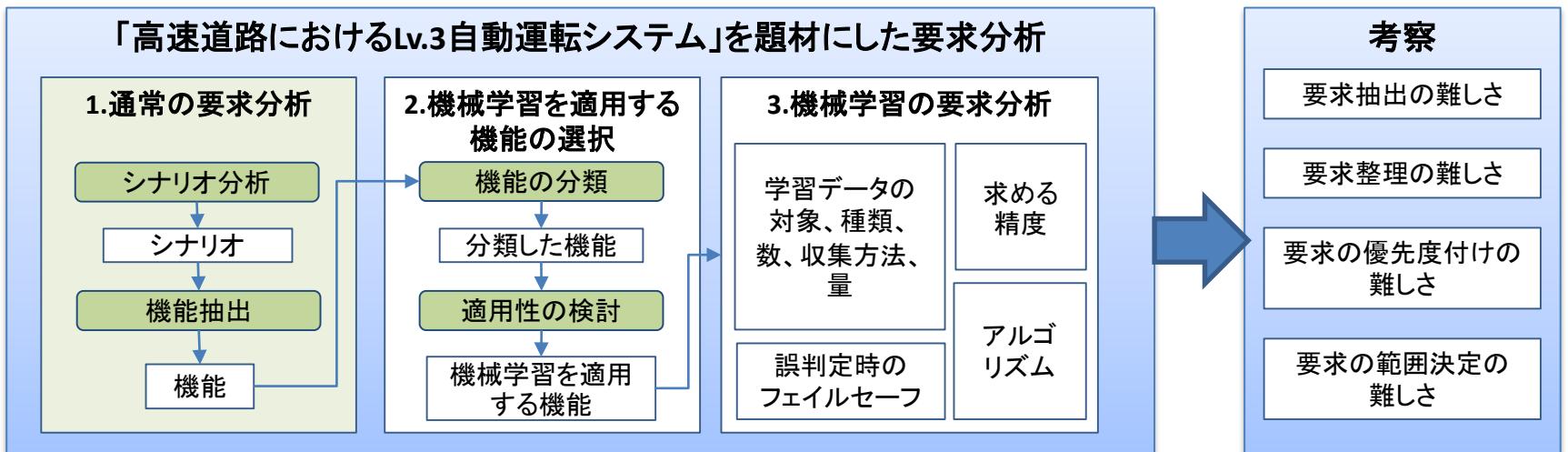
要求分析における問題点

機械学習を組み込んだソフトウェアシステム(機械学習応用システム)を構築する際、何を学習させるのか、どの程度の精度を求めるのか、どのように訓練データを収集するのかなど、機械学習応用システム特有の要求を分析する必要がある。

手法・ツールの適用による解決

高速道路での自動運転に必要な機能を抽出するため、シナリオ分析を実施。そこから必要機能を抽出し、認識・判断・制御の3分類にカテゴリ化した上で機械学習の適用性の検討を実施。中でも機械学習の適用効果が高いと思われる機能を選定し、具体的な要求の導出を実施。

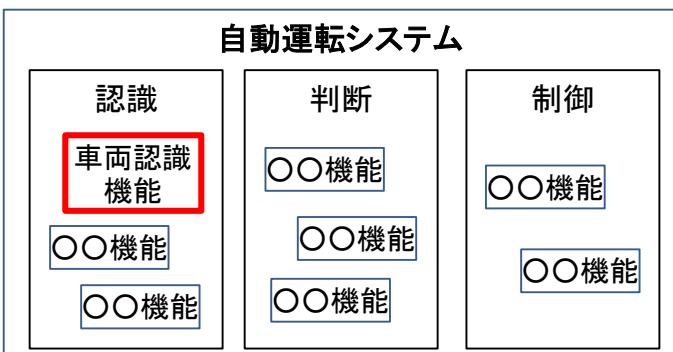
要求分析プロセス



結果

以下の機能は機械学習の適用に向かないと考え除外。
 ・機械学習以外の手法の方が精度が得られると想定される機能
 ・上位機能の出力結果に従うだけとなる「制御系」機能
 ・センサ+計算で算出する方が精度が得られるか、そもそも適する機械学習アルゴリズムが存在しない「判断系」機能
 ⇒車両認識の機能に絞って最終的な要求導出を行った。

要求導出では、「学習対象、データ種、精度、機械学習の種別、データ量、データ収集方法、フェイルセーフ」について検討・考察した。



考察

- 【要求抽出の難しさ】
 - ・データ量や精度の設定根拠が不明。
 - どれほどの影響度がある場合に、どれだけの精度が必要か？ある精度を出すためには、どれだけのデータ量が必要か？フェイルセーフとの絡みで精度を下げてでも許容されるか？
- 【要求整理の難しさ】
 - ・機能の粒度を揃えるのが難しい(機械学習を適用する粒度も難しい)。
 - 今回はできるだけ細分化したが、画像認識→操舵、の研究もある。
- 【要求の範囲決定の難しさ】
 - ・システム丸ごとを範囲とするか？特定の機能に絞るか？絞る場合にはどの粒度か？
- 【要求の優先度付けの難しさ】
 - ・費用対効果をいかに評価するか？
 - ・利用者、開発者、管理者、運用者など全員が納得する、メーカーが訴訟に勝てる等、非技術面の要素も大きいと考えられる。

Kaggleを参照した機械学習アルゴリズム適用パターンの抽出と評価

NECソリューションイノベータ(株)
 (株)日立製作所
 (株)日立製作所

大内 一哲
 岡留 有哉
 神崎 元

日本電気(株)
 東芝デジタルソリューションズ(株)
 日本電気(株)

土屋 俊雄
 松岡 賢
 晦日 慶太

機械学習適用における課題

近年, データ利活用によるビジネスが拡大している. **データ活用には知識と経験が必要**であるが, これらを持つ機械学習エンジニアは現在不足している.

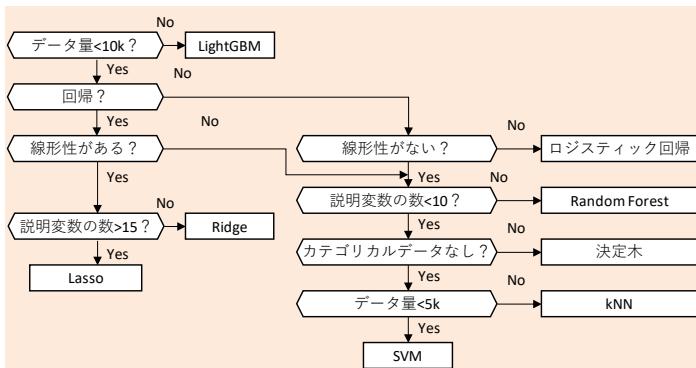
手法・ツールの適用による解決

機械学習によるデータ処理方法をパターン化し, スキルが十分でない人材でも, 適切なアルゴリズムの選択と効率的な実装を可能にする.

機械学習アルゴリズム適用パターン

ソフトウェア実践演習の活動成果物

選択ガイド



適用ガイド

Ridge回帰

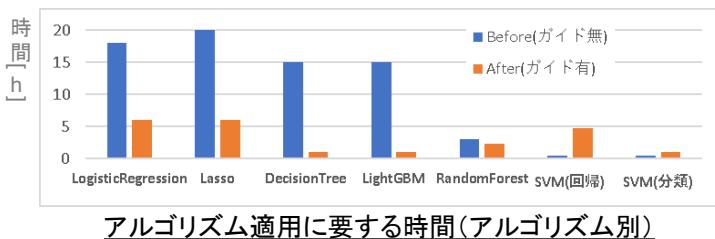
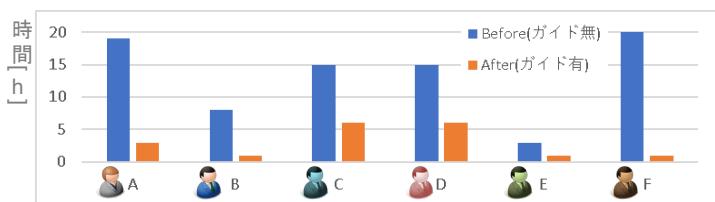
Lasso回帰

- パターンが解こうとしている問題の例
- アルゴリズムが適用可能な条件
- アルゴリズムを適用する手順
- 実装上の注意点
- サンプルコードと適用結果

実装・評価

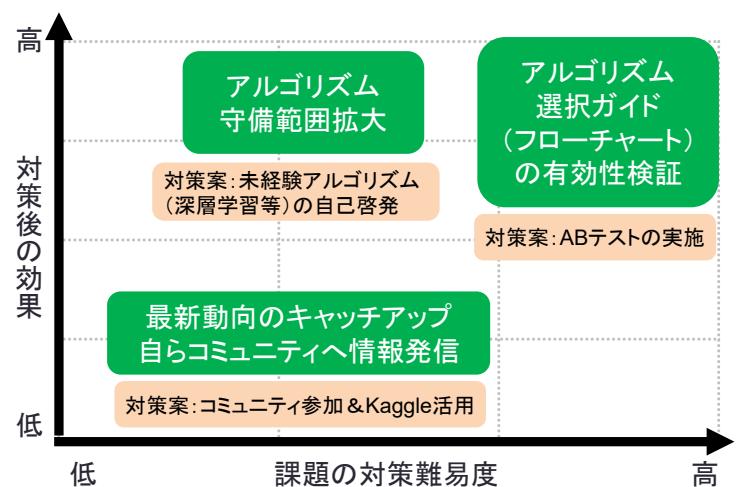
適用ガイドの有効性検証

アルゴリズム適用に要する時間が
62h(77.5%) 短縮!



今後の課題

アルゴリズム選択ガイドの有効性検証と
 IN(理論習得)/OUT(情報発信)の継続



トップエスイー ソフトウェア開発実践演習

Black Jack AIの作成

富士通株式会社
 日本電気株式会社
 日本電子計算株式会社
 三菱電機マイコン機器ソフトウェア

菅原久嗣
 川澄明裕
 森大雅
 太田貴之

hisa.sugawara@jp.fujitsu.com
 a-kawasumi@cd.jp.nec.com
 hiromasa_mori@cm.jip.co.jp
 ohta.takayuki@mms.co.jp

開発における問題点

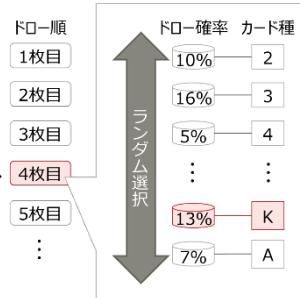
- ・Black Jackフレームワーク上で動作する機械学習モデルによるAIの作成
- ・機械学習モデルの組み立て、与えられたデータの分析手法について知見を得る
- ・メンバー各人でAIを作成しコンペを実施、勝率を競う

手法・ツールの適用による解決

- ・不均一なデッキを使用したブラックジャックを実行するJavaプログラムに対し、プレイヤーの意思決定を行うプログラムを実装する
- ・機械学習ライブラリとしてWekaを使用した
- ・メンバー4人が各自個別に課題に取り組み、最終的に、それぞれが実装したプログラムを用いてコンペを実施した

疑似デッキを用いた強化学習(菅原)

強化学習(Monte Carlo法)を採用
 ドロー順によってドローカード種の確率
 が変動することに着目。
 訓練データから疑似デッキを作成し、
 精度の高いゲーム再現環境を用意。
 2000万ゲーム分の強化学習でAIモデル
 を作成。



勝敗、BUST予測2本立て(森)

基本的な戦略は固定で、その判断材料として機械学習によるデータ分析を用いることにした。学習モデルは2種類用意し、まずカードを引いた場合において勝負に直結するBUST判定を行い、大丈夫そうであれば現状の手札での勝敗判定を行いより勝ちに近い行動を行うように設定した。

手札からの勝敗予測(川澄)

取得できる情報であるディーラーの1枚目の手札とプレイヤーの手札の合計から勝ち負けを予測できるようにWekaで決定木による機械学習を実施。
 予測が勝ちの場合はHITし、負けの予測の場合、STANDする戦略とした。

訓練データAIの改善(太田)

訓練用データ作成時のAI(手札の合計値が17未満の場合HIT、17以上の場合STAND)を訓練データを機械学習することで、模倣、改善。



結果・考察

手法	勝率
疑似デッキを用いた強化学習(菅原)	44%
勝敗、BUST予測2本立て(森)	38%
手札からの勝敗予測(川澄)	23%
訓練データAIの改善(太田)	20%

- ①結果が最もよかった手法はデッキに対してアプローチし、訓練データをうまく学習用データに加工できたことだと考えられる。
- ②テーマに合わせた学習方法の選択が重要
- ③今回は対象の仕組みが明快だったが、現場利用の際はそうとは限らない。実際は仮説検証を繰り返し、モデル構築に苦労すると思った。

欲求×感情×行動の相互作用の 定量化と活用方法の検討

株式会社日立製作所
 株式会社クレスコ

鎌田 雄大
 木内 一揮

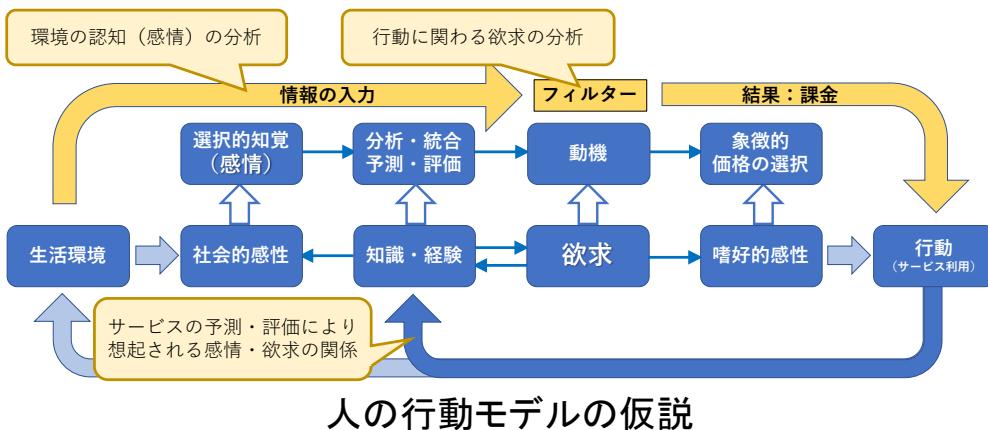
開発における問題点

ソーシャルメディアやプラットフォームビジネスの発達により、業界の垣根を越えたサービスが創出されるなか、消費者の精神性がビジネスの成否に与える影響が増している。一方、ヒトの感情や欲求といった内面的な側面を定量化することは困難であり、投資や開発のリスク要因を増大させている。

手法・ツールの適用による解決

欲求と感情がヒトの行動に与える影響をモデル化し、既存のアプリをベースに欲求・感情の影響を分析・定量化する手法を検討した。分析結果に基づいて影響を可視化・予測するツールとして利用することで、ビジネスの企画立案や成果予測を可能にすることを旨とする。

仮説の立案と分析手法の検討



スマホアプリの主要機能の抽出

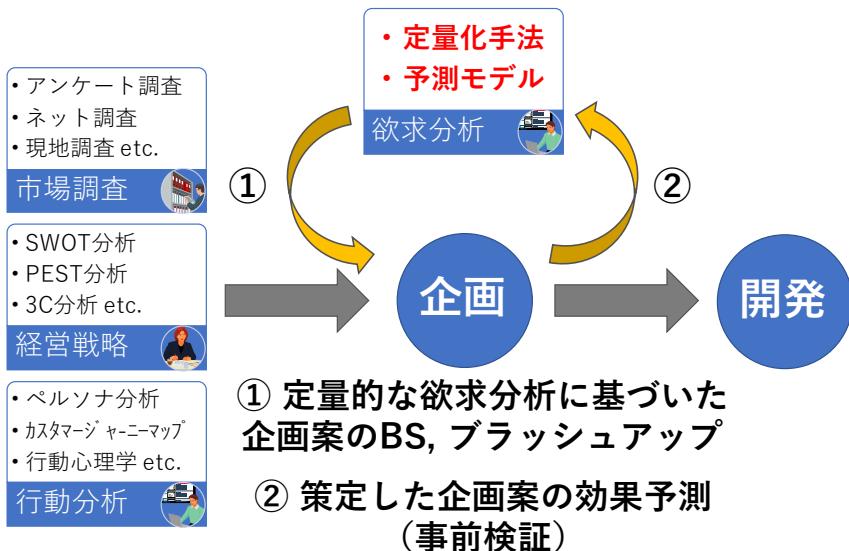
主要機能使用時の心理分析
 (マレーの欲求リストとプルティックの感情の輪)

各アプリとビジネス指標の関係調査
 (実態調査)

心理分析結果とビジネス指標の回帰分析
 (欲求・感情の影響度の定量化)

適用範囲の検討

新企画の立案・調査、開発項目の精査時の活用を想定



分析結果

一例としてモバイルアプリの課金率を分析したところ、影響の大きい感情と欲求の組み合わせとして図中の組み合わせが抽出された。



テレビCMが与える消費者の購買行動への影響度分析

NTTテクノクロス株式会社
 株式会社東芝

奥野 拓也 okuno.takuya@po.ntt-tx.co.jp
 織田 達弘 tatsuhiro.oda@toshiba.co.jp

TVCMにおける問題点

テレビCMは視覚・聴覚に訴えられ**消費者に与える影響が高い**といわれている。また広告主である企業は購入意欲や売り上げを高める際に**テレビCMを重視**しており多数の企業が高額な投資が行われている。しかしテレビCMは単一方向なメディアであるため**その効果がどの程度なのかを知る方法は難しい**。

手法・ツールの適用による解決

テレビCMの広告効果を検証する。一般的に広告効果を売上や利益の増加などの指標とした場合には、売上は商品力・販促・流通競合等の要因が複雑に影響するため純粋な広告効果を測定することは難しい。そこで数理モデルを援用による分析を行った。

アプローチ

広告マネジメントのプロセス

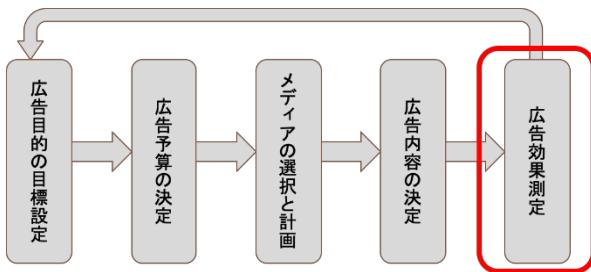
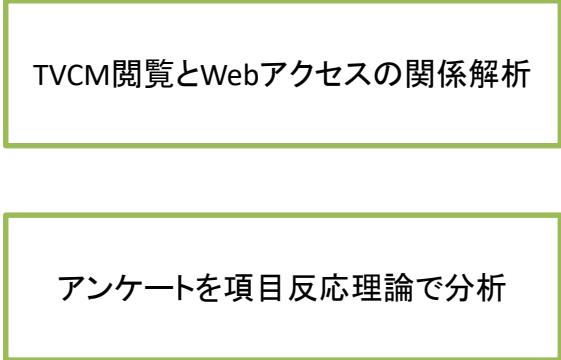


表1 提供されたデータとそのサイズ。

項目	レコード数
アンケートデータ	約 280 万行
TV 接触ログ	約 180 万行
TV 再生ログ	約 480 万行
WEB サイト閲覧ログ	約 4750 万行
CM 時点データ	約 150 万行
番組情報	約 120 万行
標本情報	約 190 万行
局マスタ	約 7 行
職業マスタ	約 16 行
番組分類マスタ	約 119 行
広告主マスタ	約 2700 行
銘柄マスタ	約 6700 行
アンケートマスタ	約 4700 行



アンケートを項目反応理論で分析

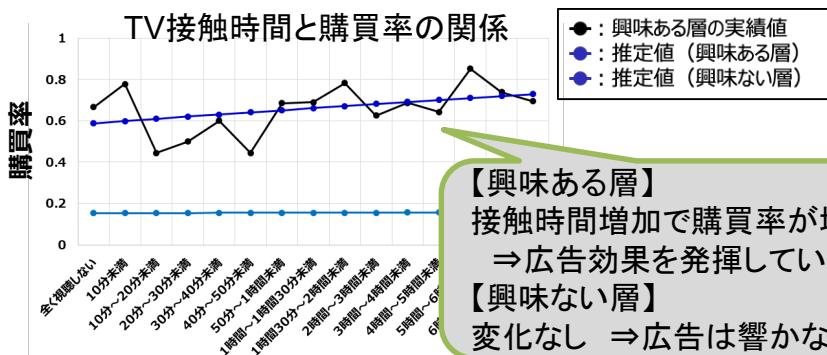
モデル (製品Aを購買する確率P)

$$P(x_{ik} = 1 | \theta_i) = c_j + \frac{1 - c_j}{1 + \exp[-1.7a_k(\theta_i - b_k)]}$$

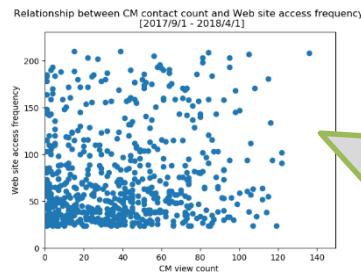
正答率82%
 で実績を再現

a_k : 識別力(≒好意, ブランド力, 広告の完成度)
 b_k : 困難度(≒価格等の買いやすさ, 購買頻度)
 c_j : 当て推量(≒偶然購買する確率)

結果



TVCM閲覧とWebアクセスの関係解析



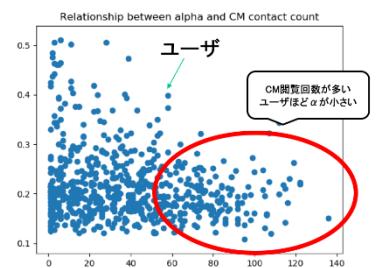
TVCMの閲覧数とWebサイトアクセス数には相関がない

モデル

$$\lambda^*(t) = \mu + \sum_{t_i < t} \varphi(t - t_i)$$

$$\varphi(t - t_i) = \sum_{i < j} \alpha \beta \exp(-\beta(t - t_i))$$

結果



MicroOpsCI with LC4RI ~ Notebook手順書にCIを適用する ~

富士通(株) 阿部 秀一 富士通(株) 井浦 陽一郎 富士通(株) 宇野 耕平
 (株)クレスコ 原野 昌幸 (株)日立製作所 山崎 航史

ITインフラの運用における問題点

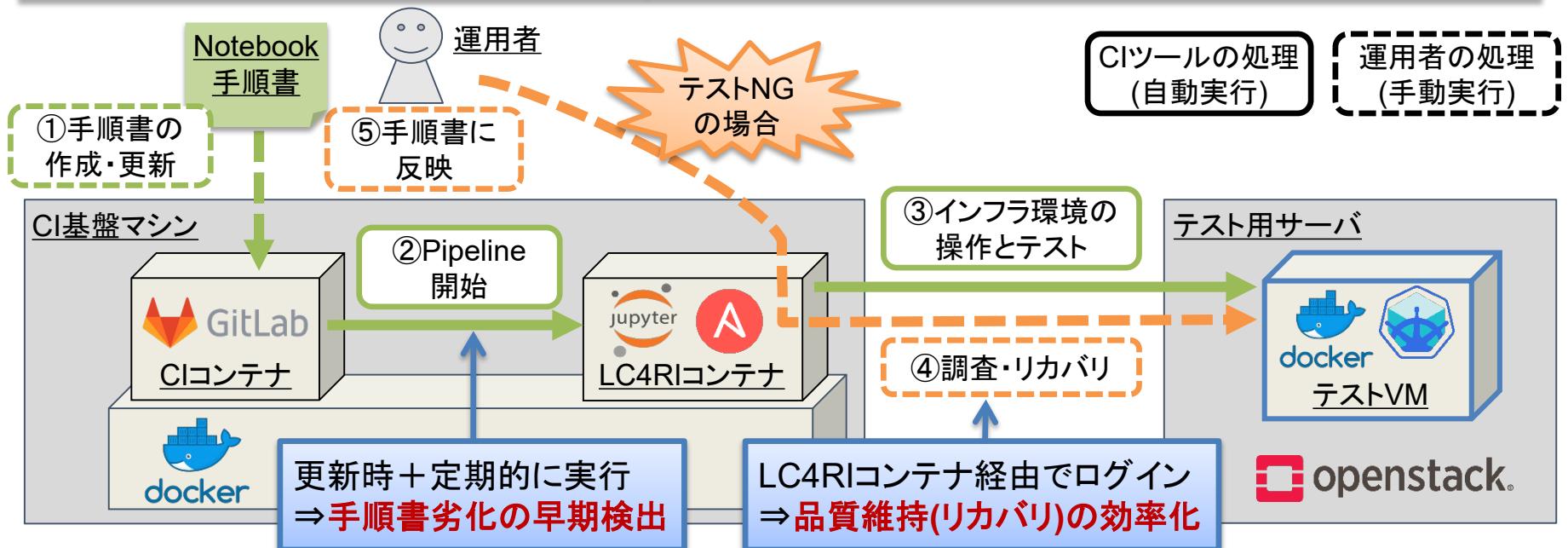
ITインフラの複雑化により、Infrastructure as Codeなどの自動化が取り組まれている。しかし、単純な自動化は、保守性が低く、属人化が課題である。また、インフラの周辺環境は頻りにアップデートされるため、自動化コードの品質を維持することは困難であり、エラー発生時の調査とリカバリには高いスキルと多くの工数が必要とされる。

手法・ツールの適用による解決

仕様書とコードを一元管理するLC4RI*に対して、小さな(Micro)単位の操作(Ops)をテスト対象とする継続的インテグレーション(CI)手法を適用した。運用者は、保守性に優れた実行可能な手順書を作成し、その品質を効率的に検証・維持することが可能である。

[*] <https://literate-computing.github.io>

構築したMicroOpsCI with LC4RIのテスト環境

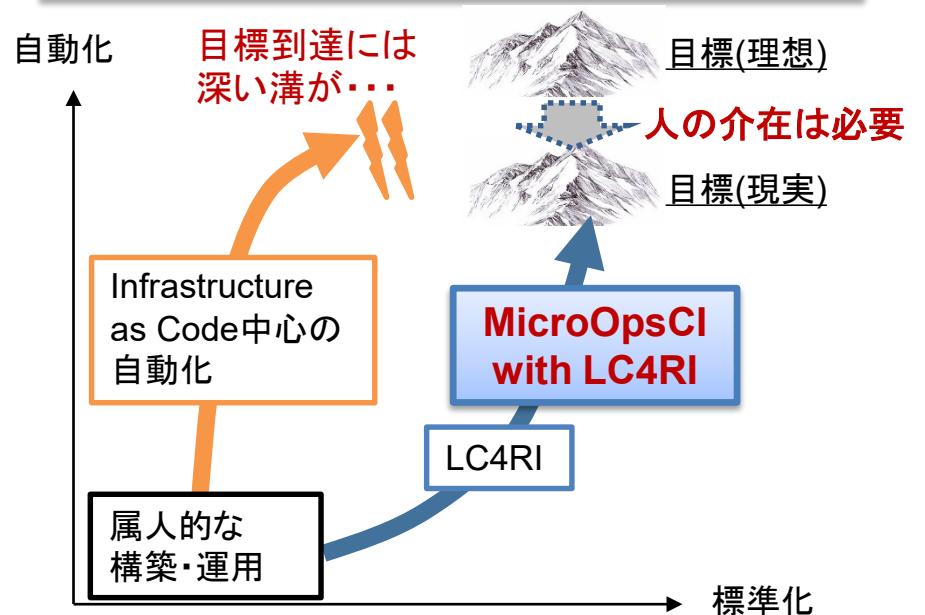


Notebook手順書

仕様記述 (Markdown)	1.2.1 Dokcerのインストール 2018/11/25現在、CentOS版の最新版 18.09 のDockerにKubeadmが対応していないため、Minikubeの起動に失敗する。今回は動作確認済みである 18.06.1-ce をインストールする。 参考： https://github.com/kubernetes/minikube/issues/3323
コード実行	<pre>ansible-playbook -l { target } playbook/docker.yml</pre>
実行結果	<pre>PLAY [minikube*] ***** TASK [Gathering Facts] ***** ok: [157.1.141.96]</pre>
テスト実行	1.2.2 Dockerの確認 Dockerのバージョンが 18.06.1-ce であることを確認
テスト結果	<pre>ansible { target } --become -a 'docker version'</pre> <pre>157.1.141.96 SUCCESS rc=0 >> Client: Version: 18.06.1-ce</pre>

仕様・コード・実行結果を一元管理 ⇒ **保守性の向上**

提案手法の位置付け



トップエスイー ソフトウェア開発実践演習

ソルバー活用演習～Alloyを用いたAgile Specificationの提案～

キヤノン株式会社
 富士通株式会社
 キヤノン株式会社
 NECソリューションイノベータ株式会社

菊地明美
 木村昇一
 小林正季
 高岸正人

kikuchi.akemi@mail.canon
 sho.kimura@jp.fujitsu.com
 kobayashi.masaki007@mail.canon
 m-takagishi@pb.jp.nec.com

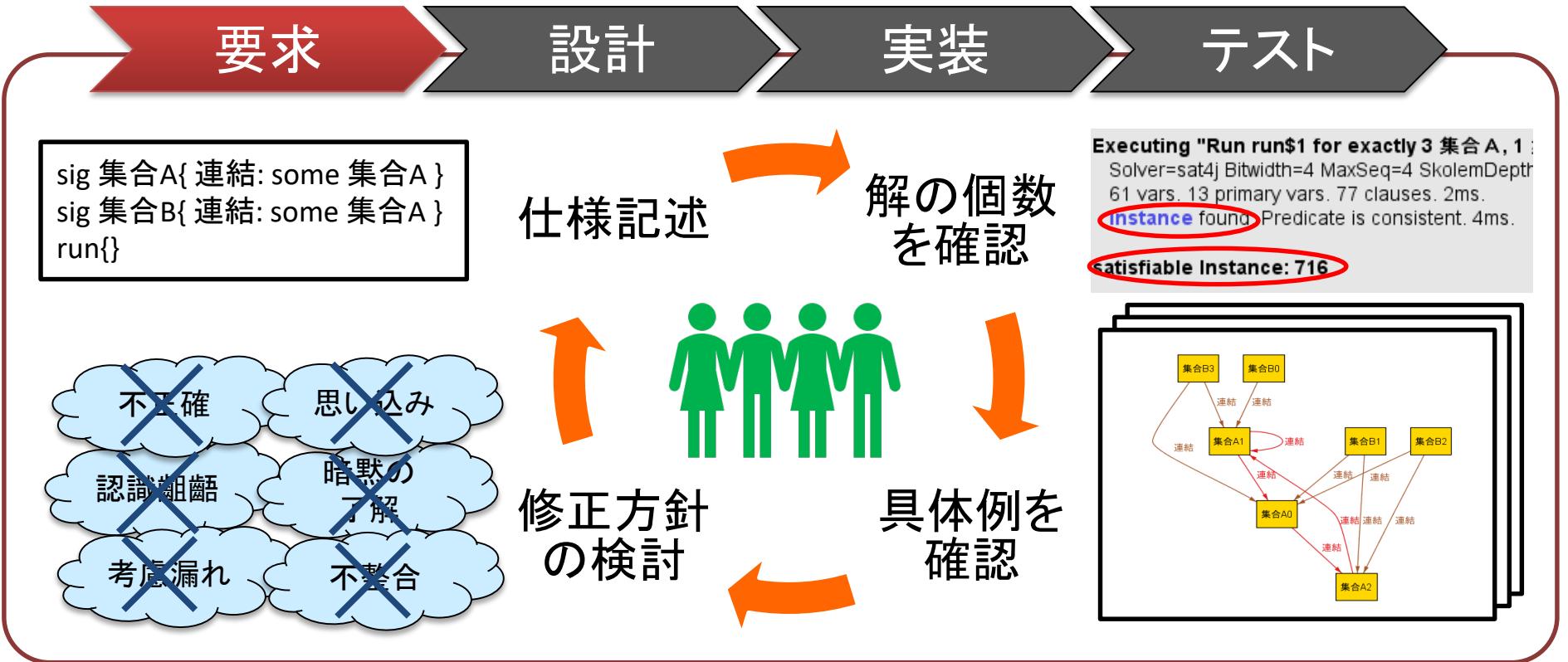
開発における問題点

ソフトウェア開発の上流工程に形式仕様記述を適用すると、自然言語による曖昧さが排除され、上流品質の向上と開発トータルコストの削減が期待できる。しかし、上流工程工数増加の問題と、プログラム動作確認開始までのリードタイム遅延リスクがある。

形式仕様記述の新プロセス提案

軽量形式仕様言語Alloyとアジャイル的手法を組み合わせた新しい形式仕様記述プロセスを提案した。
 要求分析を小さな記述から開始し、仕様修正・追加と具体例確認を繰り返す新プロセスにより、上流工程の品質確保とコスト削減を実現し、具体例による動作確認の前倒しを可能とした。

Alloyを用いたAgile Specification



特徴

アジャイル的手法を取り入れた形式仕様記述

イテレーション

小さく始めて段階的にすばやく追加・修正

モブプログラミング

具体例を図や表の形で即座に確認・議論

効果

上流コスト削減と要求仕様品質向上の両立

今後の展望

- ソルバー検査機能のさらなる活用
- 適用事例の拡大と有効性の検証
- 他手法との定量的な比較

実践ATDD / TDD

株式会社NTTデータ 加藤 史也
 株式会社 日立製作所 相樂 恭宏
 日本電気株式会社 梅原 峻

テストにおける問題点

ソフトウェア開発において、JUnitやSelenium等のテストフレームワークによるテストの自動実行は一般的な取り組みとなっている。しかし、本演習の参加メンバーが所属する現場では、設計とテストの間で整合性がとれていないケースや、チームメンバーがテストの意義を理解せず、テストが形骸化してしまうといった問題が発生している。また、テストの活動において、前述のような問題は広く発生しうると考える。

手法・ツールの検証

テストの作成を設計に準ずる活動として位置付ける手法: Acceptance Test Driven Development (ATDD)およびTest Driven Development (TDD)の適用によって前述の問題の解決に繋がるかを考察する。ATDD, TDDを実際に適用したソフトウェア開発を通して、これらの手法によるメリット、デメリットを検証する。また、得た知見をもとに、実際の業務に適用するにあたっての留意点について考察する。

演習の実施内容

開発対象のアプリケーション

本演習では、TODOリストのWebアプリケーション開発を対象として、ATDD, TDDを適用した。今回の演習では4種類の機能(TODOの登録, 一覧表示, 完了, 期限が近いものの強調表示)を開発対象として設定し、実装を行った。

演習のフローと所感

本演習で実施した開発のフローを図1に示す。開発の初めに受入テストシナリオを作成し、それを用いて顧客と合意形成を行う。その後、対象とする受入テストシナリオを合格するようにプログラムを実装した。受け入れテスト, 単体テストの先行作成は、設計に相当する活動であり、実装するプログラムが設計(=テスト)通りであることを常に確認可能であることがATDD, TDDの利点であると感じた。

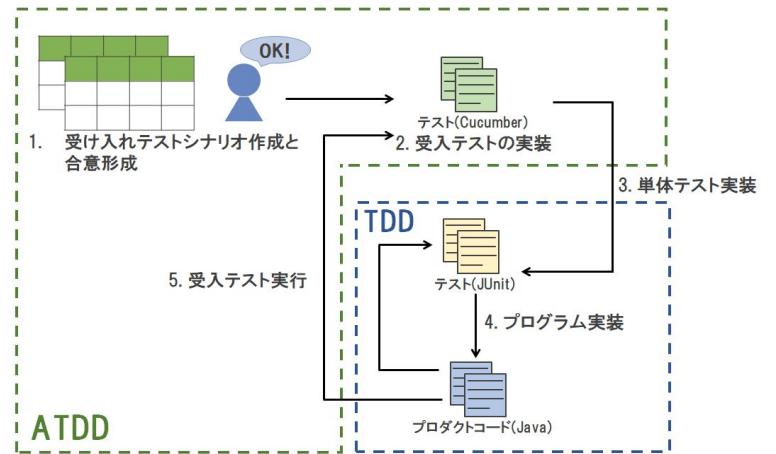


図1. 開発のフロー

期待できる効果

有意義なテストの作成

ウォーターフォール開発では、テストを作成する時点で実装が完了しているため、機能テストが実装を追求するだけの無意味なものになってしまう懸念がある。ATDDでは、テストを作成する時点で実装が存在していないことを保証できるため、機能要件の達成を厳密にチェックするテストを作成することができ、テストの形骸化の防止に役立つと考える。

機能仕様の品質向上

ATDDでは、仕様をテスト項目の集まりとして表現する。ソフトウェアの操作や動作を具体的に検討する必要があるため、仕様の漏れや曖昧な点を早期に発見することができる。

考慮・工夫が必要な点

テスト駆動開発という概念の導入コスト

これから開発するソフトウェアのテストを作成するためには、完成品を実装レベルで想像するという独特な感覚が要求される。さらに、テストフレームワークを利用する場合はその習得も必要であり、チーム内に未経験者が多い場合、モブプログラミングなど導入コストを削減する工夫が必要であると感じた。

ATDD適用の向き不向き

どのような開発にもATDDが適しているとは限らないと考える。例えば、テスト1件ごとの開発規模の見積りが難しいため、担当の割り振りが必要な大規模開発には不向きである。また、非機能要件のテストは、実装が完了していないと一般的に作成が困難である。これらの観点を含め、ATDDの適用可否、適用範囲の考慮が必要である。

事例に基づくシンプルデザイン

相澤 亮太, 青山 祐三, 小牟田 俊介, 中尾 陽一
 廣田 智紀, 深瀬 智紀, 山本 淳一

背景

近年ウォーターフォール開発（以下WF開発）だけではなくアジャイル開発（以下AG開発）の導入が進んでいるが、AG開発に関する理解度が低く、AG開発に期待する成果が十分に得られていないという課題がある。
 主要な原因は、AG開発の経験値不足（人材不足、プロセス未理解）である。

主旨/ゴール

主旨
 ①AG開発のプラクティスを実践することでメリットを体感する。
 ②WF開発とAG開発の適材適所(メリット/デメリット)を理解する。
 上記より、プロジェクトの性質に応じて最適な開発手法を選択する方法論を提案する

アプローチ/結果/特徴

2チームに別れて、同じ例題をWF型とAG型で開発してその成果物を比較分析する。

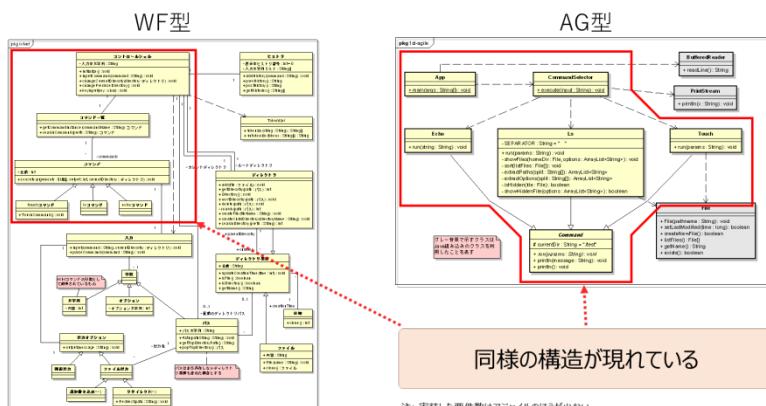


図1: 詳細設計レベルのクラス図比較

演習結果より、各開発手法の特徴に関する様々な見解を得た。

- ・ウォーターフォール開発
 - 全体像が見えているためスケジュールや見積りが立て易い
 - 意見が発散しやすく合意形成が図り難い
- ・アジャイル開発
 - プロジェクトの後期段階のスケジュールが不明確であり、全体的なスケジュールを立て難い
 - 小規模の機能に集中することにより十分な検討ができ、合意形成が図り易い
- ・共通
 - 品質は開発プロセスよりもプラクティスや技術者の熟練度の影響が大きい
 - 開発プロセスに関わらず、ペアワークは有効なプラクティス

考察

以上の演習結果より、開発手法（ウォーターフォール開発/アジャイル開発）は、システム開発の規模/期間に合わせて使い分けることを提案する。

結論/今後の課題

以上より、下記結論を得た。

- 開発手法は、システムの開発規模と開発期間を基準に決定する。（大規模な長期開発は要件確度も基準とする）
- AG開発の経験値を増加させるために、小規模なシステム開発は全てAG開発で実施することが望ましい。
- AG開発のプラクティスの1つであるペアワークはWF開発でも活用可能。

今後の課題として下記3点が挙げられる。

- 実装する要件数を揃えた上での設計比較の実施
- 成果物の品質を測るためにコード比較の実施
- 規模や期間の大きい開発での追加検証の実施



図2: システム開発と開発手法の関係性

機能制約モデルからのテスト生成手法の提案

(株)日立製作所

齋藤 英美

emi.saito.tj@hitachi.com

テスト設計における問題点

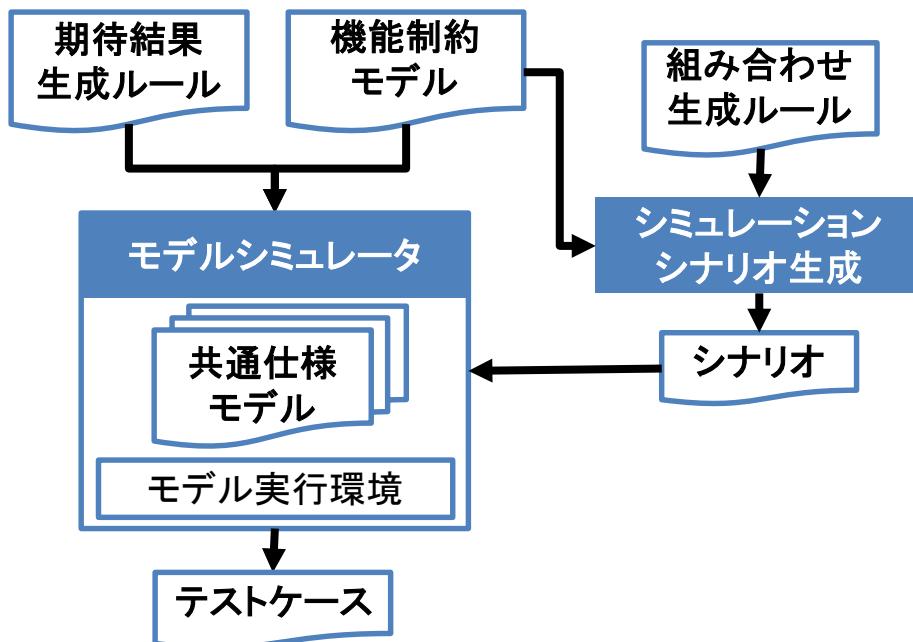
接続機器や機能が多い製品のテスト設計において、以下の問題がある。

- (1)仕様が複雑になり、考慮不足による曖昧な仕様が生じやすい
 - (2)テストが必要な条件(接続機器や機能)の適切な組み合わせ設計が難しい
- 以上により、テスト設計の効率が低下している。

手法・ツールの適用による解決

- (1)の解決:形式手法で仕様を記載し、曖昧さを事前に排除する。
 - (2)の解決:テスト対象機能をモデル化し、別途研究中の組み合わせ生成ルールを組み合わせることで、テストを生成する手法を構築する。
- 上記を組み合わせ、形式手法で記載したテスト対象機能のモデルからテスト生成する手法を実現。

テスト生成手法



- 機能制約モデル (テスト対象機能のモデル)
テスト対象機能の動作範囲と、動作範囲の制約条件(接続機器等)を記載
- 共通仕様モデル
モデルシミュレーションに必要となる、対象製品の共通的な仕様情報を記載
- 組み合わせ生成ルール
制約条件の最適な組み合わせを算出するルール
- 期待結果生成ルール
制約条件の組み合わせから、機能の動作範囲(=テストの期待結果)を算出するルール
- シナリオ
各テストごとの条件組み合わせといった情報をモデルシミュレータが解釈可能な形式で表記したもの

手法の利用例

空調のコントローラに対し手法を利用する場合の、モデルやテストケースの例を以下に示す。

- 機能制約モデルの例
 - ・ 動作範囲: 暖房, 冷房, 送風, ドライ
 - ・ 制約条件: 冷房専用機の接続時, 暖房使用禁止 (実際には手法で規定した形式的な表現で記載)
- テストケースの例
 - ・ 事前条件: 冷房専用機の接続
 - ・ イベント: 運転変更操作
 - ・ 期待結果: 冷房, 送風, ドライが使用可能

評価

- ・ 形式手法で複雑な仕様を表記・検査し曖昧な仕様を複数検出, 曖昧性排除の効果を確認できた
- ・ 形式手法でモデルやモデルシミュレータを表記して手法を構築, 実際に熟練者が設計したのと同じテストケースを生成できた

今後の展望

- ・ 形式手法の知識がなくても利用可能な形式をめざし, モデルの簡易な作成方法を検討する
- ・ 仕様の定義とテスト設計を完全に1プロセス化することで作業の簡略化をめざす

知識バックグラウンドが異なるメンバー間での効率的な問題共有を可能にするコミュニケーションテンプレートの提案

株式会社富士通研究所

菅原茉莉子

sugawara.mariko@fujitsu.com

開発における問題点

背景

・SlackやRedmineなどテキストベースのコミュニケーションが頻繁に行われているがうまくいかないことも多い

課題

・要因の一つとして、質問者側の当該分野における経験不足により、質問自体をうまく行えていないことがあげられる

解決手法の提案

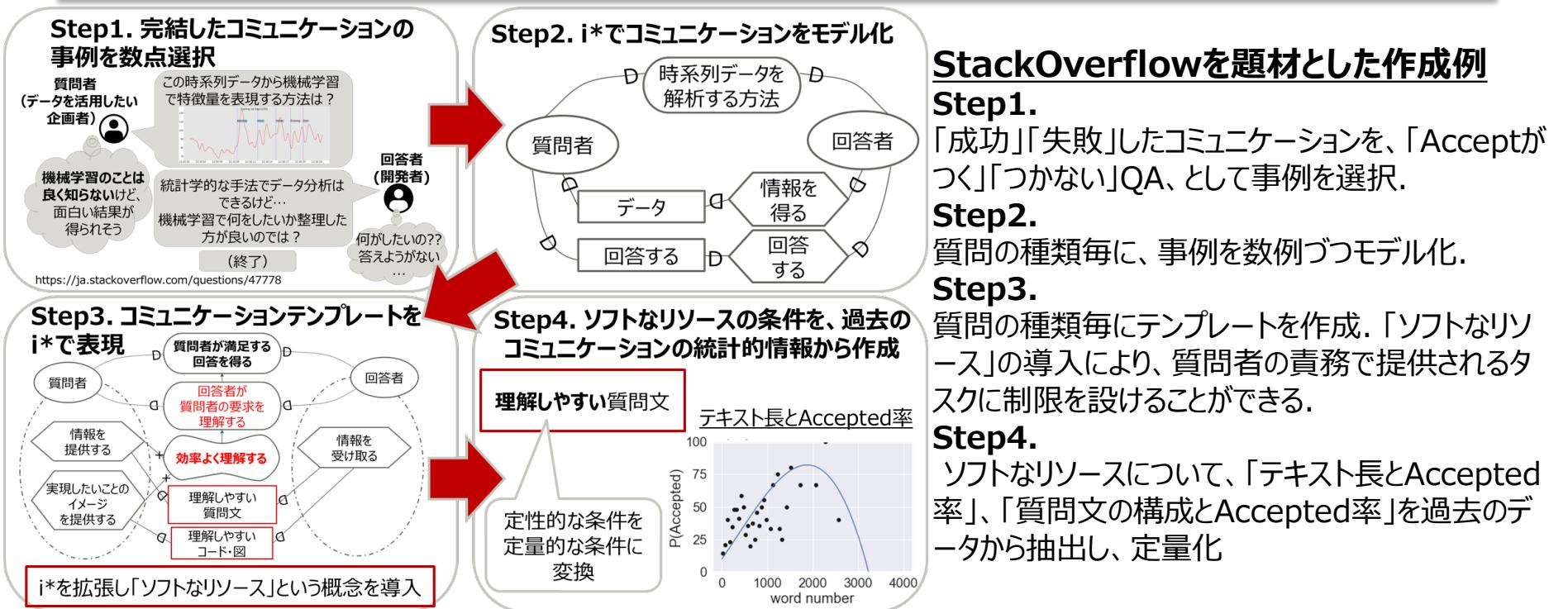
提案するもの

・質問者がうまく質問できるようなコミュニケーションテンプレート及び、その作成プロセス

コミュニケーションのモデル化手法

・テンプレート作成のためのコミュニケーション分析手法として、ゴール指向要求分析手法の一つであるi* (アイスター) を採用。

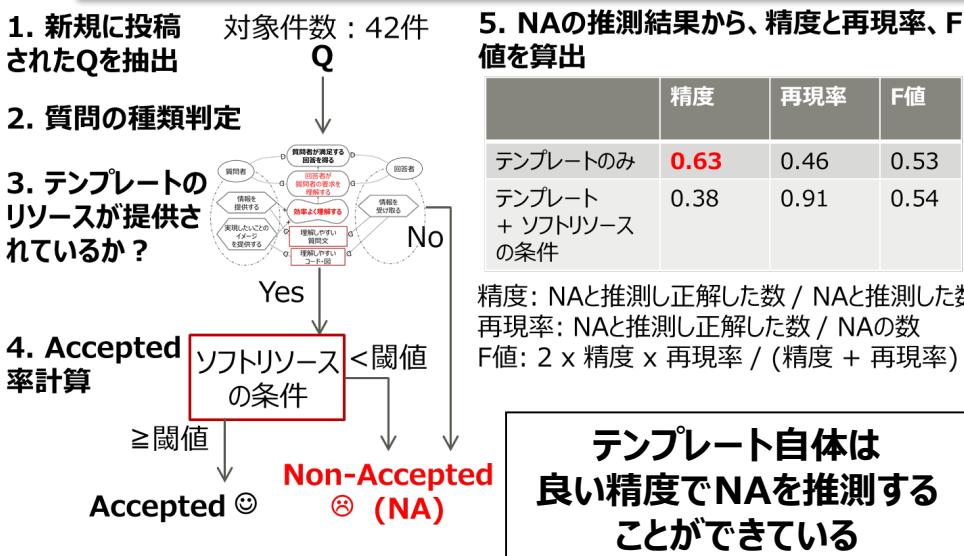
質問者のためのコミュニケーションテンプレート作成手法



StackOverflowを題材とした作成例

- Step1.** 「成功」「失敗」したコミュニケーションを、「Acceptがつく」「つかない」QA、として事例を選択。
- Step2.** 質問の種類毎に、事例を数例ずつモデル化。
- Step3.** 質問の種類毎にテンプレートを作成。「ソフトなリソース」の導入により、質問者の責務で提供されるタスクに制限を設けることができる。
- Step4.** ソフトなリソースについて、「テキスト長とAccepted率」、「質問文の構成とAccepted率」を過去のデータから抽出し、定量化

作成したテンプレートの評価



今後の展開

ソフトリソースの条件について

・Accepted率に寄与するパラメータの抽出が必要。過去のデータを使って機械学習を導入することも考えられる。

StackOverflow以外への適用について

・日本語でのコミュニケーションでも基本的に同じ手法でテンプレートは作成可能と考える。
 ・ソフトなリソースを導入したことで、さまざまなコミュニケーションに柔軟に対応できることを期待。

カメラの新規機能搭載における 仕様検討時の妥当性検証

キヤノン株式会社

角田 昌芳

開発における問題点

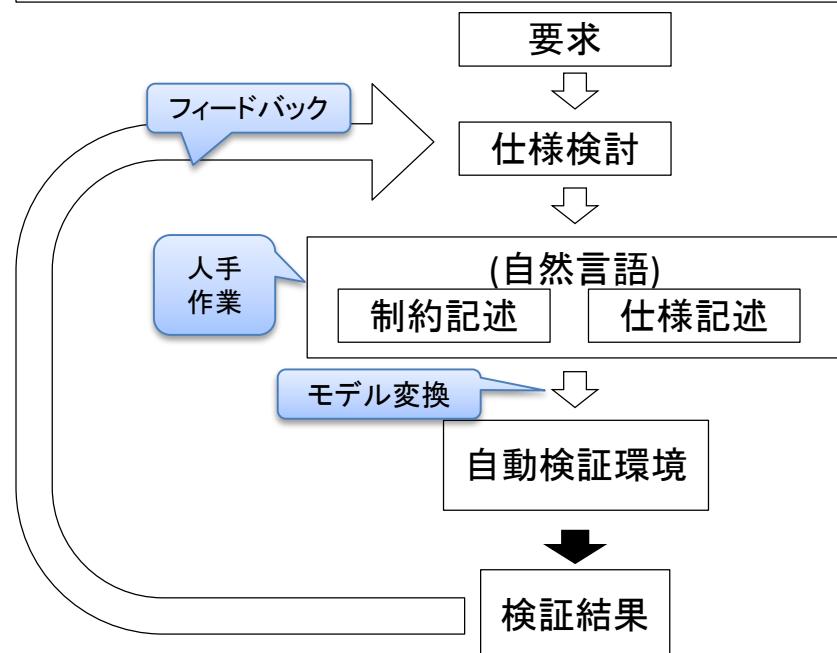
- ・カメラは、マーケティング、ハードウェア、機能特性といったさまざまな観点の制約をクリアした仕様を検討し、多機能の搭載・高性能・良い操作性の実現が求められている。
- ・制約を整理して、正しい仕様が検討できていないと開発の下流工程でバグ・仕様再検討との不要な工数が発生してしまう。

手法・ツールの適用による解決

要求される制約およびカメラの動作状態を整理し、検証ツール上にモデル化する手法を検討する。その後、検証ツール上でモデルおよび制約を実装し、自動的に検証実行することで、仕様検討支援手段を提供する。

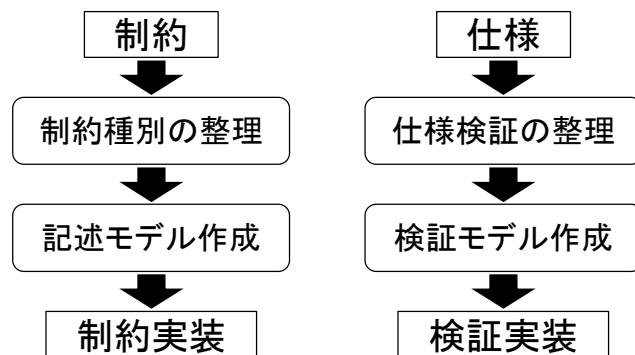
提供する仕様検討支援手段

検討支援手段の作業フロー



本演習における実現内容

- ・自動検証環境の構築
 - 検証環境が保持する機能・性能
 - カメラの制約を記述可能
 - カメラの検証すべき仕様を記述可能
 - 検証結果のフィードバックの可視化
 - 支援するに値する実行時間で実現可能



自動検証環境の評価

- ・制約・仕様矛盾を含んだ仕様を用いて、自動検証環境上に実装し提案モデルの正当性を確認
 - ・仕様実現の可能性を発見可能
 - ・仕様矛盾の発見可能
 - ・小規模な制約・仕様においては高速に検証可能
 - ・独自の知識が必要なため、単体での支援には技術が必要

今後の取り組み

- ・動作時間検証
 - ・問題サイズを現実に近いものの検証する
- ・ユーザビリティの向上
 - ・自然言語を入力とした検証環境の検討
 - ・DSLの検討
 - ・視覚化の改善

コネクティッドカーシステムの構築における DDS^[1]の適用評価

[1]Data Distribution Service

株式会社デンソー 福田 謙児 kenji_u_fukuda@denso.co.jp

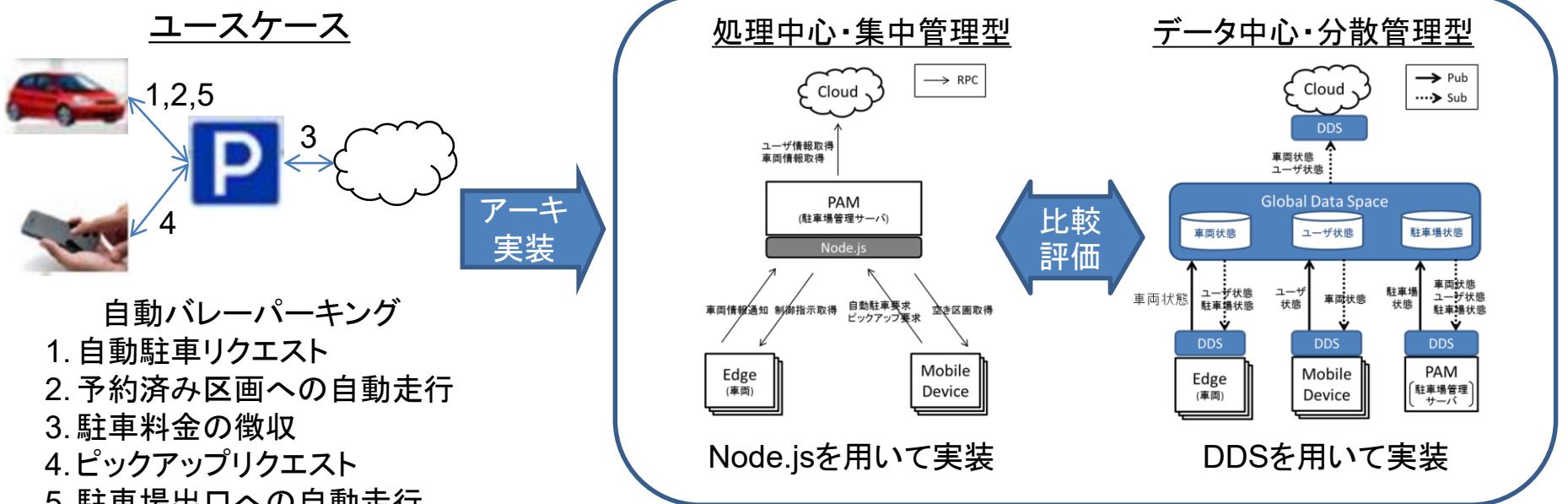
開発における問題点

- コネクティッドカーシステムでは拡張性・可用性・応答性の要求が従来より高度化
- 解決策としてDDS(Data Distribution Service)が自動車業界で注目
- コネクティッドカーのシステム要件に対するDDSの適合度が未検証

手法・ツールの適用による解決

- 特定のユースケースを用いてコネクティッドカーのシステム要件を具体化
- DDS型(データ中心、分散管理)と非DDS型(処理中心、集中管理)のアーキを実装
- コネクティッドカーのシステム要件に対し上記2つのアーキテクチャを比較評価

問題解決のアプローチ



評価

コネクティッドカーのシステム要件を観点として評価

評価観点	非DDS	DDS
拡張性	接続機器の複製による増加に強み	接続機器の種類の変化に強み
可用性	物理構成に応じて再送処理等を設計	物理構成に応じてQoSパラメタを適合
応答性	同期処理発生時の応答時間の悪化に注意	制御周期の設定に注意 [本演習での計測結果] 車-駐車場間の応答時間 ・22 - 67118ms (1ms周期) ・100 - 137ms (100ms周期)

考察

- コネクティッドカーシステムの実現に向けてDDSは機能的には十分
- 一方で、所望の性能を実現するためには設計パラメタの最適値の探索が必要
- 今後は、実開発への適用に向けて、以下の3点で性能評価と課題抽出を継続
 - 現実的なシステム規模
 - 機能の抜き差し
 - 設計支援ツール

FAQ作成支援のためのクラスタ内文書ランキング

株式会社富士通研究所 溝渕裕司 mizobuchi.yuji@jp.Fujitsu.com

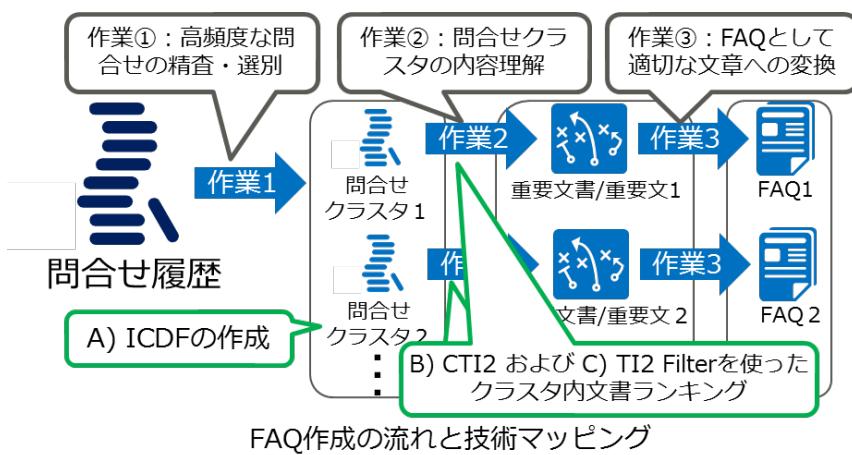
FAQ作成における問題点

- FAQ作成は過去の膨大な問合せ文書からの①精査選別、②内容理解、③文章作成を要し、多大な労力を要す
- なかでも内容理解は、膨大な文書読解が必要で的確な選別が必要である

解決方法

- クラスタ内文書ランキングの提案とその高精度化
 - A) : FAQ化タスクに特化した単語の重み付け方法の開発
 - B,C) : また、それを活用したクラスタ内ランキングの高精度化

FAQ作成の流れと提案手法

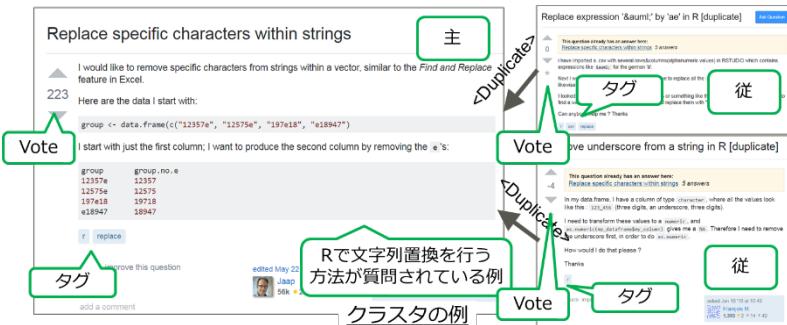


FAQ作成の流れと技術マッピング

- A) 新単語の重みづけ手法(ICDF) **新手法**
 - 文書全体では出現頻度が低く、同一クラスタ内では多くの文書で出現する単語を重要とする方法
 - 通常のランキングで使われるTF-IDFをベースにICDF項を追加して高精度化を図る
- B) 新文書メトリクス手法(CTI2) **応用1**
 - 文書内のICDF値の総和を文書の重要度とする方法
- C) 新特徴量選択手法(TI2 Filter) **応用2**
 - ICDF値を使って単語の選択方法
 - 今回は深層学習の判別モデルに適用

実験・評価方法

- データ : 2018/9/5発行のStack Overflow Dump Dataを活用し以下を抽出
 - 頻出する投稿のクラスタ
 - 3つ以上の文書からなる物で35352個抽出
 - 各クラスタの正解文書順序をVoteから収集



- 評価指標 : Adjusted-NDCG
 - 通常の文書ランキングの評価指標であるNDCG(Normalized Discounted Cumulative Gain)に補正項DCG_worstを加えたもの

結果

- CTI2の実験結果

機械学習モデル	メトリクス	A-NDCG
教師なし	Tags Count	0.563
	Cumulative TF-IDF	0.485
	Gunning Fog Index	0.515
	CTI2	0.563 ← 改善!
教師あり	線形回帰	0.585
	+RM(Readability Metrics)	0.587 ← 改善!
	多層パーセプトロン	0.586
	TM+RM+CTI2	0.598 ← 改善!

- TI2 Filterの実験結果

Case	説明	入力サイズ (単語数)	隠れ層のサイズ	トータルパラメタ	A-NDCG
1	フィルタなし	2032504	20	40650142	0.688
2	TF-IDF-ICDFによる単語選択	60549	50	3027602	0.701
3	ランダムに単語選択	60549	50	3027602	0.554



アドバンス・トップエスイーコース
プロフェッショナルスタディ

GRACE



アジャイル開発に対して、リスクベースドテストを実施するケーススタディ

所属 株式会社クレスコ

古谷恒平

k-furutani@cresco.co.jp

開発における問題点

アジャイル開発等の短納期開発・反復的な少人数チームでの開発の中で「定められた期間の中で」「効率的に品質を向上させる」テストを実施する必要性が高まっているが、どの様な方法で効率的に実施する事で開発の負担にならず効率的なテストが行えるかわかっていない。

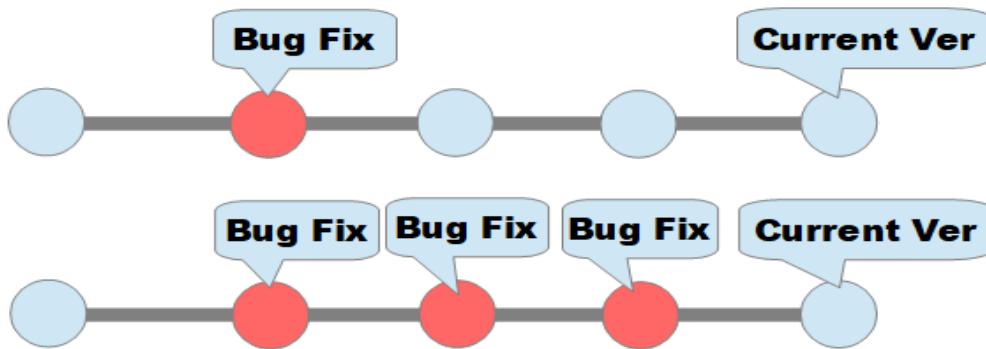
手法・ツールの適用による解決

少人数チームでの開発では従来のリスクベースドテストを運用すると、プロジェクト負担が大きかったため、軽量な3分類と、ソースコード履歴から追えるファイル単位での過去のバグ修正コミット回数の2要素だけを用いて、精度を維持した効率の良いテスト計画が行えることを確認した。

提案手法

以下のルールでテストの実施優先度・実施可否を判断する

1. 修正したファイルの過去のバグコミット数を確認し、バグ数大・中・小に分類する
2. 修正した要件の重要度を、大・中・小に分類する
3. 1をリスク発生確率、2をリスク影響度と判定し、右下のようなマトリクスにマッピングする
4. マトリクスの各マス毎に優先順位の数字を付け、各スプリントの工数が足りる範囲でテストを実施する（例えば、下の図のように①～⑥の順番で数字を付け、工数との相談で①～③まで実施する、等にする）



この場合、下の方がリスクが高いと判断。

		リスク発生確率		
		小	中	大
リスク影響度	小	-	-	④
	中	-	⑥	③
	大	⑤	②	①

評価

実際の小規模なプロジェクトデータを基に、リスクの判定結果が正しく出るかどうかを判定した結果

- ・10%のファイルのテストだけで全体の25%のバグ抽出を確認。
- ・通常のリスクベースドテストのリスク算出手法と比較しても精度が劣らない(今回の結果であれば上回る)事を確認。

	今回の手法で判断したリスク発生分類		
	リスク小	リスク中	リスク大
実際のバグ発生率	16%(208/1295)	36%(36/100)	58%(35/60)

今後

この手法が既存のリスクベースドテストのリスク算出手法と同程度の精度を出せることを確認した。今後は、

- ・ファイルの修正実績が溜まっていない開発初期においても通用する軽量な判断基準の作成
- ・プロジェクトの規模が拡大した場合にも活用しやすい判断基準の拡張
- ・今回の手法以外の軽量な判断基準の提案等を検討していく

スポーツにおけるコーチング支援のための データ活用

日本ユニシス株式会社

太田裕一

yuichi.ohata@unisys.co.jp

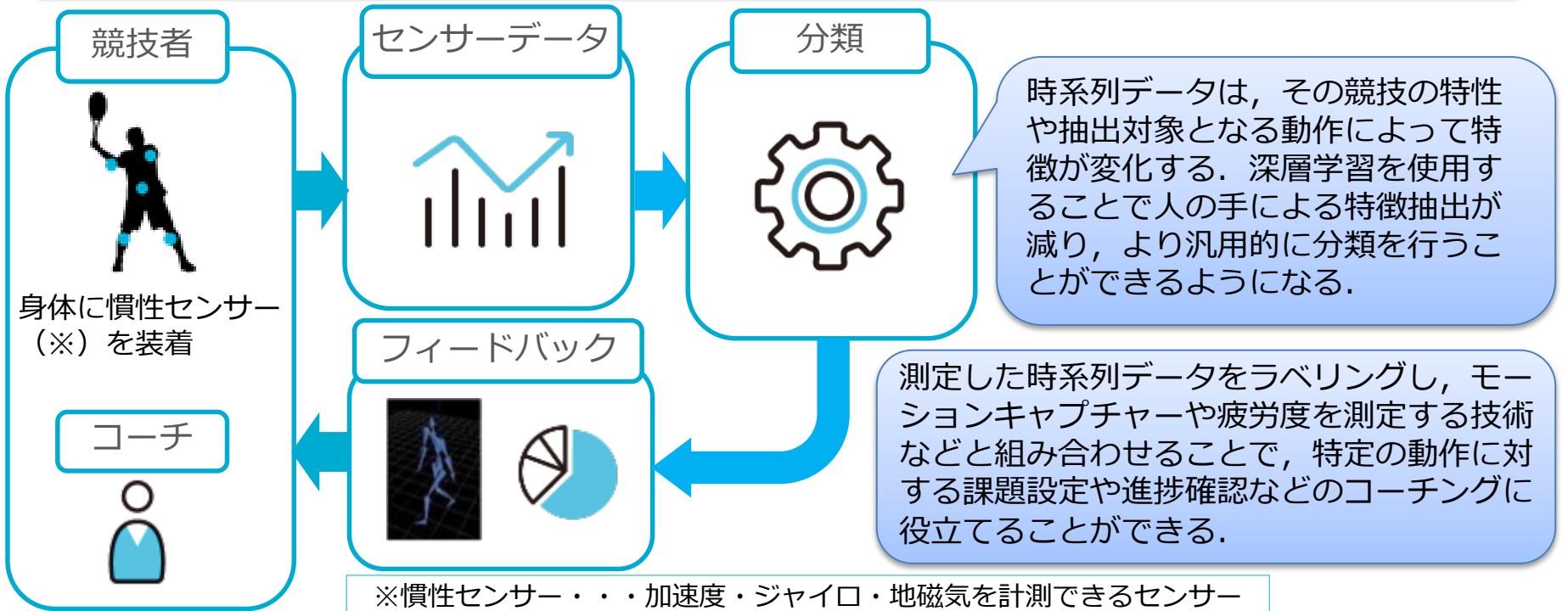
センサーデータの活用

加速度、ジャイロセンサーを使い、人間のモーションや疲労度合などを捉える技術をコーチングで活用したい。しかしながら、センサーデータは画像に比べ、人間にとって直感的でないため、特定の動作に着目するには人間がわかるようにラベリングする必要がある。

手法・ツールの適用による解決

画像分類で精度を上げているCNNを用いてセンサーデータの活動認識をする。この手法は歩行や階段の昇降など日常的な動作に関する分類で検証されてきた。ここでは、バドミントンを題材にスイングフォームを分類し、有用性を確認する。

提案手法



結果

同一人物によるバドミントンのスイングを2日にわたり、センシングした。

- ①Day1のデータで学習したモデルで、Day1のデータを分類した結果約90%の精度だった。同じ日のデータであれば、日常的な動作の分類と近い精度が出ることを確認した。
- ②1で作ったモデルで、Day2のデータで分類を行った結果約49%の精度だった。Day2は被験者のコンディショニング不良もあり、普段よりも抑えたスイングであった。

今後

分類精度の向上

- ①個人差やコンディショニングの差異を無くすため、サンプリングの数を増加させる。
- ②動作と相関の高い装着箇所を探す。

他の競技における有用性確認

特徴的な動きが少ない動作での分類精度を確認する。

動作以外の情報活用

力の入り具合や緊張などの状態について、筋電や視線情報などを使って認識できないか調査・検討する。

為替レート予測における機械学習システムの モニタリング手法の検討

株式会社 日本総合研究所

北野 健太

kitano.kenta@jri.co.jp

開発における問題点

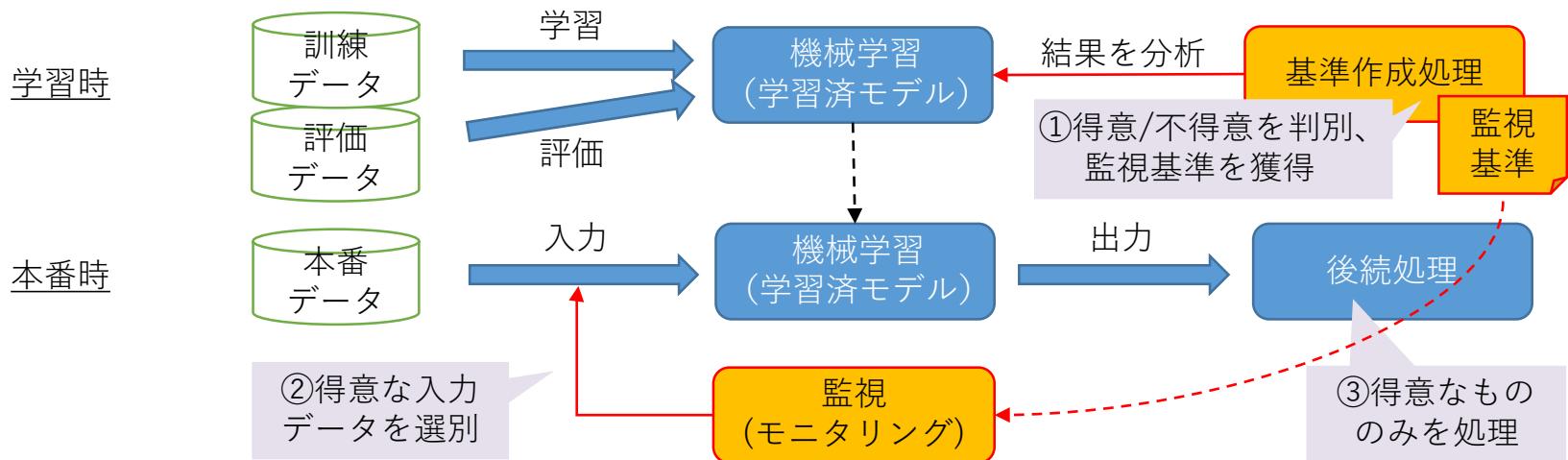
従来のシステム開発において、人間のエンジニアがシステムの振る舞いをプログラムコードとして直接的に記述するのは異なり、機械学習では、訓練データから帰納的に振る舞いが生成される。つまり、人が明示的に定めたルールで動くわけではなく、データを与えてルールを作らせる間接的な手法であり、直接的に制御できず、できることとできないことの境界を明確に把握することも容易ではない。

手法・ツールの適用による解決

本番時(推論時)に機械学習モデルに対する入出力をモニタリングすることで問題解決を目指す。まずは、学習時の評価結果を元にモデルの得意な入力データを特定しフィルタリングの基準を作る。本番時には入力データをモニタリングし、得意な入力データのみ処理を限定する。外国為替レートの予測を行う機械学習モデルを対象にした投資シミュレーションの実験を行い、その有用性を確認した。

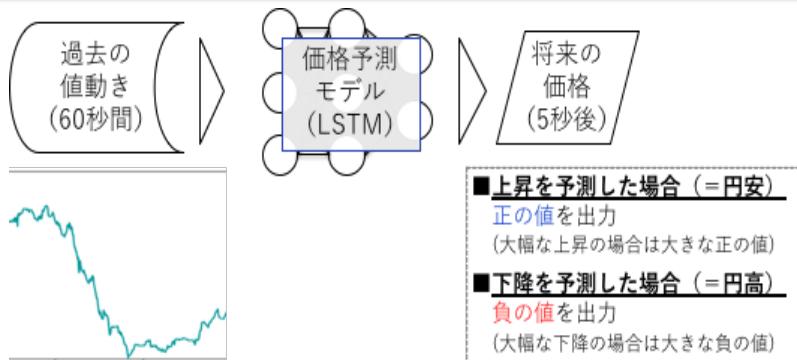
本研究で提案するモニタリング手法

- ・学習時において、入力データの特徴量と正誤判定の結果をみて、モデルの得意/不得意な入力データを判別。
- ・本番時において、入力データをモニタリングし、得意なデータのみ処理する



結果

- ・外国為替カバー取引業務を想定
- ・60秒間の入力を元に5秒後を予測するモデルを作成
- ・分散値を特徴量に活用し、モニタリングを実施
- ・投資シミュレーションの結果、効果を確認した



	取引数	正解率	収益合計(円)
モニタリング無	1,107,433	50.04%	-123,793
モニタリング有	304	61.51%	39,503

今後の展開

- ・モニタリングの基準を自動的に獲得する方式の検討 (現在は、分散、平均などの特徴量を人手で算出し、基準値を策定)
- ・複数のモニタリング基準の候補から、最適なものを選択する方式の導入 (幅広いシチュエーションへ対応)
- ・本番時の出力結果も踏まえたモニタリング手法の検討

新規製品開発時の発想支援ツールの提案

辻脇 優一

yuichi.tsujiwaki@gmail.com

開発における問題点

新規製品を開発する時には、既存の製品に対し、新しい価値を付加する必要がある。製品における価値とは、顧客のゴールを達成することであり、製品とはそのゴールを満たす手段である。筆者の所属するチームでは、現状そのような新しい価値のアイデアは、開発者の閃きや発想に頼っており、コンスタントに生み出すことができていない。

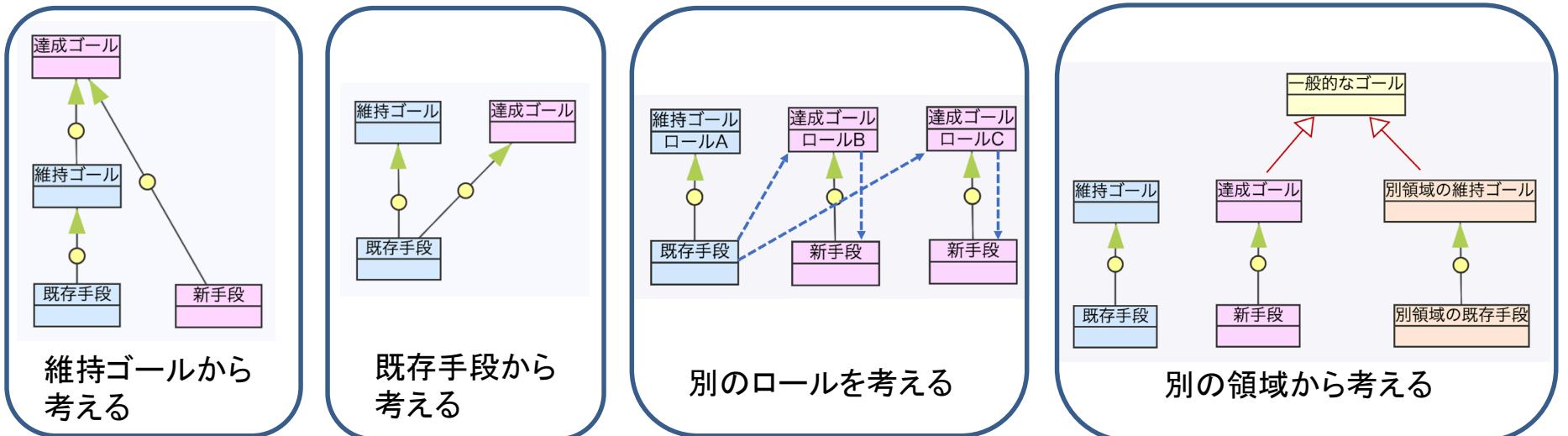
手法・ツールの適用による解決

経験的な知見に基づいて作られた独自の発想支援手法に対し、目的・手段の関係などの表現方法を洗練すべくKAOS法との融合を試みた。実際の新製品開発のアイデアを本手法に適用した結果、新たな発想の構造を可視化することができた。本手法の発想支援ツールも開発し、実務へ適用する際の課題を明らかにした。

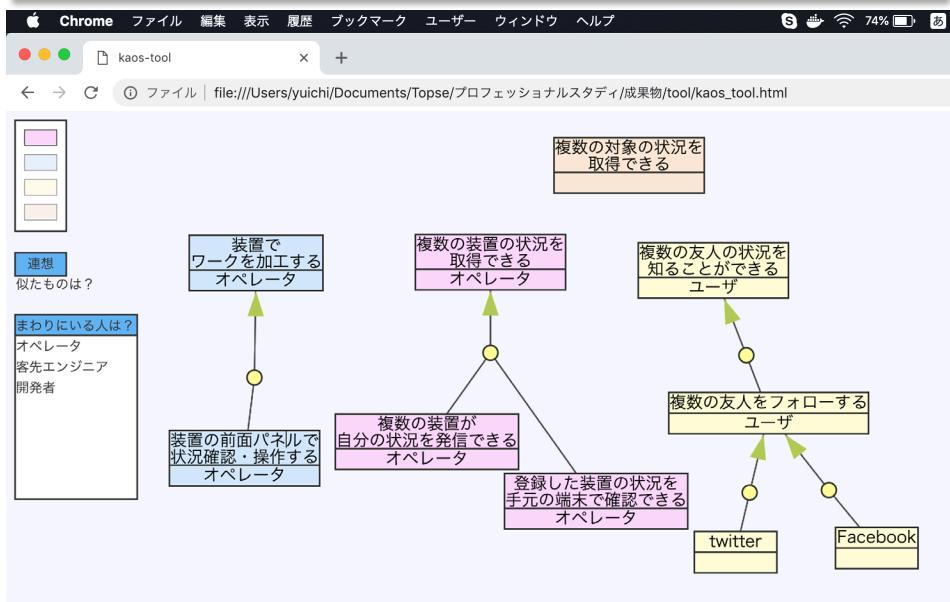
課題解決するためのアプローチ

(既存の)手法+KAOS+ペルソナ

価値を創造するためには達成ゴールを定義しなければならない
 達成ゴールを考えるための考え方を提供することが必要



ツールの開発と例題への適用



現場へ適用する際の課題

- ツールを使うということへの心理的ハードルへの対処
- 実際に効果があるのかという疑念の解消

課題解決への取り組み

- 紙とポストイットを用いて、本手法の考え方の枠組みを導入
- 4つのアイデアで多様なステークホルダの価値を導出するために
- 簡単な質問を用意→思考支援

Webシステムにおけるレイアウト崩れの検出手法の研究

日本電気(株) 早川 芳昭 y-hayakawa@aj.jp.nec.com

画面設計・テストでの問題点

- Webシステムは、デバイスの多様化でマルチデバイス対応(レスポンシブデザイン)、複数ブラウザ対応(Chrome/Firefox/IE...)が求められている。
- 結果として画面レイアウトの設計/評価パターンが複雑化し、システム開発工数の増加・品質低下が問題となっている。しかし、既存のテスト自動化の手法では、この問題を解決できない状況であった。

手法・ツールの適用による解決

- 画面レイアウト設計上、顕著に発生しうる「レイアウト崩れ問題」をターゲットに選び「テスト工数削減と品質保証の両立」に取り組んだ。
- ブラウザのリモート操作技術とUI部品の衝突判定技術を用いてツールを開発。Webサイトを対象に、既存手法(人間の手作業)と提案手法のテスト工数を比較し、最大で約7.8倍の効率化に成功した。

取り組み概要



レイアウト崩れ事例 (横浜市HP)

凡例:

アクション

データ

WEBページ
DOM

ブラウザ
(Chrome Headless)

Webページ

検査結果

検証部

レイアウト検証記述
(JavaScript)

特徴

- 重要バグ「操作ができない」、「情報が読み取れない」、「デザイナーの意図(要求仕様)と異なる表示」が検出対象
- カスタムルール(レイアウト検証記述)をJavaScript構文で記述でき、画面設計を行うエンジニアが利用できるツール
- 検査対象の絶対位置やサイズを算出の上、レイアウト検証記述内の検証式を評価し、検査結果を出力

ツールの構造

評価

- ✓ **評価結果(工数削減の評価)**
 - ✓ 従来手法に比べ13%以下の時間で問題検出が可能
 - ✓ ツールによる自動化で見落としによる検査ミスを防げる
- ✓ **評価概要**
 - ✓ 141箇所を目視で検査する時間と、合計2820秒(約50分)とレイアウト検証記述を作成・ツールで実行すること360秒(6分)を比較
- ✓ **その他評価結果**
 - ✓ 「操作ができない」、「デザイナーの意図(要求仕様)と異なる表示」についても検出可能なことを評価で実証済み
 - ✓ 複数ブラウザ対応の評価結果では、主要なOS/ブラウザの組み合わせに対応できたことを実証済み

考察・今後の展開

- 研究過程において、複数の商用WEBサイトでレイアウト崩れを発見しており、レイアウト崩れの検出の難しさを再認識した。
- 本研究では上記問題も検査にて検出可能なことを確認しており、研究成果(本ツール)を活用することで、ソフトウェア開発の品質向上に貢献できると考える。

本ツールの活用推進のため、対応OS/ブラウザを増やす必要がある。現状スマートデバイス(実機)やIE11において課題があり、解決に向けて順次対応を行う予定

スマートコントラクトを利用した ソフトウェアライセンス認証方式の提案

キヤノン株式会社

前田 泰晴

maeda.yasuharu@mail.canon

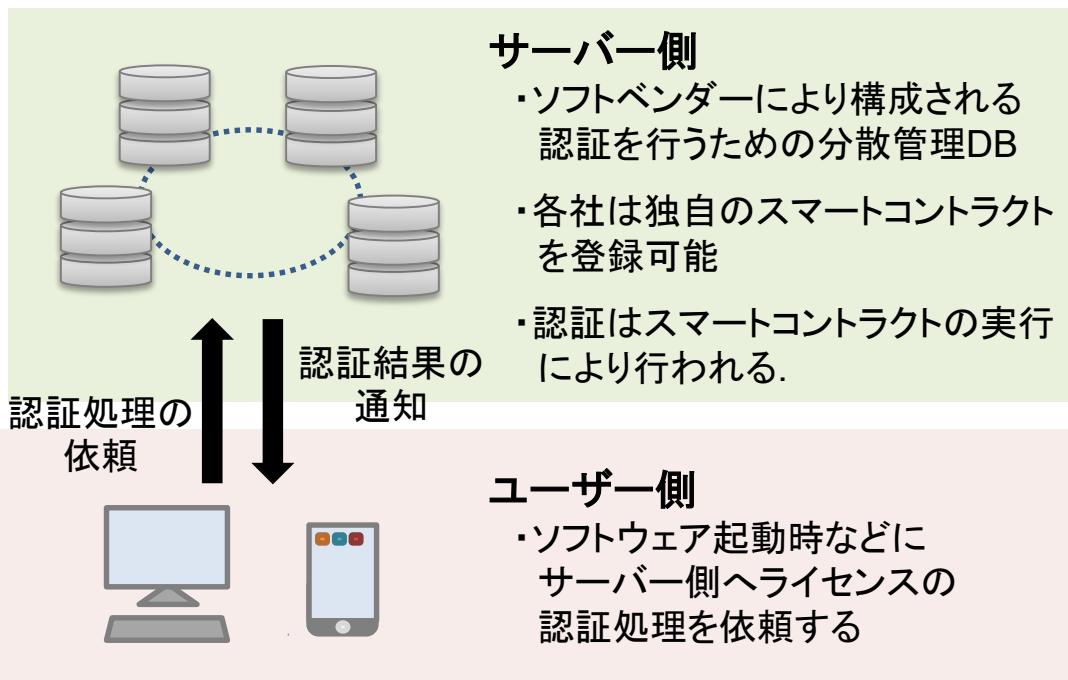
ライセンス認証における問題点

ライセンス認証は、ソフトウェアの不正利用を防止するための手段として重要視されている。従来手法では認証サーバーを設けることが一般的だが、システムの開発費や維持費がかかる。また、Blockchainを利用した分散管理方式が提案されているが、認証処理がクライアント側で実行されるため、セキュリティ上の課題が残る。

手法・ツールの適用による解決

スマートコントラクトは分散管理された基盤上で安全にデータベースへの処理を行う技術である。これを利用することで、複数のソフトウェアベンダーにより分散管理された基盤上で、各ベンダーが独自に定義したライセンス認証処理を安全に処理可能となる。また、提案した手法に関して、処理性能・柔軟性・セキュリティ・コストの観点で評価を行った。

提案手法の概要



特徴

- ① ベンダー毎に独自の認証処理をスマートコントラクトに記述可能なため **柔軟性**の高い認証システムを実現
- ② 認証処理はサーバー側で処理され、更に認証の実行結果は複数のDB間で検証し合うため **高セキュリティ**
- ③ ソフトベンダーが分散管理された認証システムに参画可能な形態をとることで導入にかかる **コスト削減**を実現

評価

処理性能

分散管理DB構築フレームワーク HyperLedger Fabricにて環境を構築して評価を行った。

DBのRead : 約750[ms]
 DBへのWrite : 約3600[ms]

⇒ライセンス認証のユースケースではReadがメインとなるため十分な性能と言える。

セキュリティ

ライセンス認証に関わる従来からの課題だけでなく分散管理DB特有の課題を考慮する必要がある

- ① **多数派獲得によるDBの操作**
⇒悪意のあるノードが参加しにくいコンソーシアム型にする
- ② **脆弱なスマートコントラクトの記述**
⇒記述のガイドラインや、認証クラスの継承による記述

今後の課題

技術的な課題

・ライセンス認証に関わる従来からの課題の根本的な解決方法の検討が必要 (ライセンスキーの複製, コード改竄, etc)

運用上の課題

・認証システムを運営する組織間でのシステム権限、ユーザー数、サーバーのスペックの差による不公平の解消