最先端ソフトウェア工学ゼミ[個別ゼミ1] 成果報告

2021年7月15日

塚田 祥弘 (キヤノン株式会社)
関 堅吾 (株式会社NTTデータ)

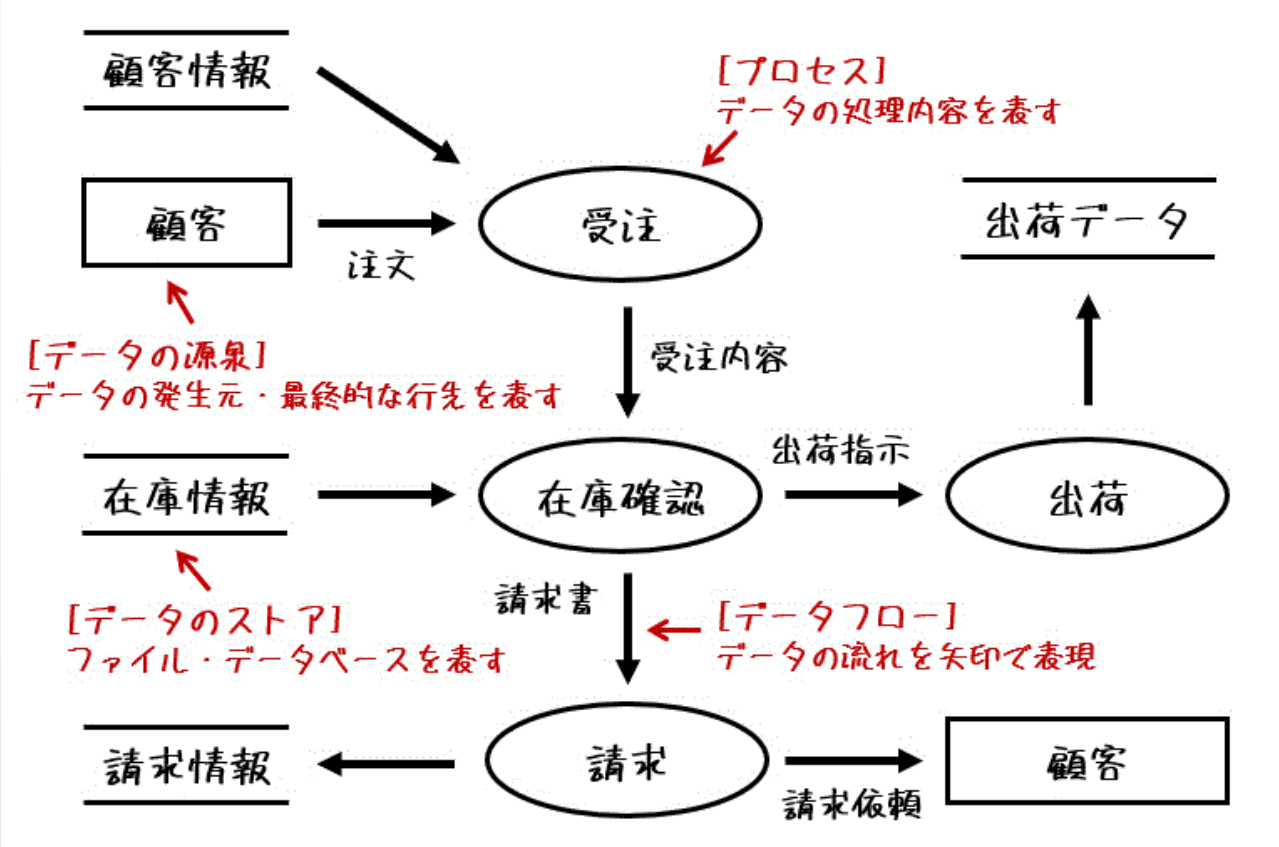
1．設定したテーマとその理由

説明　2分

- テーマ：機械学習を用いた、要求仕様書からの設計ドキュメントの自動生成に関する研究の調査
- 設定した理由
  - 本ゼミ参加者は、いずれも業務で上流工程のレビュアを務めることがあり、設計ドキュメントの品質改善に関心があるため
    - 現場のレビューで重要視されるのはデータ関連の設計図（DFD,DLD）である。
  - 要求仕様から設計ドキュメントを執筆する際に有識者が用いる暗黙知を、機械学習を用いてシステム化することで、以下の効果が期待できると考えた
    - 誤った解釈・要件見落し防止による、成果物の品質向上
    - ドキュメント生成の自動化による、執筆者の負担軽減

DFDが対象の設計書の場合

- 要求仕様書から、データの源泉、プロセス、データのストア、データフローを特定する必要がある。



2. 調査方法

説明　1分

- Scopus から抽出した, “requirement engineering” と “machine learning” の両方に関連する, 2017年以降の国際論文のリスト250件から調査。

- 対象論文のabstract を分析し、要求仕様に関連しそうな23論文に絞り込んだ。
  - 量が多い・対象分野に詳しくないため調査方法を工夫した。
    - クラスター分析（20クラスに分割）
      - Noun, Proper Noun, Verb のみ対象。
      - Ward法, 距離：Cosine, 値：TF-IDF
    - 各クラスの特徴は、共起ネットワーク図作成し視覚判断。

3. 調査結果

説明　0.5分

調査の結果、
「要求仕様からデータ関連の設計図を生成できる論文」は発見できなかった。
ただし、他の設計図を生成する論文はいくつか見られたので、関連しそうな3パターンの論文を共有します。

関連設計図	対象論文
UC図	Identifying Use Case Elements from Textual Specification: A Preliminary Study
FeatureModel	Extracting Software Product Line Feature Models from Natural Language Specifications
クラス図	Towards Queryable and Traceable Domain Models

3－1. ユースケースの抽出に関する研究

説明　2分

- Identifying Use Case Elements from Textual Specification: A Preliminary Study
（Tiwari S., Rathore S.S., Sagar S., Mirani Y.）

- テキストで書かれた要求仕様書からユースケースの要素を特定する予備研究。

研究内容

学習データ作成：
サンプルシステムのデータセット28種に対して手動タグ付けで、UC名、アクター名、その他の3カテゴリに分類。

学習データを元に、UC要素の推論：
右図の手順のように、抽出したワードに対して、学習済みデータを用いて予測。

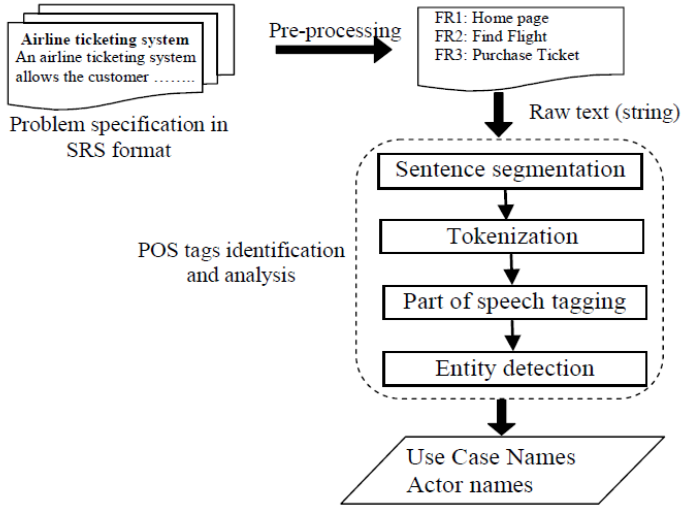
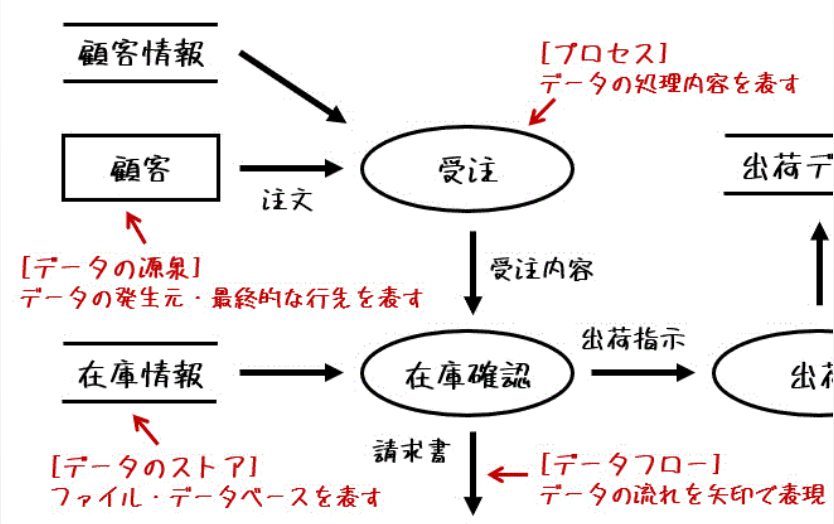


Fig. 1. Overview of the approach

- 結論
  - 基本、代替、例外フロー抽出は本アプローチの拡張では不可。
  - アクター名予測は精度、再現率、Fスコア高い。
    - 精度0.91 再現率1.00 F値0.92
  - UC名予測は精度、再現率、Fスコア低い。

Class labels	Multinomial Naïve Bayes			Perceptron			Linear classifier with SGD			Passive Aggressive classifier		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
UC-NAME	0.34	0.57	0.43	0.56	0.22	0.32	0.54	0.18	0.27	0.38	0.50	0.43
P-ACTOR	0.70	1.00	0.82	0.91	0.86	0.88	0.91	0.87	0.89	0.86	1.00	0.92
Weighted average	0.51	0.78	0.62	0.73	0.53	0.59	0.72	0.51	0.57	0.61	0.74	0.67

所感

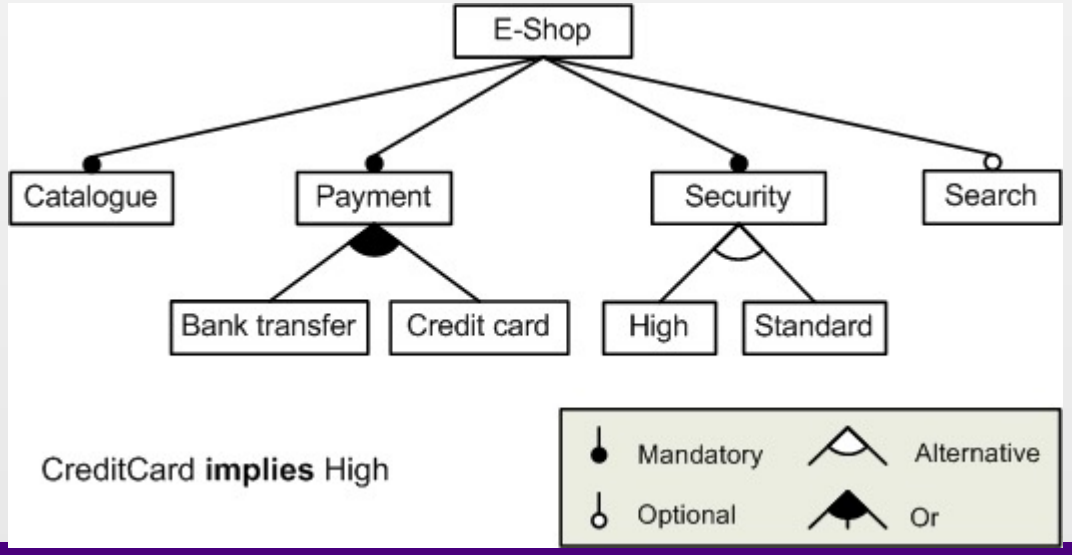


- アクター名はDFDのデータの源泉に使用できそう。
  - 例：顧客は注文する。この場合、アクター名は、顧客となり、「データの源泉」にあたる。
- 要求仕様書中の主要な主語を抽出する際には、perceptron、passive aggressive が使えそう。
- 学習データはどのように用意すべきか。英語で学習した結果なので、改めて日本語での教師データを用意してラベリングする必要がある。

3－2. フィーチャーモデルの抽出に関する研究（１）

説明 2分

- Extracting Software Product Line Feature Models from Natural Language Specifications (A Sree-Kumar, et al., 2018)
- https://doi.org/10.1145/3233027.3233029

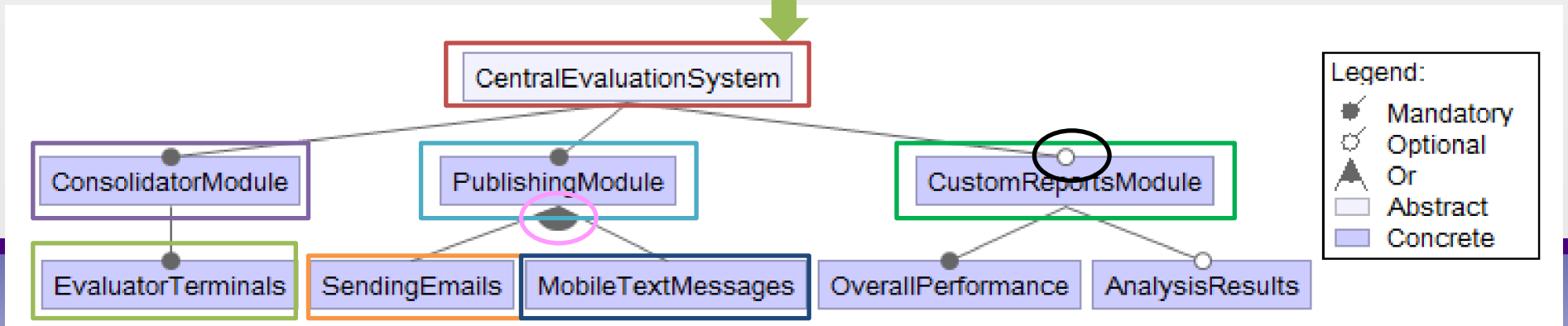


フィーチャモデルの例
(https://en.wikipedia.org/wiki/Feature_model)

研究の目的

- 自然言語で記述されたSPL仕様書からフィーチャモデルを抽出

*"The **central evaluation system** collects all the evaluation results from the various **evaluator terminals** using the **consolidator module** and enters them into the results database where the scores for various courses are segregated or grouped based on candidates. These results will be used by the **publishing module** which is responsible for **sending emails** or **mobile text messages** to the candidates with their complete score cards. The publishing module can be configured using a **custom reports module** for generating customized views of the reports which can be used for understanding the **overall performance** of the class and provide informational **analysis results** on the exam to the teachers and other administrators."*



先行研究の課題

- 先行研究は非公開のツールを用いてなされており結果を再現できない
- 本研究では、一般に利用可能な以下のツールを組み合わせて、フィーチャモデルを抽出・構築可能なフレームワークである FeatureX を提案 (https://github.com/5Quintessential/FeatureX)
  - NLTK (https://www.nltk.org/)
  - Pattern (https://github.com/clips/pattern)
  - WordNet (https://wordnet.princeton.edu/)
  - PDFMiner (https://github.com/euske/pdfminer)

FeatureXの処理フロー

FeatureXの処理の流れは以下の通り。

1. 字句解析でトークン化や品詞のタグ付けを行い、名詞句を抽出してモデル中のフィーチャの候補とする
2. 全フィーチャ候補の TF-IDF を計算して、もっともスコアが高かったものを最上位のフィーチャとする
3. 文章中で主語・目的語として出てくる順序に従い、フィーチャモデルのツリーを構成していく
4. あらかじめ定めた品詞の並びが見つかった場合は、それに応じてモデルに情報（必須・任意・排他など）を付与したり、ツリー構造を修正したりしていく。以下のような要素をもとに判断。
   - 助動詞: can, could, may, might, must, shall, should, will, would
   - 副詞: often, never, always, frequently, normally, usually, generally, regularly, occasionally, sometimes, hardly, rarely
   - 限定詞: all, every, each, any, some
   - if-then 構文 など

結果

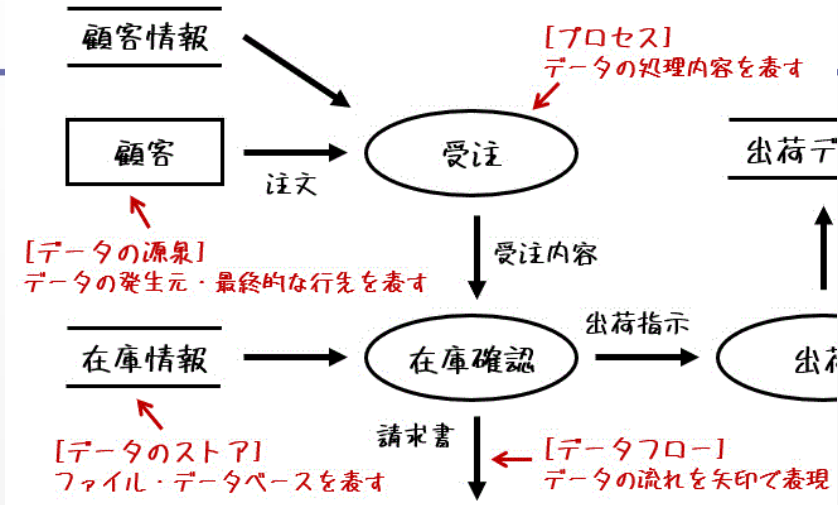
- 先行研究と比較して、フィーチャおよびフィーチャ間関係のrecallが改善し、precisionも同程度（と著者らは主張）
- 先行研究と異なり、本研究のツールはPythonで実装されており、広く一般に利用可能

Table 8: Comparing feature and relationship extraction between FeatureX and state-of-the-art approaches.

	FeatureX								Other tools							
	Features				Relationships				Features				Relationships			
Case study	Rel	Extr	**Precision**	**Recall**	Rel	Extr	**Precision**	**Recall**	Rel	Extr	**Precision**	**Recall**	Rel	Extr	**Precision**	**Recall**
CS(1a)	15	32	0.44	0.82	14	31	0.45	0.75	-	-	-	-	-	-	-	-
CS(1b)	20	27	0.73	0.77	23	26	0.87	0.76	-	-	-	-	-	-	-	-
CS(2a)	13	32	0.41	0.88	12	31	0.41	0.58	-	-	-	-	-	-	-	-
CS(2b)	15	26	0.59	0.65	16	25	0.65	0.68	7	13	0.53	0.55	6	12	0.50	0.37
CS(2c)	6	16	0.40	0.57	8	15	0.54	0.48	13	24	0.54	0.45	12	23	0.57	0.43
CS(2d)	13	29	0.46	0.93	17	28	0.63	0.52	16	27	0.59	0.57	15	26	0.66	0.55

Rel: Number of extracted features considered relevant by a domain engineer. **Extr**: Total number of extracted features (both relevant and irrelevant).

所感

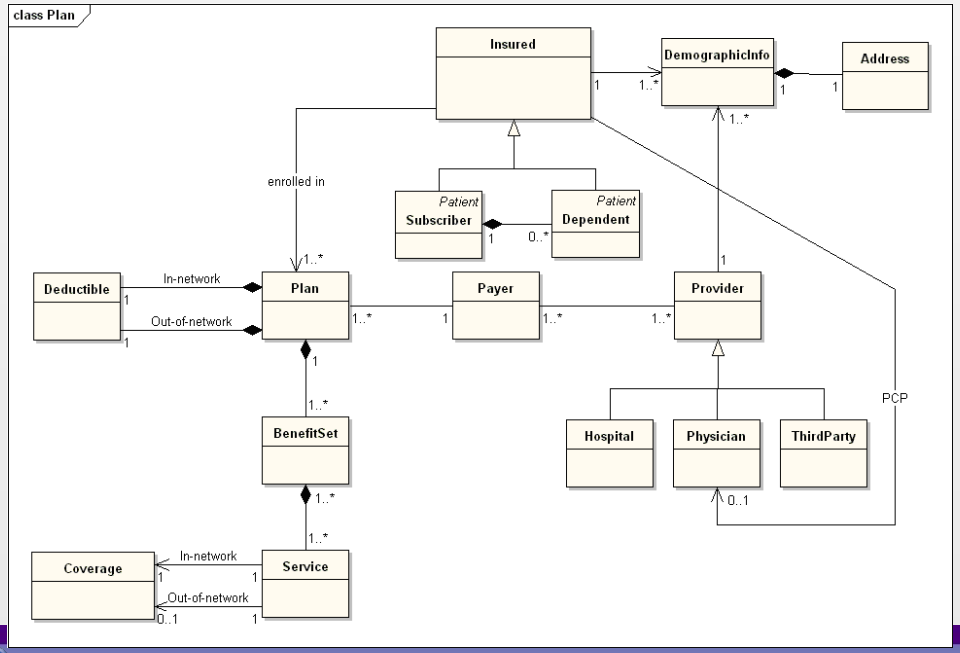


- 文章中の主語を抽出することで、(システム内の処理の主体である)プロセスの候補を特定できそう
  - 例:「**受注システム**は顧客から注文を受け取り、顧客情報DBと突合して受注内容を在庫システムに送信する」
  - 書き方によってはデータの源泉も主語になりそうなので、識別する手段が必要(例:「**顧客**は受注システムに注文を送信する」)→3-3で議論
- 同様に目的語を抽出することで、データフローの候補を特定できそう
  - 例:「受注システムは顧客から**注文**を受け取り、顧客情報DBと突合して**受注内容**を在庫システムに送信する」
  - 書き方によってはデータストアも目的語になりそうなので、識別する手段が必要(例:「受注システムは**顧客情報DB**を参照し、顧客が存在することを確認」)→3-3で議論
- 抽出された候補が単なる一般名詞か、システム中の固有名詞かは、名詞句かそうでないかと、TF-IDF によるスコアリングである程度判別できそう
  - 例:「受注(システム|プロセス)」「顧客情報データベース」

説明 2分

3－3. クラス図の抽出に関する研究（1）

- Towards Queryable and Traceable Domain Models (R. Saini, et al., 2020)
- https://ieeexplore.ieee.org/document/9218176



クラス図の例
(https://ja.wikipedia.org/wiki/%E3%83%89%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%A2%E3%83%87%E3%83%AB)

概要

- ドメインの問題を記述した文書から、ドメインモデルを自動で抽出するためのフレームワークを提案
- NLPベースの既存研究に、MLを導入することで精度を改善

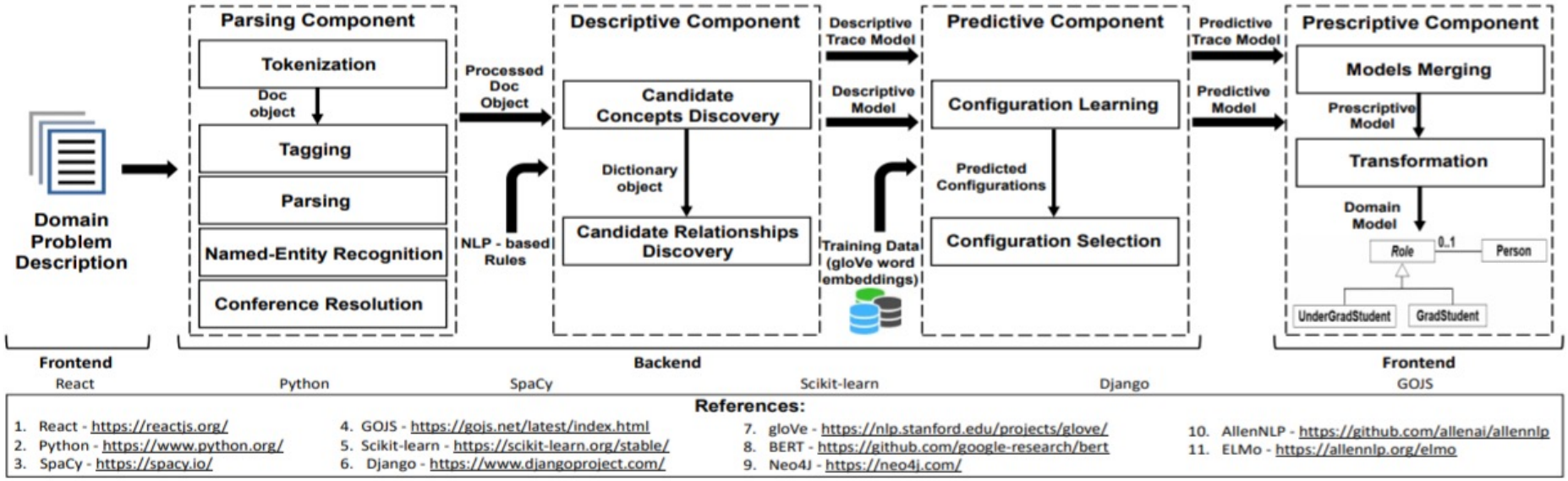


Fig. 1. Our Approach for Queryable and Traceable Domain Models

処理フロー

1. トークン化、品詞タグ付けなどの字句解析
2. 名詞句を抽出することで、ドメイン内のエンティティ候補を発見。また主語—動詞—目的語の関係から、エンティティ間の関連の候補を発見。
3. それぞれのエンティティをクラスと属性のどちらとしてモデル化するか、後者の場合データ型は何にするかを、機械学習を用いて判定
4. 3までで構築したモデルをクラス図として可視化し出力

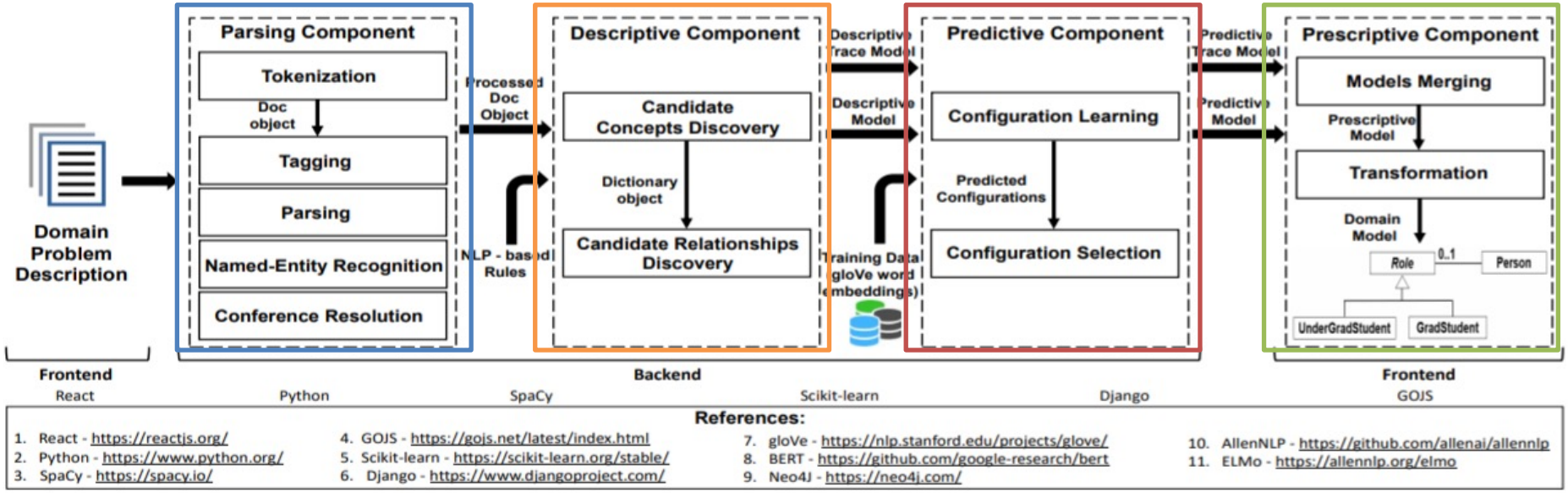


Fig. 1. Our Approach for Queryable and Traceable Domain Models

結果：先行研究との比較

- Arora et al. (2016) をベースラインとして比較すると、すべての課題について正答率が向上した
- 特に、MLを組み合わせることで属性推定の精度が大幅に改善した

Fig. 5. Evaluation of Extracted Domain Models (3 Case Studies)

所感

- 名詞句をエンティティとして抽出したり、主語ー動詞ー目的語の組合せからエンティティ間の関係を見出す手法は、3-2の研究と同様。
- 抽出したエンティティの種類（クラス or 属性）を判定するのに ML を使って精度を上げているのが特徴。
  - 3-2の所感で、「品詞に基づく判断だけではプロセスとデータの源泉、データフローとデータストアの識別が難しいかもしれない」と述べたが、この課題にもMLが使えるのではないか。
  - 先に挙げた例で言えば、「注文を『受け取る』」「受注内容を『送信する』」なら、目的語が指すものはデータフローだが、「顧客情報DBを『参照する』」ならデータストアである。
  - 人間は後続する動詞に基づいてこういった判断を行っているが、類語は多数存在し、それらに対してルールを逐一記述するのは現実的でないため、MLで例を与えて学習させるのが適していそうに思える。

3. 調査結果(まとめ)

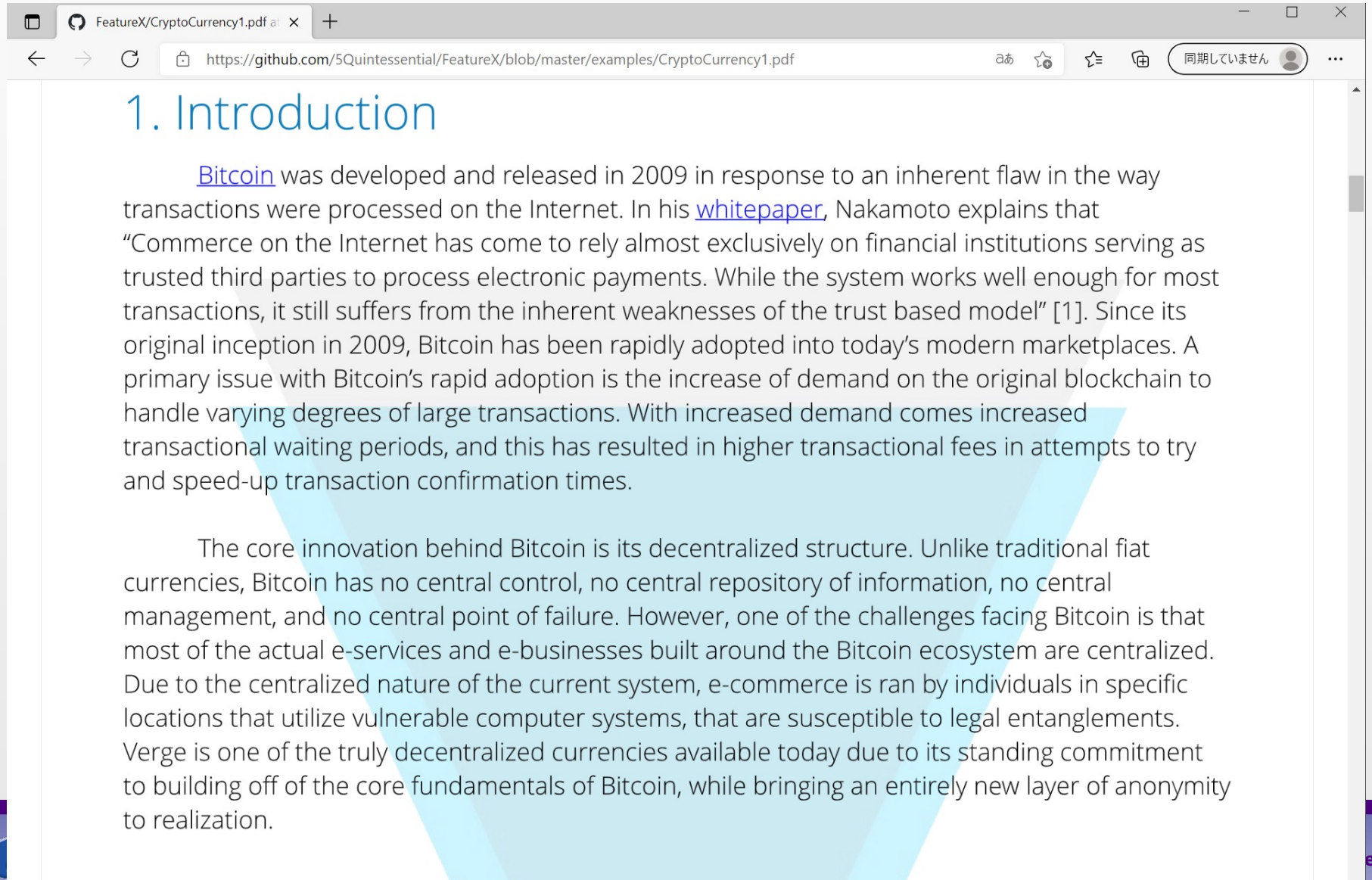
説明　1分

- 調査結果のまとめ
  - DFDの各要素を特定できそうな手段を見つけることができた。
    - データの源泉(3-1)
    - プロセス(3-2)
    - データストア(3-2,3-3)
    - データフロー(3-2,3-3)
  - FeatureX(3-2)の仕組みで、DFDの要素は概ね抽出可能であるため、FeatureXを参考にDFD抽出手法を確立する。
    - Step1:FeatureXを用いたフィーチャモデル抽出の実践
    - Step2:FeatureXをベースに3-1,3-3の手法を取り込む。(こちらはゼミ2)

4. FeatureXを用いたフィーチャモデル抽出の実践

- 目的
  - 3-2で紹介したFeatureXを実際に動かしてみることで、自然言語処理や機械学習を体験するとともに、FeatureXが我々の目的に使えそうかどうか、そのままでは使えない場合はどういった課題が存在するかを明確にする。
- 入出力
  - 入力：英語で記述された暗号通貨の仕様書
  - 中間生成物：品詞タグ付け結果、フィーチャ候補群、最上位フィーチャ
  - 出力：ツリー状のフィーチャモデル
- 所感
  - 使用しているNLPのライブラリが日本語に対応していない
  - コードの品質も特に高いわけではなく、そのまま使うのは難しそう
    - 明らかなバグや論文の記述との乖離も存在し、そのうちいくつかは本体にフィードバックを実施 (https://github.com/5Quintessential/FeatureX/pull/2, https://github.com/5Quintessential/FeatureX/issues/3).

入力


FeatureX/CryptoCurrency1.pdf
https://github.com/5Quintessential/FeatureX/blob/master/examples/CryptoCurrency1.pdf
同期していません
1. Introduction
Bitcoin was developed and released in 2009 in response to an inherent flaw in the way transactions were processed on the Internet. In his whitepaper, Nakamoto explains that "Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model" [1]. Since its original inception in 2009, Bitcoin has been rapidly adopted into today's modern marketplaces. A primary issue with Bitcoin's rapid adoption is the increase of demand on the original blockchain to handle varying degrees of large transactions. With increased demand comes increased transactional waiting periods, and this has resulted in higher transactional fees in attempts to try and speed-up transaction confirmation times.
The core innovation behind Bitcoin is its decentralized structure. Unlike traditional fiat currencies, Bitcoin has no central control, no central repository of information, no central management, and no central point of failure. However, one of the challenges facing Bitcoin is that most of the actual e-services and e-businesses built around the Bitcoin ecosystem are centralized. Due to the centralized nature of the current system, e-commerce is ran by individuals in specific locations that utilize vulnerable computer systems, that are susceptible to legal entanglements. Verge is one of the truly decentralized currencies available today due to its standing commitment to building off of the core fundamentals of Bitcoin, while bringing an entirely new layer of anonymity to realization.

中間生成物

Activities Text Editor Jul 8 18:39

Open SubjectObject.txt ~/repos/FeatureX/examples/2021-07-08_18-29-59 Save

	WORD	TAG	CHUNK	ROLE	ID	PNP	LEMMA
2							
3	Because	IN	PP	-	-	-	because
4	the	DT	-	-	-	-	the
5	routing	VBG	VP	-	-	-	rout
6	of	IN	PP	-	-	PNP	of
7	communication	NN	NP	SBJ	1	PNP	communication
8	is	VBZ	VP	-	1	-	be
9	partly	RB	VP ^	-	1	-	partly
10	concealed	VBN	VP ^	-	1	-	conceal
11	at	IN	PP	-	-	PNP	at
12	every	DT	NP	-	-	PNP	every
13	hop	NN	NP ^	-	-	PNP	hop
14	in	IN	PP	-	-	PNP	in
15	the	DT	NP	-	-	PNP	the
16	Tor	NNP	NP ^	-	-	PNP	tor
17	circuit	NN	NP ^	-	-	PNP	circuit

Plain Text Tab Width: 8 Ln 1, Col 1 INS

Open CandidateTerms.txt ~/repos/FeatureX/examples/2021-... Save

```
1 'routing', 'communication', 'concealed', 'hop', 'tor', 'circuit',
  'method', 'single point', 'communicating peers', 'be',
  'determined', 'network surveillance', 'source', 'destination',
  'ip integration ip', 'built', 'provide', 'hidden services',
  'people', 'host', 'servers', 'unknown locations', 'ip', 'many',
  'same benefits', 'anonymous access', 'online', 'content', 'use',
  'pp-style routing structure', 'layered', 'encryption',
  'designed', 'network', 'internet', 'see', 'figure', 'traffic',
  'contained', 'borders', 'performs', 'based', 'opposed', 's
  circuit based', 'benefit', 'route', 'congestion', 'service
  interruptions', 'manner', 'similar', 's ip', 'level',
  'reliability', 'redundancy', 'first time', 'client', 'contact',
  'query', 'distributed', 'network database', 'custom structured',
  'distributed hash table', 'dht', 'kademlia', 'algorithm [ ]',
  'done', 'find', 'other client', 'inbound tunnels', 'subsequent
  data', 'information', 'further network database lookups',
  'required', 'obfuscated', 'service', 'ipv', 'verge', 'data',
```

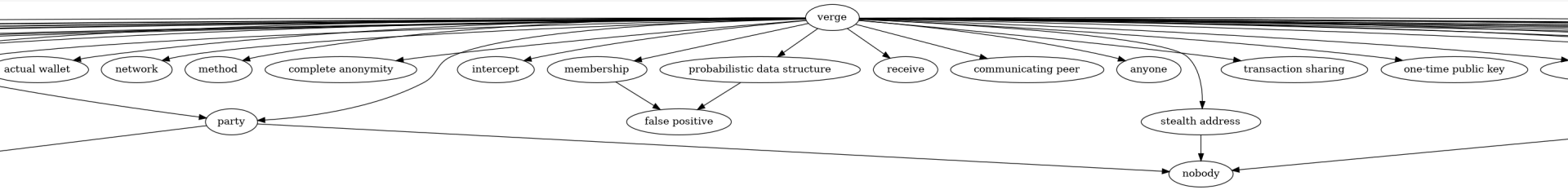
Plain Text Tab Width: 8 Ln 1, Col 419 INS

Open RootFeature.txt ~/repos/FeatureX/examples/2021-... Save

```
1 Verge
```

Plain Text Tab Width: 8 Ln 1, Col 6 INS

出力



- 実際のマニュアルを対象にした際のFeatureXの実力を確認するために
  キヤノン製品Canon i-SENSYS MF742Cdwの欧州向けユーザマニュアル（英語）に対してFeatureXによる解析を実行。

対象は機体管理の章
（p.359-p456）

Outline

Managing the Machine

3S21-06X

To reduce the various risks associated with the use of this machine, such as leaks of personal information or unauthorized use by third parties, constant and effective security measures are required. An administrator should manage important settings, such as access privileges and security settings, to ensure that the machine is used safely.

■Configuring the Basic Management System

Setting Access Privileges(P. 361)

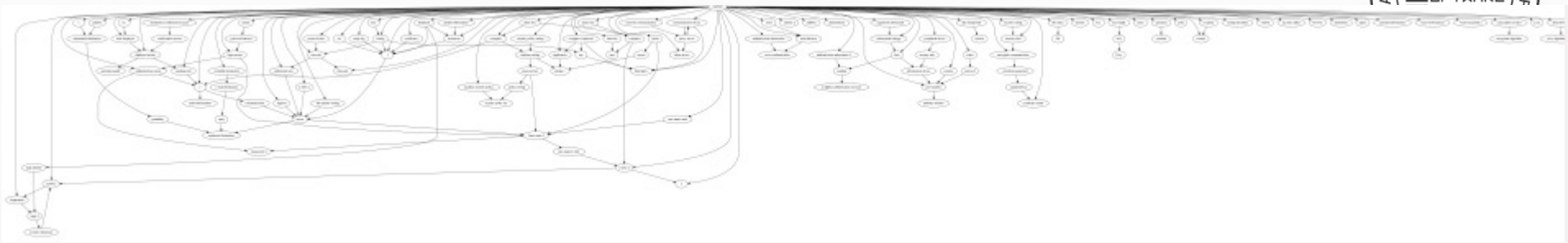
Configuring the Network Security Settings(P. 376)

■Preparing for Risks from Negligence or Misuse

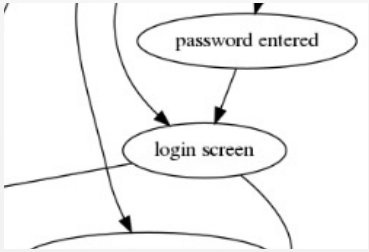
Restricting the Machine's Functions(P. 410)

Increasing the Security of Documents(P. 424)

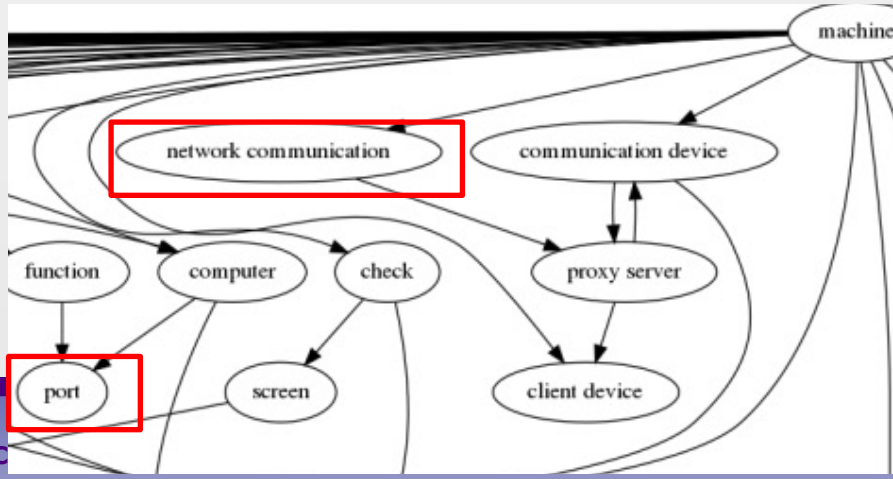
■Ensuring Effective Management



- 結果:
  - rootFeatureがmachineであることを特定。
  - DFDのデータフローとプロセスに関連するものを関連付けれていた。
    例:「password entered 」と「login screen」の関連。



- 問題点:
  - ドメインエンジニア観点では、関連を示す線が不足している。
    - ネットワークのportがNetwork communicationと関連していない。
  - 同義語などがノードとして出現するので、ツリー構造が複雑化。
- 考察:
  - ドメインのFeature関連図を反映。
    - 別のFeatureModelとの結合。
    - 他の周辺機器のマニュアルの結果も考慮する。
  - 同義語をまとめる仕組みを考える必要がある。
    - 辞書(手動で作ったリスト、MLを使って作るなど)



まとめ

- 本ゼミでは、機械学習を用いて、要求仕様書から設計ドキュメント（特にデータの流れに着目したもの）を自動生成する研究について、ここ4年ほどの論文の調査を行った。

- 自然言語で記述された要求仕様書から設計ドキュメントを自動生成する先行研究としては、ユースケース図・フィーチャーモデル・クラス図などを対象としたものがあり、それらの手法を適用することで、DFDの構成要素の抽出と要素の種類の識別が、ある程度の精度でできそうなことが判明した。

- 論文の一つに、要求仕様書からフィーチャモデルを生成するツールを公開しているものがあったため試してみたが、日本語に未対応・コードの品質が悪い・同一のエンティティがモデルの異なる場所に複数回現れる、といった問題があり、そのまま使うのは難しいことがわかった。

- 個別ゼミ2では、今回学んだ手法を活用することで、日本語で記述された要求仕様書からDFDを自動生成するツールの開発に取り組みたい。