3rd ISRITI 2020
IEEE INDONESIA SECTION
Conferences.id INDONESIAN CONFERENCE NETWORK
STMIK AKAKOM YOGYAKARTA
ITS Institut Teknologi Sepuluh Nopember

Extraction Dependency Based on Evolutionary Requirement Using Natural Language Processing

Institut Teknologi Sepuluh Nopember
Rakha Asyrofi, Daniel Oranova Siahaan, Yudi Priyadi

Index


Introduction

Preliminary studies

1. **Dahlsteid 2001** ,provides a continuation **framework** on the area of **interdependency requirements**.

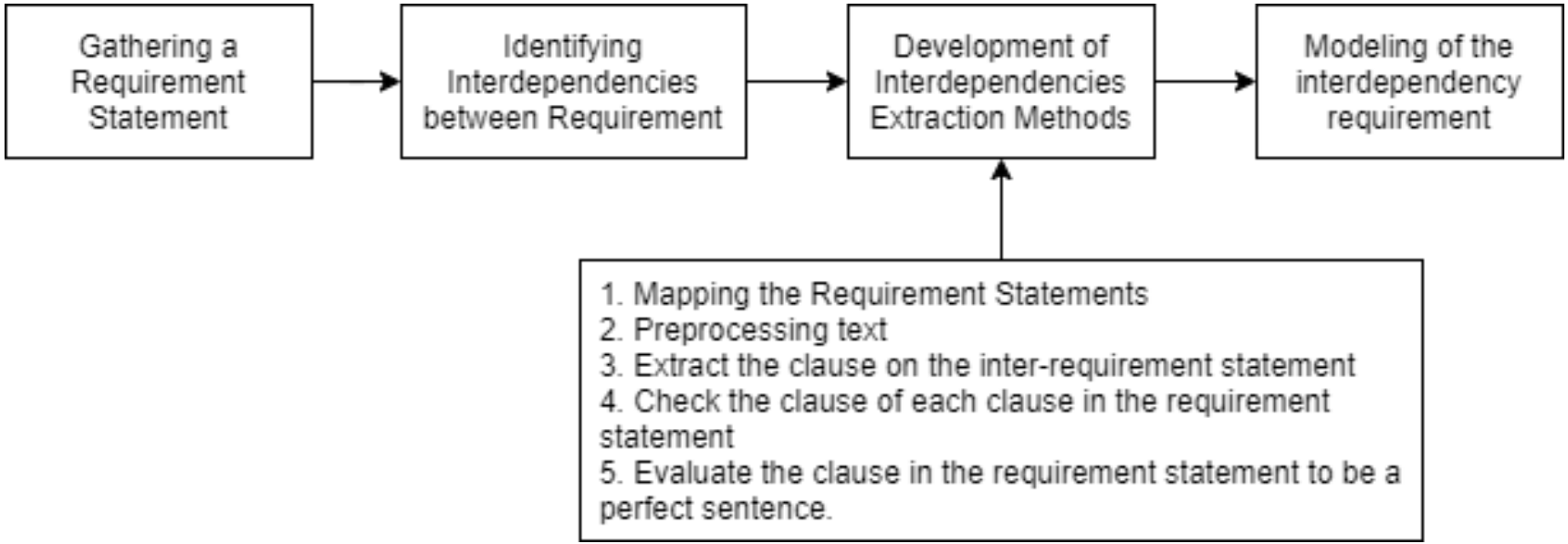
2. **Samorsir 2019**, introduced another method to build an **interdependency requirement model**. based on the association of relations between requirements and classes.

3. **Priyadi 2019**, introduced an **interdependency requirement graph** model from the SRS (Software Requirement Specification) document.

4. From the continuation of previous studies, the authors propose **the extraction of dependency** based on the requirement for conditional and evolutionary dependence with Natural Language Processing

Research Method

Research Method



Gathering Requirement Statement

This stage explains the collections of requirement statements contained in several SRS documents. There are two types of requirement statements in the SRS document, i.e., functional and non-functional requirements. The two requirements are used as a test dataset for identifying interdependencies between requirements in the SRS document.

Identifying Interdependencies between Requirement

Mechanisms for determining interdependencies requirement [17] are consists of: 1) How these **requirements** can **affect one to another**, 2) **Identify** the **interdependency requirements**, a paired analysis, and traceability relationship between those requirements [6].

1. **AND**: R1 requires R2 to function [18].
2. **OR**: R1 requires or does not R2 to operate [19]
3. **XOR**: another kind of dependency between R1 and R2, which can choose one of the two requirements to function [20].
4. **PART OF**, is used to describe the relationship between complex and simple requirements which are related to produce fractions of these complex requirements.

Development of Interdependencies Extraction Methods

1. Mapping the Requirement Statements
2. Preprocessing text
3. Extract the clause on the inter-requirement statement
4. Check the clause of each clause in the requirement statement
5. Evaluate the clause in the requirement statement to be a perfect sentence.

Mapping the Requirement Statements

Based on SRS documents, it is necessary to tabulate special requirements which consist of functional and non-functional. We do the extraction process based on the type of each document. The requirement statement map explains that the distribution of user requirement contains several columns, namely **ID** as a label indicating a functional or nonfunctional type, A requirement followed it in the form of the **requirement** contained in the description sentence, the functional requirement is used with a minimum sentence that consists of the subject and predicate while non-functional requirements in the form of a requirement of time service restrictions, process development, and other standards.

	ID	Requirement Statement	UC / Quality	Priority
0	F01	Users can create a new diary.	UC01	Must
1	F02	Users can add titles to the diary.	UC01, UC02	Must
2	F03	Users can add weather to the diary.	UC01, UC02	Optional
3	F04	Users can add dates to the diary.	UC01, UC02	Optional
4	F06	Users can add hours to the diary.	UC01, UC02	Optional
5	F07	Users can add seconds to the diary.	UC01, UC02	Optional
6	F08	Users can save diaries.	UC01, UC02	Must
7	F09	Users can add photos to the diary.	UC01, UC02	Must
8	F10	Users can read the diary that has been created.	UC02, UC03	Must
9	F11	Users can share diaries in the form of postcards.	UC02	Must
10	F12	Users can save postcards.	UC02	Optional
11	F13	Users can change the postcard background color.	UC02	Optional
12	F14	Users can change the color of posts on the postcard.	UC02	Optional
13	F15	Users can delete the diary.	UC02	Must
14	F16	Users can edit the diary that has been created.	UC02	Must
15	F17	Users can open a calendar that contains a diary.	UC03	Must
16	F18	The system can open a diary editor.	UC01, UC02	Must
17	NF01	The application has a high level of availability and can operate continuously for seven days per week and 24 hours per day without stopping.	Reliability, Survivability	Must
18	NF02	Application must have a high degree of flexibility, which must be able to run on devices using the Kit-kat version of the Android operating system and above	Portability, Flexibility	Must
19	NF03	This application must have a high level of integrity for data security by including security in the form of a password for the user.	Integrity	Must
20	NF04	This application has a high usability aspect and is easy to use through an interface so that after the application is introduced for two hours	Usability	Must
21	NF05	This application must have a response time value that is quite good and fast when accessed. The most extended performance result that can be tolerated is three seconds.	Performance	Must
22	NF06	Applications must have a high level of interoperability, because one of the needs in this application is to connect with social media such as Line, instagram and facebook	Interoperability	Must
23	NF07	Documentation must be included in the system source code so that developers can reread when changes or additions to the source code will be made and have a high level of maintainability.	Maintainability	Must
24	NF08	The application must have a high degree of flexibility, which can be used by users aged seven years and above	Flexibility	Optional
25	NF09	The modules in the system will be designed and programmed in a structured way so that they can continue to be used if there is a system development	Reusability, Maintainability	Must

Preprocessing text

Text preprocessing is implementing on every software requirement. At the text preparation stage, there are four main processes. The following are examples of the pre-prepared results on software requirements.

1. **Sentence Splitting**: By using sent tokenize, this function wraps a sentence to become a one-dimensional array so that it can be useful for other processes
2. **Tokenization**: Tokenizing has the role of breaking the text into the smallest units in language processing or called word pieces in the text documents
3. **POS Tagging**: This process carries out to find out the categorization of word classes, such as nouns, verbs, adjectives, etc
4. **Parsing**: This process evaluates a sentence with references by [21] How it is made based on structure, whether it consists of the single or compound sentence by seeing the sign "S" as a unit sentence

```
(ROOT
 (S
  (NP (NN Documentation))
  (VP (MD must)
   (VP (VB be)
    (VP (VBN included)
     (PP (IN in)
      (NP (DT the) (NN system) (NN source) (NN code)))
     (SBAR (IN so) (IN that)
      (S
       (NP (NNS developers))
       (VP (MD can)
        (VP (VB reread)
         (SBAR
          (WHADVP (WRB when))
          (S
           (NP
            (NP (NNS changes)
             (CC or)
             (NNS additions))
            (PP (TO to)
             (NP (DT the) (NN source) (NN code))))
           (VP
            (VP (MD will)
             (VP (VB be)
              (VP (VBN made))))
            (CC and)
            (VP (VBP have)
             (NP
              (NP (DT a) (JJ high) (NN level))
              (PP (IN of)
               (NP (NN maintainability))))))))))))))
  (. .)))
```

Extract the clause on the inter-requirement statement

The next step is to form a clause in the statement between requirement by the minimum sentence consisting of criteria divided into **subjects, predicates, and objects**.

```
(ROOT
 (S
  (NP (NN Documentation))
  (VP (MD must)
   (VP (VB be)
    (VP (VBN included)
     (NP (DT the) (NN system) (NN source) (NN
code)))))))
```

Check the clause of each clause in the requirement statement

This process is carried out to see whether a clause in the requirement still has a term or not

```
(K1a
 (S
  (NP (NN Documentation))
  (VP (MD must)
   (VP (VB be)
    (VP (VBN included)
     (NP (DT the) (NN system) (NN source) (NN
code)))))))
(k1b
 (NP (NNS developers))
 (VP (MD can)
  (VP (VB reread)
   (NP (DT the) (NN source) (NN code)))))))

(k1c-1a
 (NP (NNS changes)
  (PP (TO to)
   (NP (DT the) (NN source) (NN code))))
  (VP
   (VP (MD will)
    (VP (VB be)
     (VP (VBN made))))
```

```
(k1c-1b :
 (NP (NNS changes)
  (PP (TO to)
   (NP (DT the) (NN source) (NN code))))
  (VP (VBP have)
  (NP
   (NP (DT a) (JJ high) (NN level))
   (PP (IN of)
    (NP (NN maintainability)))))))))))))))

(k1c-2a :
(NNS additions))
(PP (TO to)
    (NP (DT the) (NN source) (NN code))))
     (VP
      (VP (MD will)
       (VP (VB be)
        (VP (VBN made))))

(k1c-2b :
(NNS additions))
(PP (TO to)
    (NP (DT the) (NN source) (NN code))))
    (VP (VBP have)
     (NP
     (NP (DT a) (JJ high) (NN level))
     (PP (IN of)
      (NP (NN maintainability)))))))))))))))
```

Evaluate the clause in the requirement statement to be a perfect sentence.

This process is carried out to evaluate each clause in the requirement statement following perfect sentences or not. If it is not appropriate, then it needs to be re-evaluated. In this case, **NF07** has six clauses consisting of **K1a, K1b, K1c-1a, K1c-1b, K1c-2a, K1c-2b** These results are following reference by [22] to minimize ambiguity in a sentence.

K1a = Documentation must be included the system source code
K1b = Developers can reread the source code.
K1c-1a = Changes to the source code will be made
K1c-1b = Changes to the source code have high level of maintainability
K1c-2a = Additions to the source code will be made.
k1c-2b = Additions to the source code have a high level of maintainability

Modeling of Interdependency Requirement

This step is presented with a stage, which describes all the activities carried out in the research proposal, based on the research contained in the previous section. The next stage is made to realize the framework that has been designed in this study.

1. Analyze software development projects in ITS & public repositories in the Department of Information as a dataset
2. Make a form an Interdependency Model between requirement.
3. Graph union is a hypothesis as a method used to determine the interdependency requirement graph.
4. In this section, an explanation of each stage of the research process is carried out based on functional & non-functional requirements.
5. As preliminary preparation, there is an example of defining the association extraction from SRS, which staging a requirement for the type of interdependencies.

TABLE 2
FUNCTIONAL & NON-FUNCTIONAL REQUIREMENT ASSOCIATION EXTRATION

Association Extraction	*Requirement Statement (RS)*	*Dependency Type*
AND	F01: Users can create a new diary FO2: Users can add titles to the diary F03: Users can add weiather to the diary	RS1 REQUIRE RS2 AND RS3
OR	F01: Supervisor can see item data F02: Supervisor can add item data. F03: supervisor can update item data	RS1 REQUIRE RS2 OR RS3
XOR	F18: The system can open a diary editor F01: Users create a new diary F15: Users can delete the diary	RS1 REQUIRE RS2 XOR RS3
PART-OF	NF07: Documentation must be included in the source code system. So that developers can reread when changes or additions are made to the source cde has a high level of maintainability. Splitting to NF07a, NF07b, NF07c-1a, NF07C-1b, NF07c-2a, NF07c-2b	RS1A, RS1B, RS1C PART OF RS1

Conclusion

Conclusion

It can be concluded that with those methods of development models, interdependency models can be obtained among requirement attained, which are F01 REQUIRE F02 AND F03 on SRS2 document, F01 REQUIRE F02 OR F03 on SRS8 document, F18 REQUIRE F01 XOR F15 on SRS2 document, and also NF07A, NF07B, NF07C PART OF NF07 on SRS2 document.

Thank You

For Your Attention