

A Framework for Identifying and Analyzing Non-functional Requirements from Text

Vibhu Saujanya Sharma

Roshni R. Ramnani

Shubhashis Sengupta

TwinPeaks'2014 Proceedings of the 4th International Workshop on
Twin Peaks of Requirements and Architecture

Pages 1-8

目次

- 背景
- 既存研究
- 目的
- アプローチ
- **NFR**の分類手順
- 分類結果
- まとめ

背景

- 非機能要件(NFR)は，システムの動作と品質を保証する上で非常に重要である.
- NFRは早期の段階で見過ごされ，規定されないことが多く，ソフトウェア開発の後半工程に深刻な影響を与えることがある.

既存研究

- **NFR**の分析には、 4 つの工程が必要である^{[3][4]}
 1. 要件文から**NFR**を特定する
 2. 特定した**NFR**を項目ごとに分類
 3. **NFR**が完全性と網羅性を満たしていることを確認する.
 4. 分類した**NFR**が、他の分類と被っていないか確認する.

[3] L. Chung and J. do Prado Leite, “On non-functional requirements ...

[4] A. Casamayor, D. Godoy, and M. Campo, “Identification of non-functional requirements ...

目的

- 自然言語の要件文から，**NFR**を自動で検出し，分類するためのフレームワークを提示することを目的とする.

アプローチ

- **NFR**に属する用語を，自然言語のテキストから識別し，**NFR**のカテゴリを示す言語パターンを検出して分類を行う．
- 構文によるパターンの分類と，セマンティック(特定の意味を持つ)パターンの分類を用いて実験を行う．

アプローチ

(例文1)パスワードの長さは7~9文字にすること.



セキュリティ

(例文2)パスワード管理システムはモジュールBに実装すること.



セキュリティ？

アプローチ

(例文1)パスワードの長さは7~9文字にすること.



セキュリティ

(例文2)パスワード管理システムはモジュールBに実装すること.



セキュリティ？

構文によるパターン

- 構文のパターンは，キーワードによる分類方法.

(例) 「ラウンドトリップタイム」， 「許容レベル」， 「ダウンタイム」 など

セマンティックパターン

- セマンティックパターンは、ドメイン固有言語 (DLS) を用いて分類を行う。
- キーワードに関係する言葉が出てくるかどうか。

(例)<“timeunit”>は「時間」「分」「秒」などの言語集合を示す。

分類カテゴリ



Figure 1: A sample NFR hierarchy

検出ルール

R1 パターンが存在するもの
文の中に少なくとも1つのパターン(キーワードとそれ
属性ワード)が存在するもの

R2 順序付きのパターン
ある単語 p_1 が必ず単語 p_2 より先に来ているもの

R3 逐次パターン
ある単語 p_1 から p_n までが連続で出現するもの

検出ルール

R4 順序つきの一連のパターン

ある単語 p_1, p_2 と p_3, p_4 がこの順序で出現するもの

R5 リンクされたパターン

R1から**R4**のパターンが文中で複数出てくるもの

要件文の衝突問題

- **s1**：システムのダウンタイムは年間12時間未満にする必要があります。
- **s2**：システムの稼働率が99.99%を超えている必要があります。

s1とs2は同じことを言っているが、記述しているケースがある。

この問題の識別機能は追加を検討中

NFRの分類手順

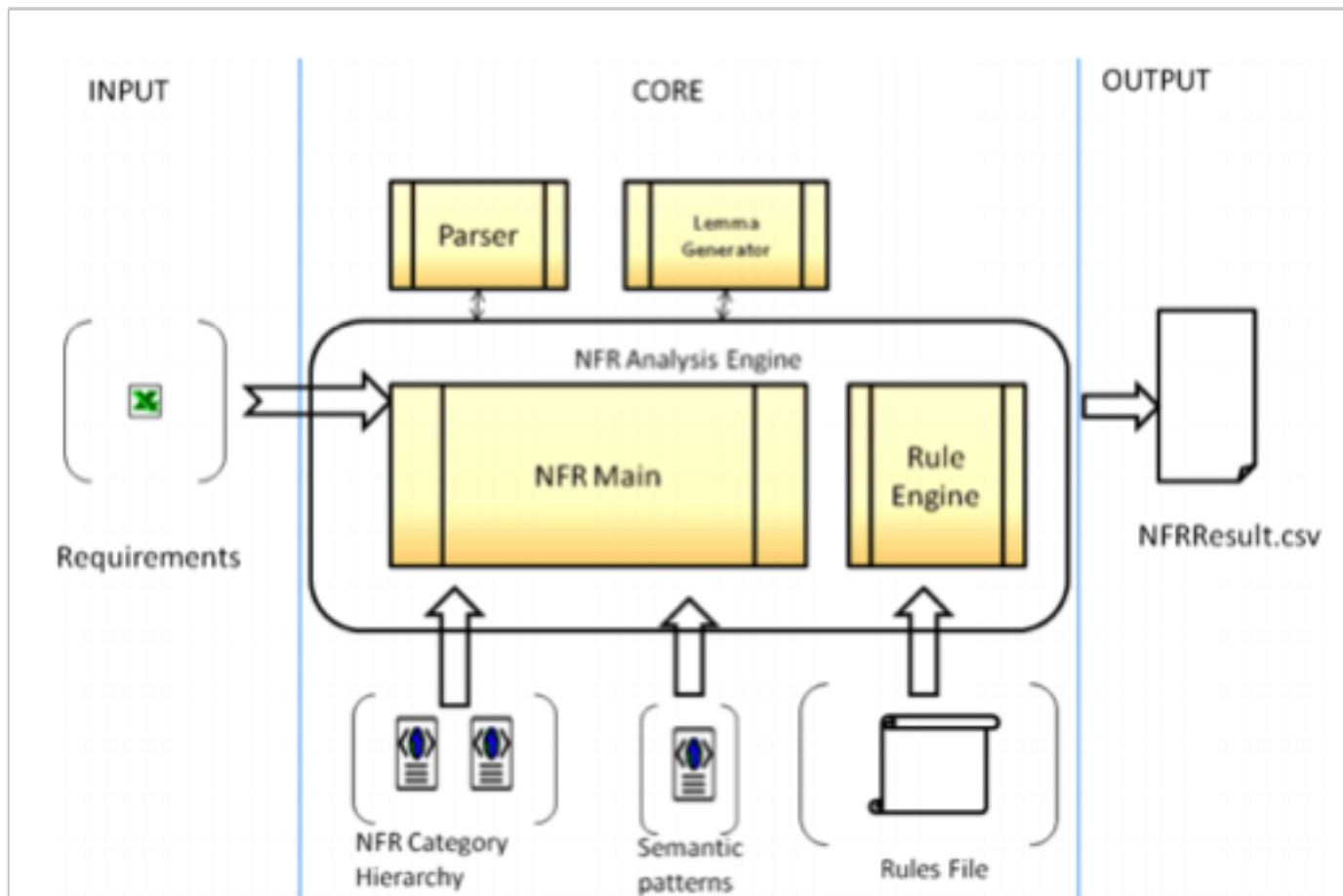


Figure 4: A schematic representation of the NFR detection prototype

分類するNFRの種類

Table 1: NFR Categories and Prototype Rules

S.No.	Category	Simple rules (only syntactic pattern type)	Complex rules (multiple pattern types & relationships)
1	Availability	32	14
2	Performance	65	20
3	Security	1450	4
4	Privacy	3	-
5	Usability	156	5
6	Compliance	32	3

Availability: 可用性

Performance: 性能性

Security: セキュリティ (Privacy: プライバシー)

Usability: 使用性

Compliance: コンプライアンス

の 5 つに分類を行う

分類に用いるデータ

- Promise リポジトリから要件(625個)を抽出して分類を行う。

Promise リポジトリであらかじめ分類されているNFR(11分類)を今回分類するNFR(5分類)に当てはめて実験を行う。

- A=Availability • L = Legal
- LF = Look and feel • MN = Maintainability
- O = Operational • PE = Performance
- SC = Scalability • SE = Security
- US = Usability • FT = Fault tolerance
- PO = Portability

分類結果

Table 2: Evaluation Results

Category Name	Requirement Count in Repository	Requirement Count Identified by the Prototype	Requirement Count Correctly Identified	Precision	Recall
Availability (A)	21	22	18	0.8181	0.8571
Performance (PE)	54	48	33	0.6875	0.6111
Security and Privacy (SE)	66	68	43	0.6324	0.6515
Usability (US, LF)	105	87	77	0.8850	0.7333
Compliance (L)	13	21	9	0.4286	0.6923

手作業での分類と提案手法での分類とで、手作業と同じくらいの各要件数を抽出でき、結果は良好であった。

手作業と提案手法とで、“Performance”と“Scalability”の分類の仕方に大きな違いが見受けられた。

分類結果

“Scalability”に属したものを手動で“Performance”にタグ付けした結果

Table 3: Results for Performance (Altered Classification)			
Category Name	# Requirements Identified by the Prototype	# Requirements Correctly Identified	Precision
Performance (PE)	48	44	0.9166

“Performance”の分類結果が向上した.

まとめ

- 2つのパターン(構文パターン, セマンティックパターン)を用いてNFRの分類を行った.
- 今回は分類する項目が少なかったため, 項目数を増やせば, さらに詳細な分類も可能であると考えられる.