

SysML図間の潜在的な要求トレーサ ビリティの自動検索に向けて

宇田川 佳久

東京工芸大学 工学部 コンピュータ応用学科

発表の流れ

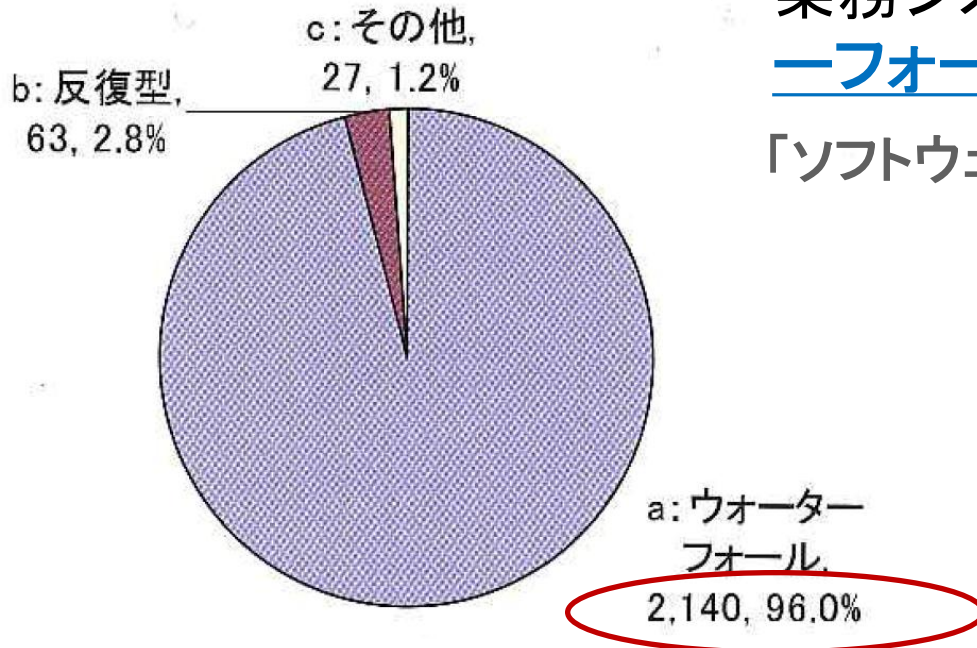
1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間モデル
4. 実験結果
5. 結果の考察
6. おわりに

はじめに

研究テーマ:

要求に合致した設計、実装であることを担保する
システム開発方法論

現実的なアプローチを模索中

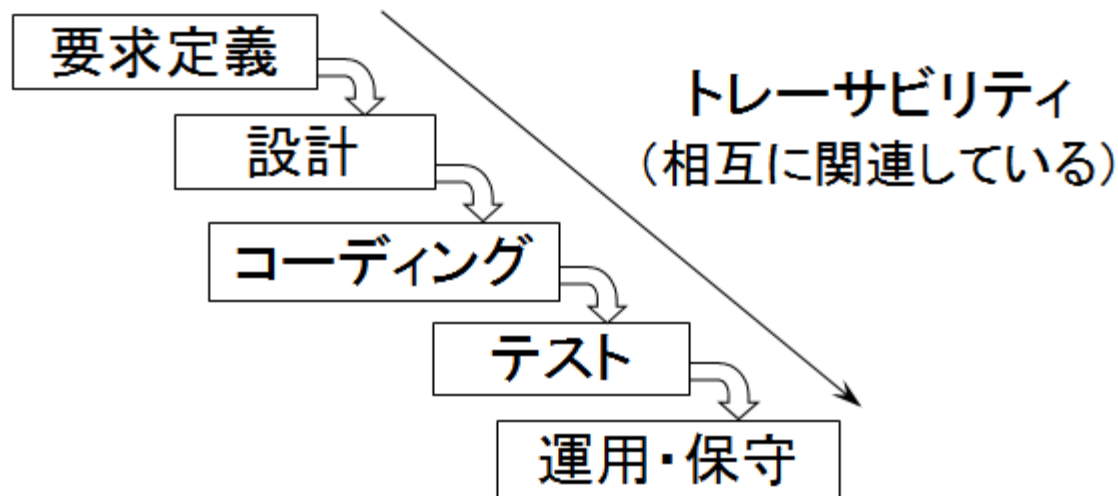


- 業務システム開発の96%は、ウォーターフォールモデルに基づいている。

「ソフトウェア開発データ白書2009」より抜粋

情報システム開発工程（プロセス）

- **ウォーターフォールモデル：**
要求定義、設計、実装へと、各プロセスが完了してから次のプロセスを開始する。
- 要求定義～運用・保守まで、要求/設計/プログラムなどで相互に関連しているが、**この相互関連は管理されていないのが実態。**



要求トレーサビリティを阻む要因

1. システム開発の成果物が異なる言語で記述されている

Artifacts in system development are written in different languages

2. 抽象レベルが異なる

They are described at various abstraction levels

3. 膨大なTRデータとツール支援の不足

Large amount of traceability information and lack of adequate tool support

本研究の趣旨

- 仕様記述手段として**SysML**を採用
 - 要求定義、設計をカバーしているため
- 改良版**ベクトル空間検索モデル**によるトレーサビリティ情報の自動抽出
 - 従来のベクトル空間検索モデルは単語レベルの類似性検索。**文脈を考慮**していない。

関連する研究

- [Maeder et al. \[10\]](#) proposes traceability link model for the development processes through requirements and implementation, especially for the case of changing requirements.
- [Huffman Hayes et al. \[7\]](#) use a vector space model augmented by a thesaurus.
- [Kagdi et al. \[9\]](#) propose an approach that combines information retrieval technique based on a vector space model and mining software repositories for capturing patterns of change history of the entities.

発表の流れ

1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間検索モデル
4. 実験結果
5. 結果の考察
6. おわりに


SysMLによる要求・仕様記述

- ・ SysMLはOMGが定めた、図式汎用モデリング言語
- ・ 最新版は、Ver 1.2 (2010年6月)

No	分類	SysML図式表現	UML2.0との関連
1	要件図	要件図	SysMLで新規に追加
2	振る舞い図	アクティビティ図	UML2.0と同じ
3		シーケンス図	UML2.0から修正
4		ステートマシン図	UML2.0から修正
5		ユースケース図	UML2.0から修正
6		構造図	ブロック定義図
7	内部ブロック図		UML2.0から修正
8	パッケージ図		UML2.0と同じ
9	パラメトリック図		SysMLで新規に追加

Webシステムへのログイン機能要件

- ・入力項目要件 ユーザID
 パスワード
 プロジェクトID
- ・パスワード管理要件(構成・変更ルール)
- ・ログイン失敗時の警告要件



The image shows a mockup of a login screen titled "Login Screen". It contains three input fields: "UserID" (a text box), "Password" (a text box), and "ProjectID" (a dropdown menu with "Project1" selected). Below the input fields is a "LOGIN" button.

Login Screen

UserID

Password

ProjectID

SysMLによる要求定義(階層化機能)

データ項目要件

req Requirement definition

Functional requirement on login function

Id="Req1000"
text="Functional requirement for login process"

Data items requirement

Id="Req1010"
text="The following data items are required."
(1) #UserID // ID for individual
(2) #PassWD // Password for login
(3) #PrjID // ID for project user participated"

Password management requirement

Id="Req1020"
text="Password setting, expiration and change policy must be satisfied."

Login failure requirement

Id="Req1030"
text="In case of login failure, a notification screen must be displayed."

《refine》 #Login_failure_screen

《refine》 #Login_process

《refine》 #Login_screen

Password setting policy

Id="Req1021"
text="Password is encoded rightly if the followings are satisfied."
Length of a password should be 8~12.
Start with an alpha character (a-z or A-Z).
Numbers and special characters(#,%,\$ etc.) are allowed."

PW管理要件

《refine》 #Login_process

《refine》 #Login_process

Password expiration policy

Id="Req1022"
text="Password should be changed within 90 days."
Password change warning be notified 10 days' before expired."

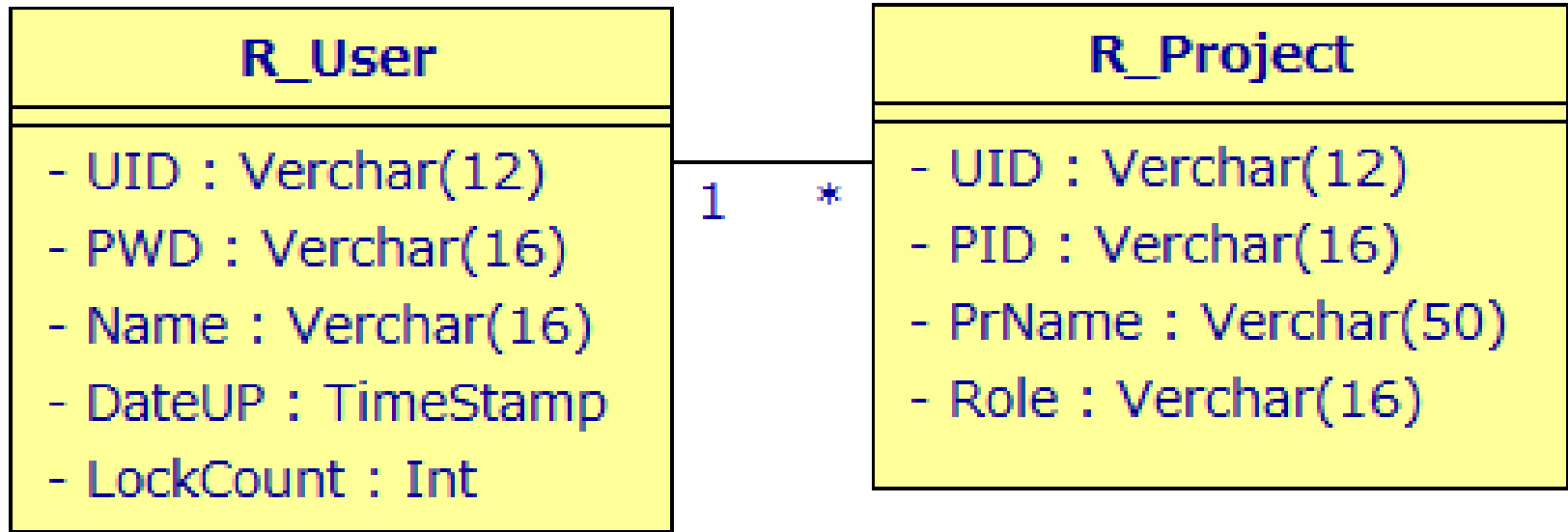
Password change policy

Id="Req1022"
text="The new password should be entered twice before change."

《refine》 #Password_change_screen

ログイン失敗警告要件

データベース設計



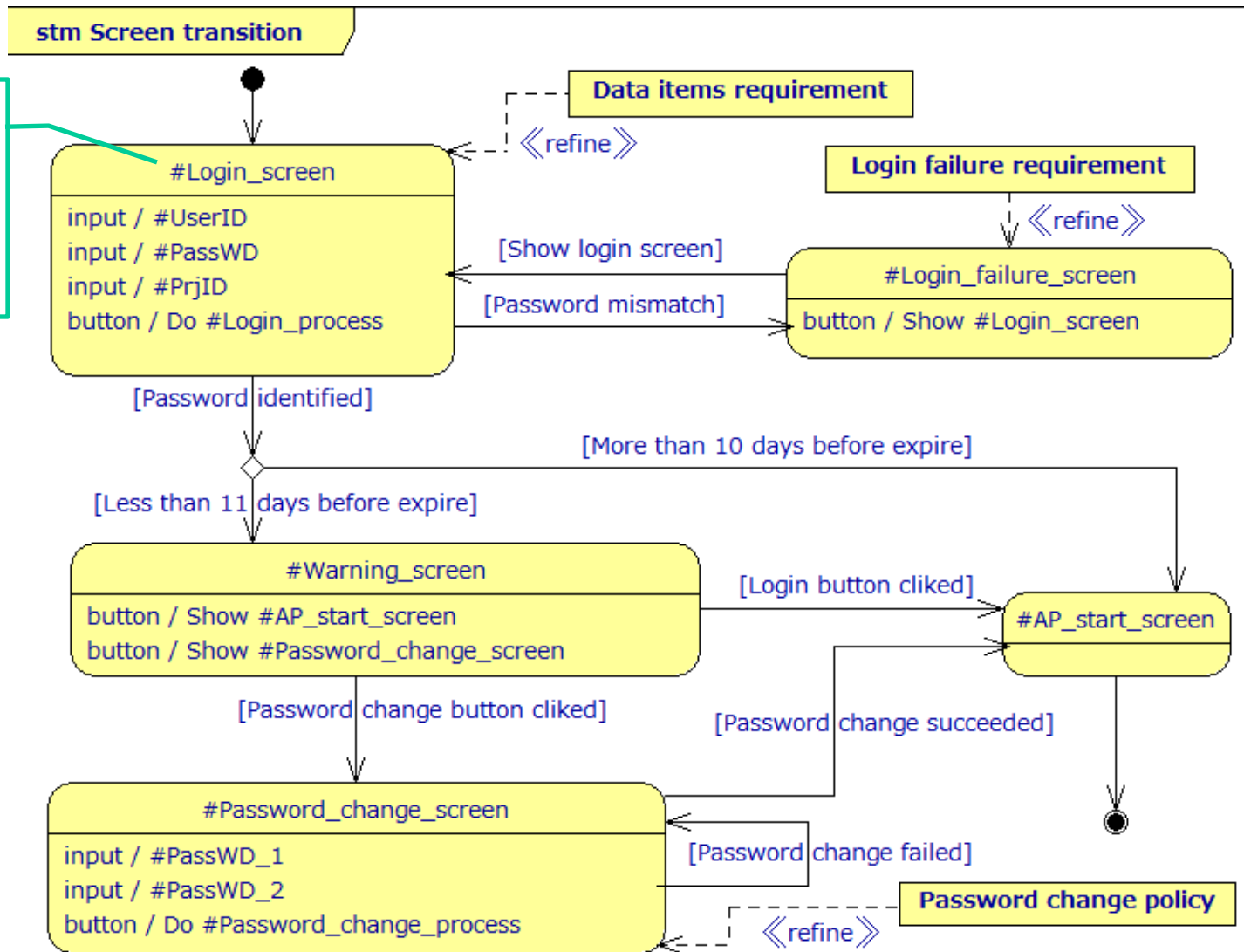
The entity **"user"** has attributes on user id, password, user name, and a date last updated, etc.

The entity **"project"** has attributes on project id, a name of the project, members of the project and his/her role.

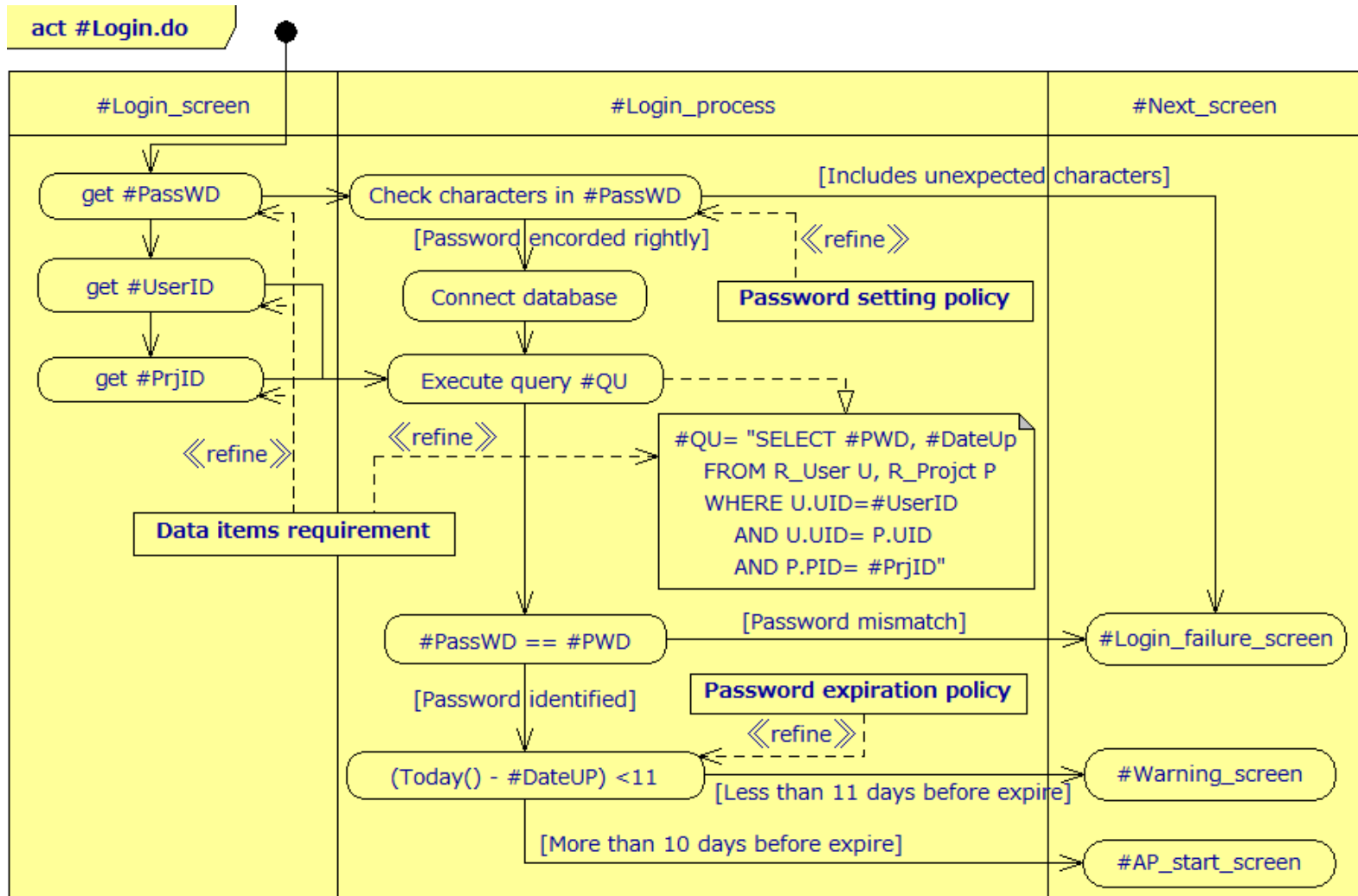
The entity **"user"** and **"project"** are translated into the relation **R_user** and **R_project**, respectively.

SysMLによる画面遷移設計(ステートマシン図)

#印はキーワード。
後述の情報検索
では通常語の約2
倍の重みを持つ。



SysMLによる処理フロー設計(アクティビティ図)



発表の流れ

1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間モデル
4. 実験結果
5. 結果の考察
6. おわりに

ベクトル空間モデル

- ベクトル空間モデルとは情報検索を行うためのアルゴリズムの一つ。
- ベクトル空間モデルによる検索は高次元のベクトル空間上に配置した検索対象のベクトル表現(ドキュメント)と検索語のベクトル表現(クエリ)の相関量を内積等によって計算して関連度を求める。

□ 出典: フリー百科事典『ウィキペディア (Wikipedia)』 (2010/06/22 16:23 UTC 版)

ベクトル空間モデルの計算式

$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{N,j})$$

$$q = (w_{1,q}, w_{2,q}, \dots, w_{N,q})$$

$$\text{Similarity}(d_j, q) = \cos(d_j, q) = \frac{\sum_{i=1}^N w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^N w_{i,j}^2} * \sqrt{\sum_{i=1}^N w_{i,q}^2}}$$

We use a well-known metric called *tf-idf* to compute $w_{i,j}$ [7].

ベクトル空間モデルの改良

1. 抽出単語に、その単語が抽出された**文書のID**を付ける

– 例:

要求定義で抽出された単語には“Requirement”
設計書で抽出された単語には“Design”

2. クエリを構成する主要語を“**ファミリー単語**”として扱う

– 例:

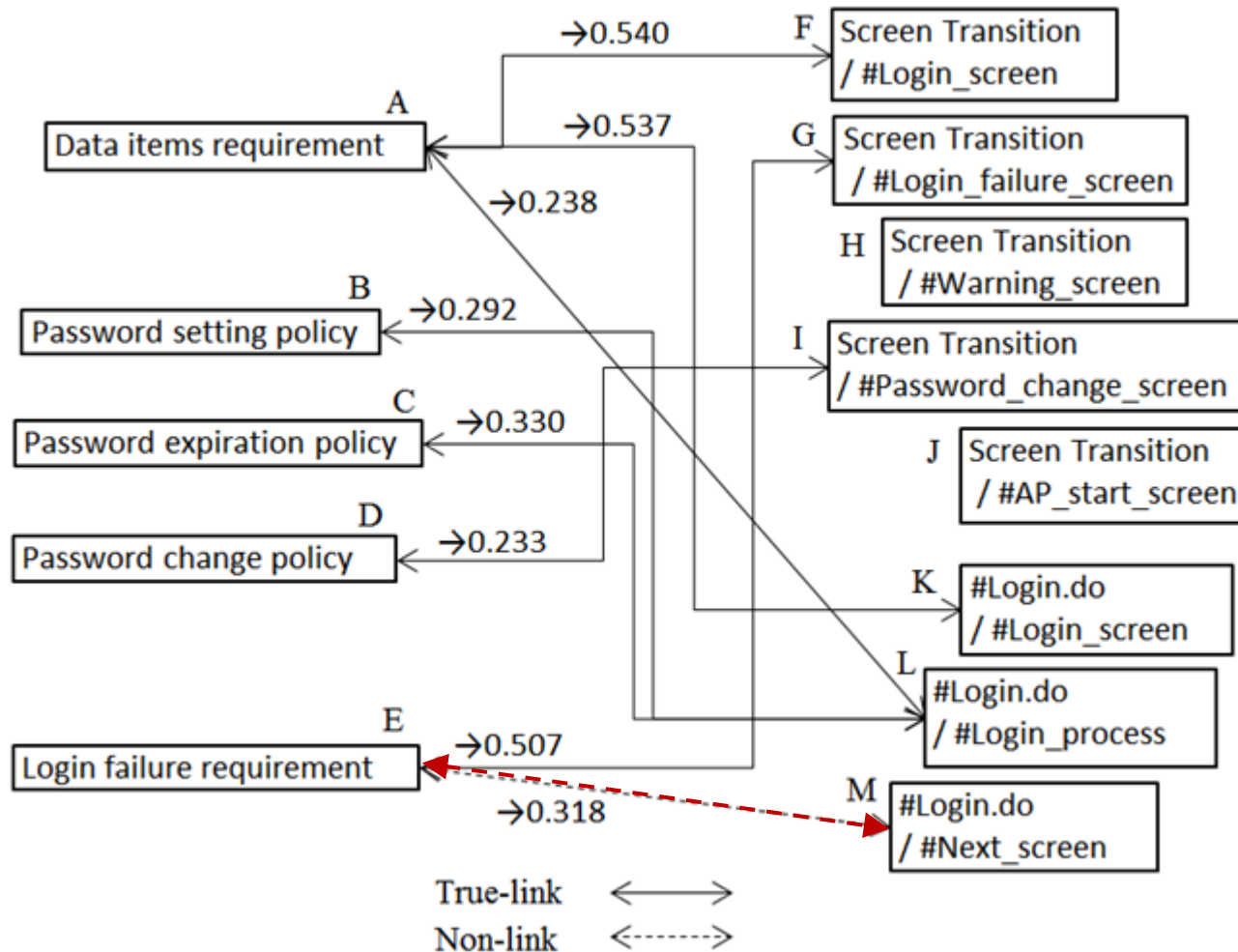
{A, B, C, D}がファミリー単語の場合、ドキュメント(検索対象)が{A, B, C}を含めば、{D}も含まれているものとして扱う(ある種の推論ルール)

発表の流れ

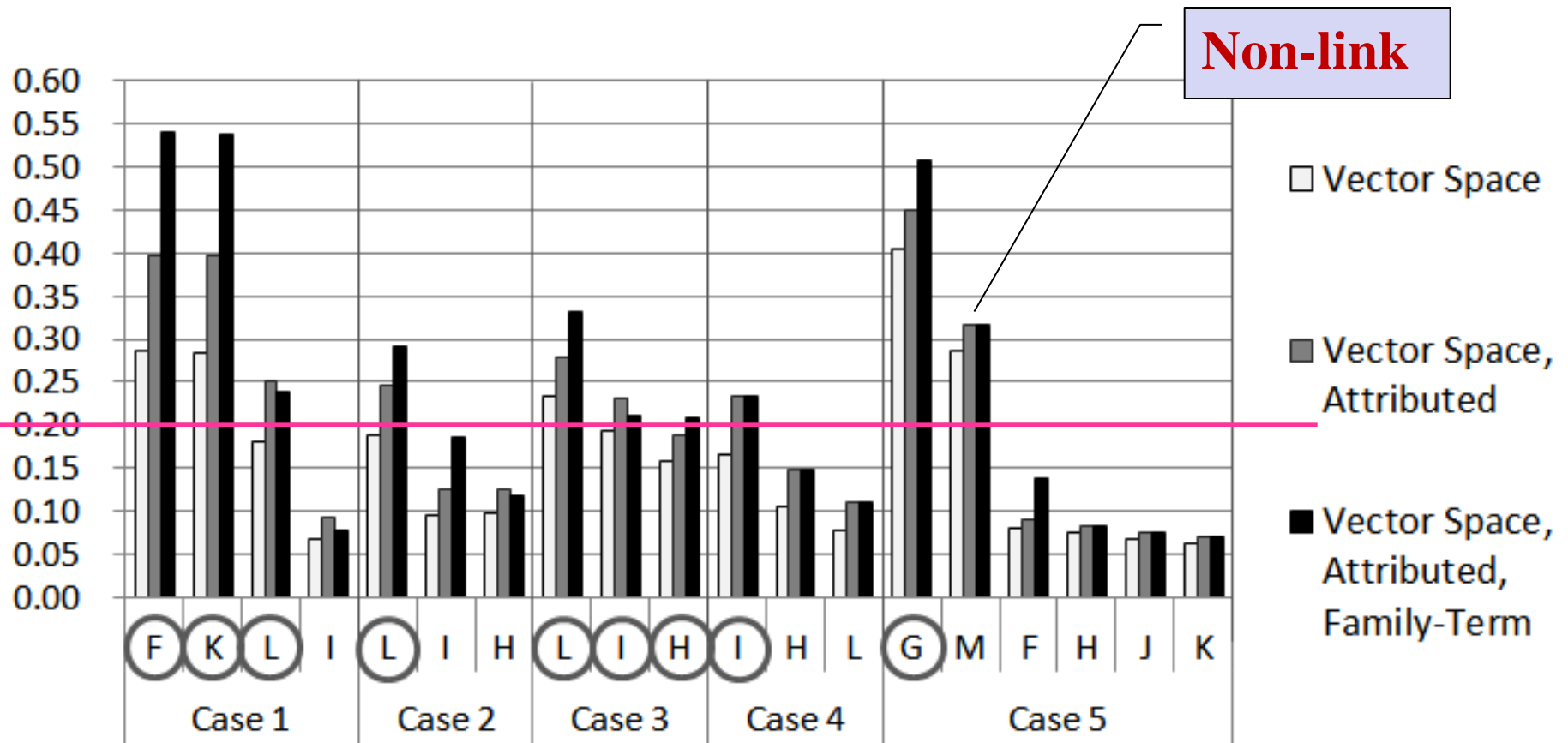
1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間モデル
4. **実験結果**
5. 結果の考察
6. おわりに

実験結果

主要単語は82個、SysML図から13個のドキュメントを抽出
⇒ 82x13のマトリクスに変換



実験1-5の結果



F: #Login_screen

G: #Login_failure_screen

H: #Warning_screen

I: #Password_change_screen

J: #AP_start_screen

K: #Login_screen

L: #Login_process

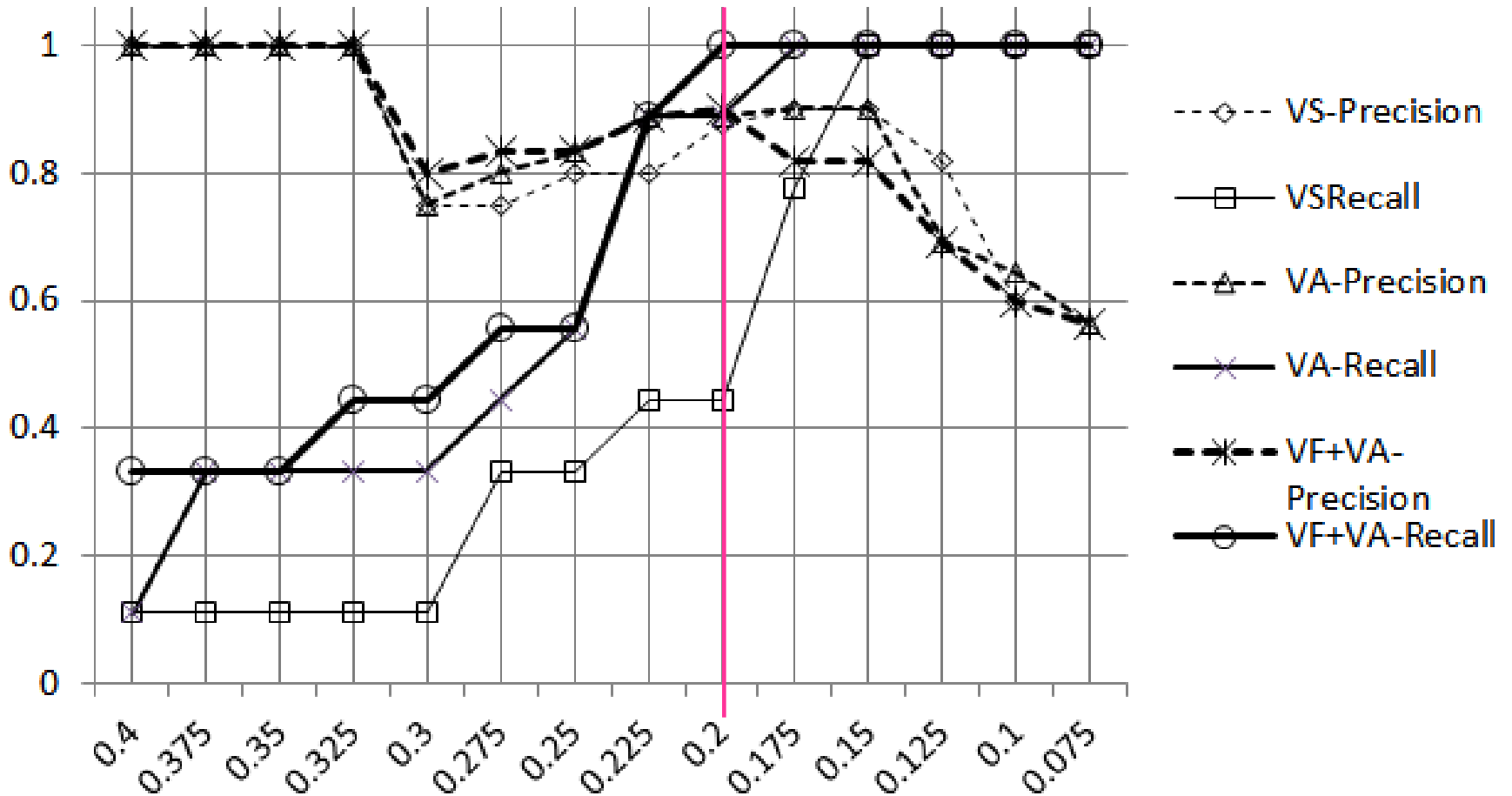
M: #Next_screen

発表の流れ

1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間モデル
4. 実験結果
5. 結果の考察
6. おわりに

結果の考察

再現率(Recall)と精度(Precision)



Recall 100%, Precision >80% の範囲

- VS model: Range of similarity **0.025 width**
- VA model: Range of similarity **0.025 width**
- VF+VA model: Range of similarity **0.05 width**

改良版の方が、広範囲の類似度に対して、再現率100%、精度80%以上を確保している

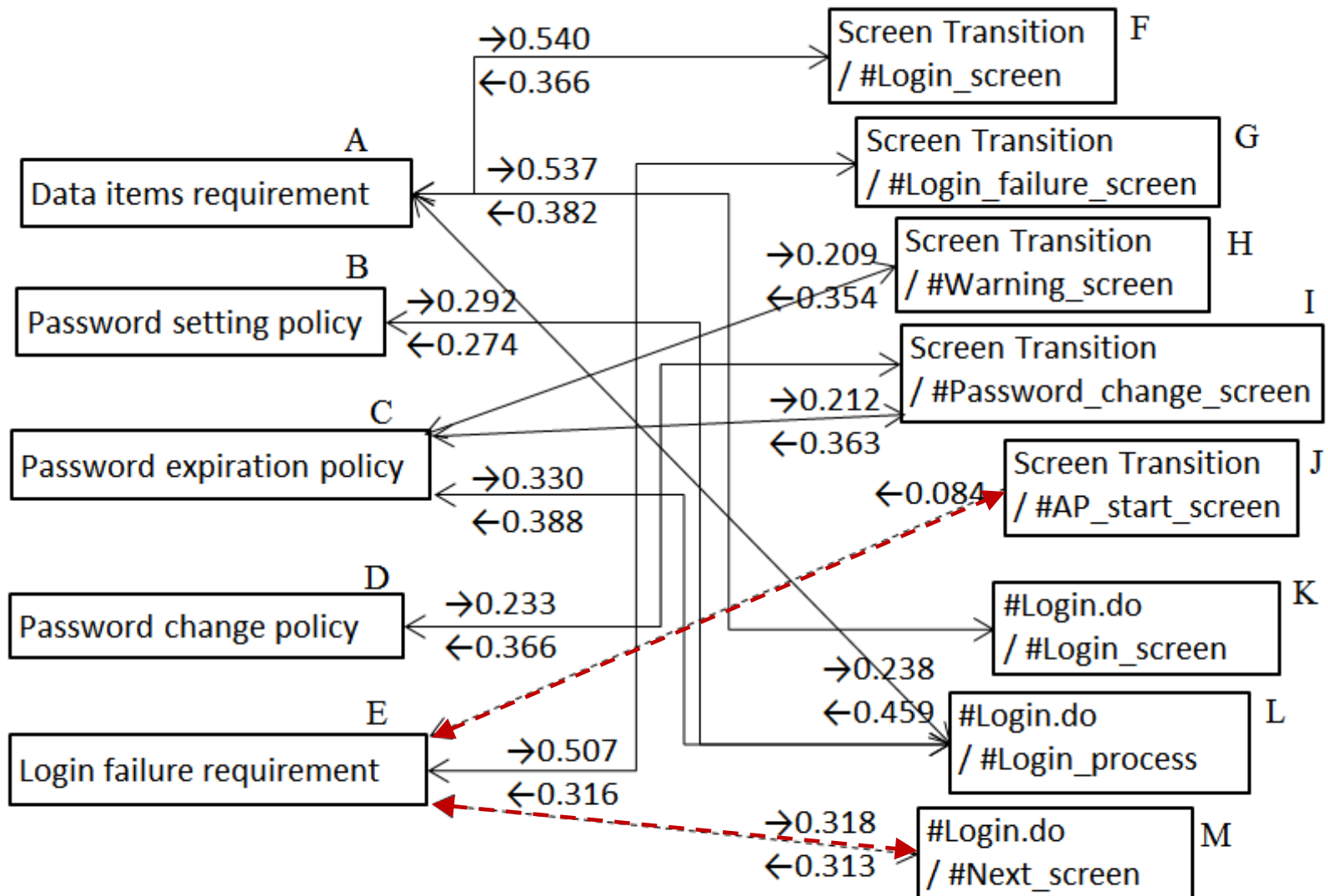
発表の流れ

1. はじめに
2. SysMLによる要求・仕様記述
3. ベクトル空間モデル
4. 実験結果
5. 結果の考察
6. おわりに

おわりに

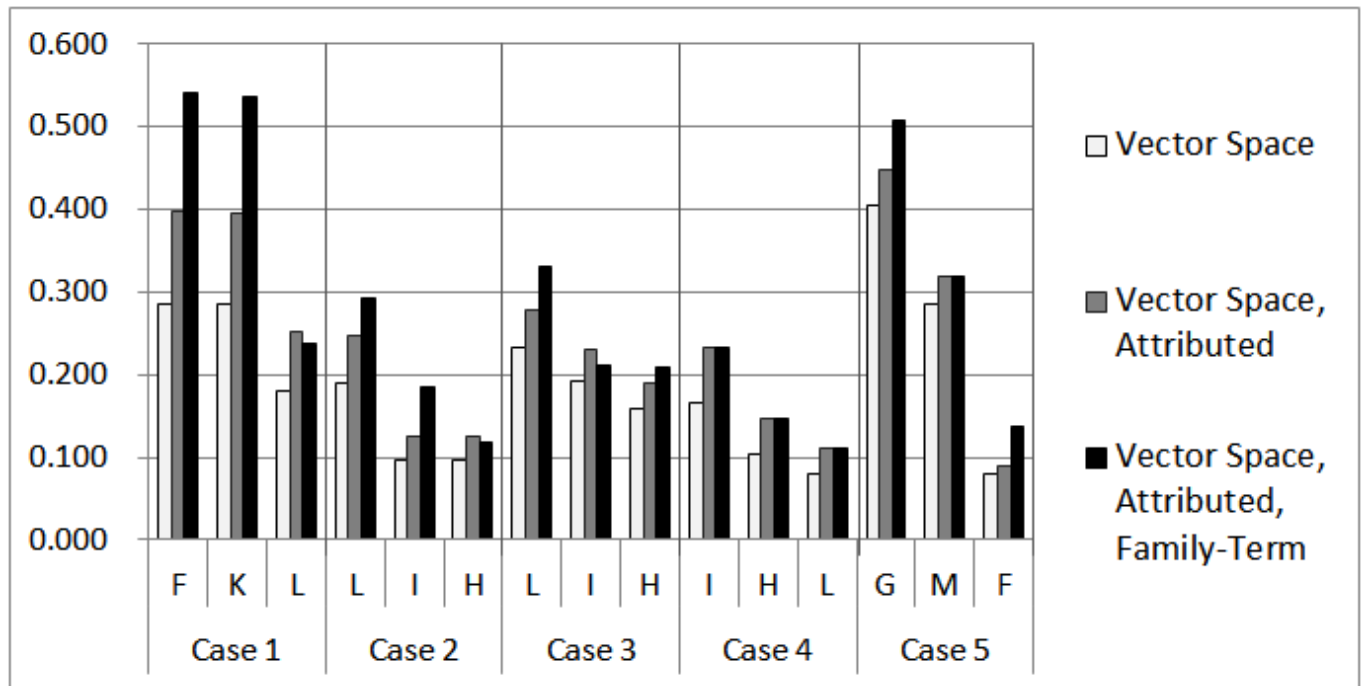
- ・ より多くのSysML図を対象とする
- ・ ベクトル空間モデルの更なる改良
特に“ファミリー単語”の扱い
- ・ 要求トレーサビリティ管理を支援するツールを開発する

要件 ⇔ 設計トレーサビリティ

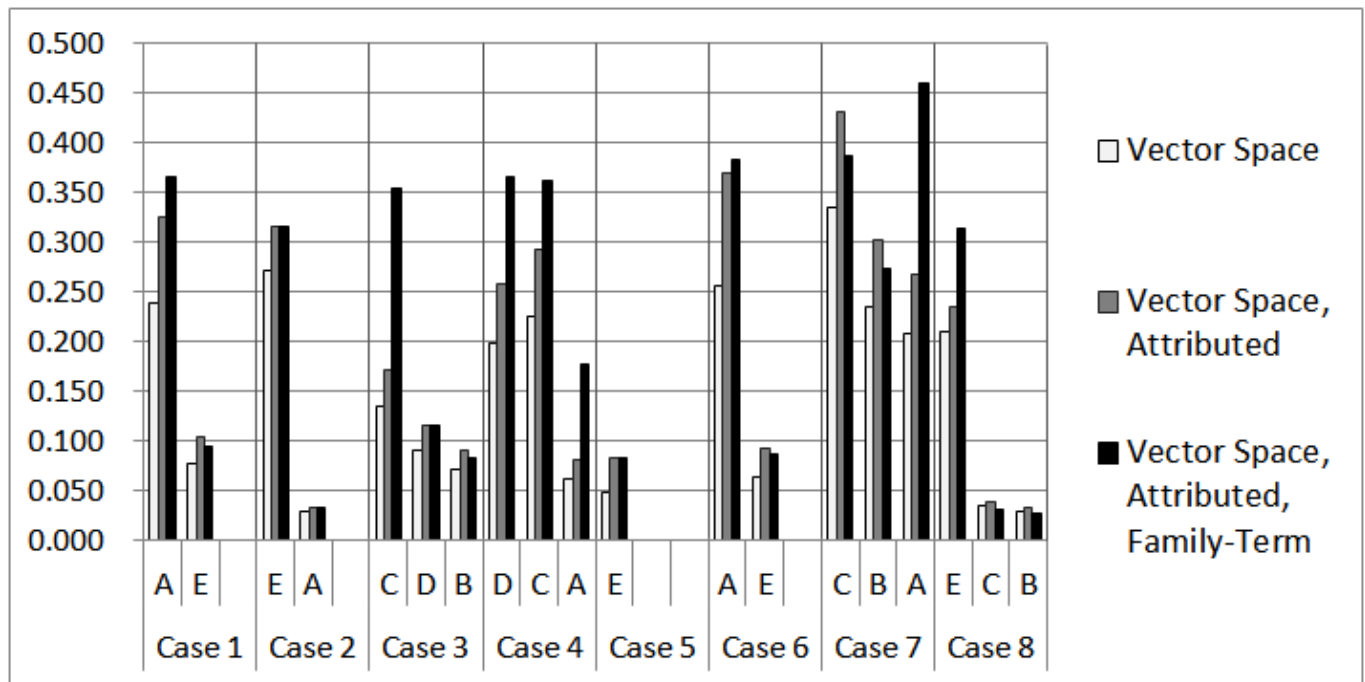


True-link \longleftrightarrow
Non-link \dashrightarrow

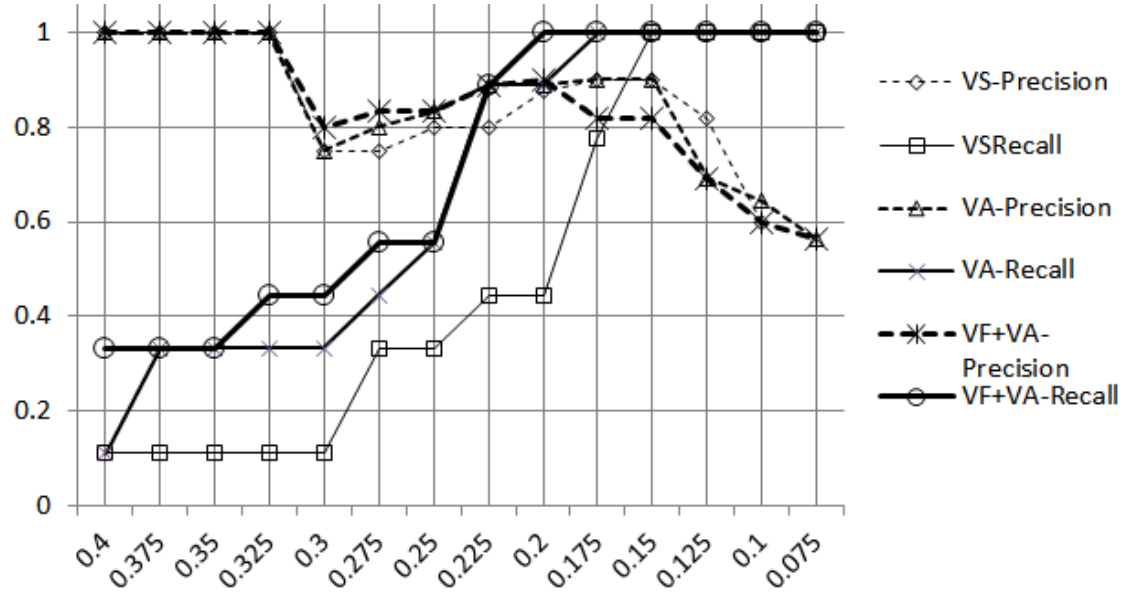
要件 ⇒ 設計



設計 ⇒ 要件



要件 ⇒ 設計



設計 ⇒ 要件

