NNGT Int. J. on Software Engineering, Vol. 2, Feb 2015

A Semi-Automated Generation of Activity Diagrams from Arabic User Requirements

Ibrahim N. Nassar
Department of Graduate Studies
Palestine Polytechnic University
Hebron, Palestine
Nassar.ibrahim@hotmail.com

**Faisal T. Khamayseh*
Department of Graduate Studies
Palestine Polytechnic University
Hebron, Palestine
faisal@ppu.edu

**Corresponding Author*

Abstract— **Requirement Engineering is currently one of the main directions in software research and applications. The traditional design methods of creating UML models have a high cost and long time. Activity diagram is an important UML model to describe the dynamic behavior of the system. Few software tools tried to achieve this part depending mainly on some constraints and rules stated by software engineers. This paper proposes a semi-automated generating method to generate the activity diagram from Arabic user requirement using MADA+TOKAN tagger. Since there is a lack of research serving Arabic user requirements, we rush towards working on taking requirements written in Arabic language as input and transform them to activity diagram with less intervention. Our approach is to lessen the dependence on human intervention as possible. This research aims to help software engineers in the analysis phase by reducing cost and time required in performing these manual processes and activities.**

***Keywords*- Activity diagram, Arabic user requirements, Natural language processing tool, taggers.**

I. Introduction

Automated software engineering is one of the most important research problems of Software Engineering [1], it can improve the quality and productivity of software development, and less cost and time of analysis of user requirements. Automated software engineering tools and methods using in a huge area of applications. UML models are one of the most significant parts of software engineering to use the automated software tools and techniques.

Activity diagrams are graphical representations for the sequences of activities [2]. It represents the dynamics of the system and model business processes. It describes how the activities are coordinated to provide how the system achieves the operations. Since there is no direct automation procedure from Arabic requirements to activity diagram, we propose a way to generate Activity diagram from user requirements with least interventions.

The contribution of this paper is to present a high-level algorithm illustrating the detailed steps of generating activity diagram and a high-level contextual view on the elements of the activity diagram and thus objectively from Arabic user requirements.

The main motivation is to come up with a cross activity to from Arabic requirements to activity diagram avoiding of the traditional methods of creating sets of UML models hence saving time and cost.

Our procedure requires an Arabic Natural Language Processing tool to extract the elements of an activity diagram from Arabic user requirements such as MADA+TOKAN, which is used in this paper to split and tokenize the Arabic user requirement.

This paper is organized as follows: Section 2 presents the related work. Section 3 presents the methodology. Conclusion is represented in section 4.

II. Related Work

Set of current research papers proposing methods for generating UML models from requirements engineering. Some of them suggests algorithms to auto-generating some UML models such as use case, sequence and class diagrams. Other papers proposed tools to do the desired tasks.

Yue, Briand, and Labiche [3] proposed an automated technique to generate activity diagrams from use cases. This approach implemented in aToucan tool. aToucan involves three steps. First, requirements engineers defined use cases manually using Restricted Use Case Modeling (RUCM) approach. The result is a textual use case model (UCModel) that expressed in a restricted natural language. Second, aToucan reads textual UCSs to identify POS and grammatical relation dependencies of sentences. Then, it records the information into an instance of the UCMeta that is the intermediate model in aToucan used to bridge the gap between a textual UCMod and a UML analysis model. Finally, it transforms the instance of UCMeta into activity diagram based on many transformation rules.

Yasser Khan and Mohamed El-Attar [4] proposed an automated approach to generating activity diagram from Use Case Maps (UCMs). They implemented a model transformation from UCM notation to activity diagram notation. This method does not automate generating activity from simple requirements; rather it works only on

DOI : 02.IJSE.2015.1.6

straightforward mapping the use cases to counterpart activity symbols.

Rodriguez, Nebut, Cuaresma, and Risoto [5] proposed an automated method to auto-generate activity diagram of each use case written in a specific format. The generation is performed by a model transformation, taking use case textual scenario as an input and producing the corresponding activity sub-diagram. The transformation is defined using QVT-relational language and implemented in Java as a prototype tool.

Nabil Arman and Sari Jabbarin [6] proposed a semi-automated approach to generate use case models from Arabic user requirements using Natural language processing. A set of steps represents their approach for constructing a use case model presented. Set of heuristics are presented to obtain use cases. These heuristics use the tokens produced by a natural language processing tool, namely Stanford parser.

Commercial auto generator tools generate activity diagrams from use cases. There exist three closed tools: Visual Paradigm, CaseComplete and Ravenflow[4] to perform generating process. Visual Paradigm and CaseComplete can transform the flows of events of a use case into an activity diagram. Each flow of events needs to be structured using a simple use case template including basic flows and extension sub use cases. Ravenflow requires a set of guidelines are proposed to generate the activity diagrams. None of them use free written requirements as an input.

III. Overview

In this section, we present a brief description of the activity diagram and MADA+TOKAN.

A. Activity Diagram

Activity diagram [2] provides a complete view of how the system is working by describing the sequence of the activities. This diagram is suitable for building and illustrating business processes and modeling the procedural flow of activities. Activity diagrams can show concurrent, parallel and alternate flows.

Elements of the activity diagram are:

- *Initial node:* indicates the first node of an activity diagram, it's shown using a small solid filled circle. It must be only one initial node drawn in an Activity diagram. It should be connected only to one Activity node.
- *Final node:* shows the end of a workflow of an activity diagram.
- *Activity node:* it represents the operations of the system.
- *Control Flow:* is an arrow connects two nodes.
- *Decision Node:* has one input and two or more outputs, because it is a conditional sentence, the choices are Yes or No. This node directs the flow to other elements.
- *Fork Node:* splits a flow into multiple concurrent flows.
  - *Join Node:* synchronizes multiple flows.

B. MADA+TOKAN Tagger

MADA+TOKAN [7] is a tagger that reads Arabic text only and assigns tags for each word, such as noun, verb, adjective, pronoun, etc. Using the online tool http://cri_nlp.kacst.edu.sa/nlp, we can insert the Arabic requirements into the specified place and choose the tagging choice (third choice) in the tool then click the button (text analysis. MADA+TOKAN is a software that can obtain wide morphological and contextual information from Arabic requirement. Internally, MADA depends on Buckwalter Arabic Morphological Analyzer (BAMA), SRILM toolkit, and SVM Tools. MADA uses the SVM Tools to operate its SVM. MADA checks a list of possible analyzes for each word generated by BAMA, then selects the best analysis using support vector machine (SVM). MADA+TOKAN trained on Penn Arabic Treebank (ATB). MADA has over 96% accuracy on basic morphological choice. TOKAN is a general tokenizer for Arabic, provides the information for tokenizing MADA into set of a possible tokenization. TOKAN separates conjunctions, prepositions, verbal particles, the definite article and pronominal. MADA+TOKAN uses a set of tags to describe the Arabic words in the statements. Table 1 shows some tags for MADA+TOKAN method

Table 1: Tags of MADA+TOKAN

Part of Speech Definition	Tags
Verb	VBD, VBN, VBP
Noun	DTNN, NN, DTNNS, NNS
Subordinating Conjunction or Preposition	IN
Proper Noun	DTNNP, NNP, DTNNPS, NNPS
Pronoun	PRP, PRP$, WP, DT
Coordinating Conjunction	CC
Adjective	DTJJ, JJ
Adverb	RB, WRB
Interjection	UH
Punctuation	PUNC
Cardinal Number	DTCD, CD
Particle	RP
Foreign Word	FW

IV. Activity DIAGRAMS CONSTRUCTION APPROACH

In this section, we discuss our approach of the activity diagram extraction in detail, also the detailed algorithm to implement our approach. We describe how MADA+TOKAN parser used to splitting and tokenizing the Arabic text. As a step of construction, we proposed limited grammar rules to

DOI : 02.IJSE.2015.1.6

achieve the goal and generate all elements of the activity diagram.

A. Elements of Activity Diagram Extraction

Initial node is the beginning of the diagram; it is connected to the next element which is activity node. Final node is the last element of an activity diagram; it is connected with the previous element.

Activity diagram elements (activity node, a decision node, control flow) are constructed from Arabic Text based on a type of the sentences.

Activity node represents the verbal sentences. Verbal sentences consist of verb, subject, and object. We present the grammars rule for verbal sentences [8-10].

1. Verbal Sentence → Verb + subject + object
2. Verb → Past | Present | Imperative mood
3. Subject → Noun | Pronoun
4. Noun → Proper noun | demonstrative | Relative
5. Object → Noun | Quasi Sentence
6. Quasi Sentence → Preposition phrase | Adverb phrase
7. Preposition Phrase → Preposition + Noun
8. Adverb Phrase → Adverb + Noun

Verbal sentences can be one of the following common structure:

1. Verb + Subject
Verb + (Noun | Pronoun)

2. Verb + Object
Verb + Noun
Verb + Preposition + Noun
Verb + Adverb + Noun

3. Verb + Subject + Object
Verb + (Noun | Pronoun) + Noun
Verb + (Noun | Pronoun) + Preposition + Noun
Verb + (Noun | Pronoun) + Adverb + Noun

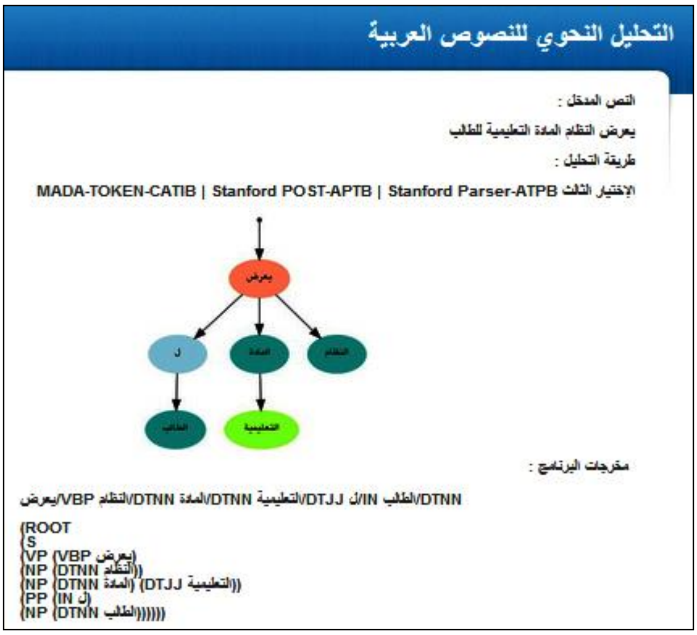
4. Subject + Verb
(Noun | Pronoun) + Verb

5. Subject + Verb + Object
(Noun | Pronoun) + Verb + Noun
(Noun | Pronoun) + Verb + Preposition + Noun
(Noun | Pronoun) + Verb + Adverb + Noun

Object may be more than one in the same sentence. Relying on the structure of verbal sentences, we propose a set of common grammar rules:

1. Verb + Noun: (Example: يحلل النظام)
2. Verb + Noun + Adjective: (Example: يستقبل صورة معينة)
3. Verb + Noun +Noun (Example: يحلل النظام صورة)
4. Verb + Noun + Noun + Adjective (Example: يستقبل النظام صورة معينة)
5. Verb + Noun + preposition + Noun + Noun + preposition + Noun (Example: يعمل النظام على تحديد مربع في الصورة)
6. Verb + Noun + preposition + Noun + Noun (Example يعمل النظام على تحديد المربع)
7. Verb + Noun + preposition + Noun (example: يظهر النتيجة على الشاشة)
8. Verb + Noun + Noun + adjective+ preposition + Noun (example: يعرض النظام المادة التعليمية للطالب)
9. Verb + Noun + Noun + preposition + Noun (example: يعرض النظام المادة للطالب)
10. Verb + Noun + Noun + Pronoun + Noun (example : يسحب المستخدم مبلغ من المال)
11. Verb + Noun + preposition + Noun + Noun (example: يقوم السائق بتسجيل الدخول)

Required tags are obtained from MADA+TOKAN tagger. Figure 1 shows an example of Arabic text " يعرض النظام المادة التعليمية للطالب ", Analysis method: MADA+TOKAN, and the outcomes of analysis: يعرض/VBP النظام/DTNN المادة/DTNN التعليمية/DTJJ ل /IN الطالب/DTNN.



التحليل النحوي للنصوص العربية

النص المدخل :

يعرض النظام المادة التعليمية للطالب

طريقة التحليل :

MADA-TOKEN-CATIB | Stanford POST-APTB | Stanford Parser-ATPB الإختيار الثالث

مخرجات البرنامج :

يعرض/VBP النظام/DTNN المادة/DTNN التعليمية/DTJJ ل/IN الطالب/DTNN

```
(ROOT
(S
(VP (VBP يعرض)
(NP (DTNN النظام))
(NP (DTNN المادة) (DTJJ التعليمية))
(PP (IN ل)
(NP (DTNN الطالب))))))
```

Figure 1: MADA+TOKAN Tagger Analysis of Arabic Statement

The proposed grammar rules of verbal sentences is a new formatting method using tags of MADA+TOKAN. We match the common grammar rules with tags that mentioned in table1.

To identify the decision node; the conditional sentences represent the decision nodes in activity diagrams. The structure of conditional sentences [11] consists of:
Conditional Particle + Conditional sentence + Answer Particle + Conditional Answer.

There are two common particles (condition particles) in Arabic (لو ، إذا). Their tags are RB and NNP respectively. Conditional sentence is a verbal sentence. We should consider

© N&N Global Technology 2015
DOI : 02.IJSE.2015.1.6

the previous analysis of verbal sentences that represents the activity node.

The *answer* particle is an adverb for the condition *answer*; which is optional. The answer particles in Arabic are: (فان، سوف، فسوف حرف اللام). The tags for them are:

- NNP: فان
- CC+RP: فسوف
- IN + NN: حرف اللام + الاسم المجرور
- RP: سوف

Condition *answer* is the same condition sentence. Decision node has two directions based on a type of the statement. We therefore have two types: Proof sentence and Negation sentence. The proof sentence is the sentence that has the answer is (yes). The negation sentence has the negation particles, and the answer is negative. Following example illustrates the common proof sentences:

1. Cond. Particle + verbal sentence + Answer Particle + verbal sentence (اذا وجد مربع فان النظام يحلل الصورة)
2. Cond. Particle + verbal sentence + verbal sentence (اذا نجح الطالب تعرض له مادة جديدة

The following example illustrates the common Negation statements:

1. Cond. Particle + Negation particle + verbal sentence + Answer Particle + verbal sentence
   Example: اذا لم يجد وجه فان النظام يعرض رسالة

2. Cond. Particle + Negation particle + Verb + verbal sentence
   Example**:** اذا لم ينجح يقدم امتحان قصير

Arrow node is used to identify the connectors in the activity diagram. The connector between initial node and activity node is arrow. The connector between last nodes (activity nodes) and finial node is arrow.

According to the decision node, there are two directions. If the sentence is the proof part, the connector is an arrow that connects the next element directly. If the sentence is Negation, the connector is an arrow that connects an activity node or connects an entryway of the same decision node. The connectors for proven sentence are RP, CC+RP, NNP or IN+NN.

If the connector is the word ثم, then the tag is RB. It separates between two verbal sentences or verbal sentence and noun.

Example:

1. يقرأ الكتاب ثم يقدم امتحان.
2. يقرأ الكتاب ثم القصة.

If the connectors are (ف/ و/ أو), the tag for them is CC. They are separated between two verbal sentences or verbal sentence and noun. If the word that precedes (ف/ و/ أو) is noun or adjective, or both of them, and the next word, is noun or adjective, or both of them, then, in this case, we don't separate them.

Examples:

1. يقرأ المادة التعليمية والتوجيهية
2. يقرأ المادة التعليمية فالتوجيهية
3. يقرأ المادة التعليمية أو التوجيهية

But if the next word is a verb after CC, We check the next statement, if it begins with a conditional particle اذا، لو then CC is a decision connector. If not, CC separates between two verbal statements.

Examples:

1. يقرأ المادة التعليمية و يقدم امتحان
2. يقرأ المادة التعليمية فيقدم امتحان
3. يقرأ المادة التعليمية أو يقدم امتحان. اذا قرأ المادة التعليمية ونجح في الامتحان تظهر له مادة جديدة أما اذا قدم امتحان ....

B. Algorithm for Activity Diagram extraction

In this section, we present the high-level algorithm to implement our approach.

Input: Arabic Requirements
Output: Activity Diagram

1. Generate one Initial Node to show the first node of the activity diagram.
2. Parse each sentences using MADA+TOKAN.
3. For each word in the sentence, there exist a tag. Connect the tags together.
4. To identify the activity node, we check the tagged sentence if it contains one of the following grammars:
   - Verb + Subject :
     VP (VBP|VBD|VBN) + NP (DTNN|NN|NNS|DTTNS)

   - Verb + Object
     VP (VBP|VBD|VBN) + NP (DTNN|NN|NNS|DTTNS) | PP+NP (IN+DTNN) | NP+NP (NN|NNP + DTNN|NN)

   - Verb + Subject + Object
     VP (VBP|VBD|VBN) + NP (DTNN|NN|NNS|DTTNS) + NP (DTNN|NN|NNS|DTTNS) | PP+NP (IN+DTNN) | NP+NP (NN|NNP + DTNN|NN)

   - Subject + Verb
     NP (DTNN|NN|NNS|DTTNS) + VP (VBP|VBD|VBN)

   - Subject + Verb + Object
     NP (DTNN|NN|NNS|DTTNS) + VP (VBP|VBD|VBN) + NP (DTNN|NN|NNS|DTTNS) | PP+NP (IN+DTNN) | NP+NP (NN|NNP + DTNN|NN)
5. To identify the decision node, we check the tagged sentence if it contains one of the following grammars:
   - Conditional Particle + Conditional sentence + Answer Particle + Conditional Answer.
     (RB | NNP) + Verbal sentence + (NNP| CC+RP | RP| IN+NN) + verbal statement

   - Conditional Particle + Conditional sentence + Conditional Answer.
     (RB | NNP) + Verbal sentence + verbal statement

   - Cond. Particle + Negation particle + verbal sentence + Answer Particle + verbal sentence
     (RB | NNP) + PRT + verbal sentence + (NNP| CC+RP | RP| IN+NN) + verbal statement
   - Cond. Particle + Negation particle + Verb + verbal sentence

DOI : 02.IJSE.2015.1.6

(RB | NNP) + PRT + verbal + verbal statement

6. The connectors between elements based on the conjunctions (RB, CC):
   - The connector between initial node and activity node is arrow.
   - The connector between last nodes (activity nodes) and Finial node is arrow.
   - If the connector is RB, it is an arrow separating the two verbal sentences or verbal sentence and noun.
   - If the connector is CC, it is an arrow separating two verbal sentences or verbal sentence and noun.
   - If the word that precedes CC is noun or adjective or both of them, and the next word is noun or adjective or both of them, then we don't separate them.
   - If the next word is Verb after CC, we check the next sentence: if it begins with RB or NNP then CC is a decision connector. If not, CC separates between two verbal statements
7. Use Final node to show the end of an activity.

V. CONCLUSIONS

This paper has presented the details of extracting the elements of the activity diagram and high-level algorithm to be generated from Arabic user requirements using natural language processing tool, namely MADA+TOKAN. We presented the basic grammars for verbal and conditional sentences. We used common grammar rules of verbal and conditional sentences to obtain the activity and decision nodes. This paper contributes to help software engineers in the analysis phase of the object-oriented software development specifically in semi-automated generation of activity diagram hence reducing cost and time required for manual processes and activities.

ACKNOWLEDGMENT

We would like to thank Dr. Nabil Arman, Dr. Diya Abu Zeina, and Khalid Daghameen from Palestine Polytechnic University (PPU) for their useful notes and feedbacks.

REFERENCES

[1] Torkar R. (2006). Towards Automated Software Testing - Techniques, Classifications and Frameworks. Karlskrona, Sweden: Blekinge Institute of Technology.

[2] Alhir, Sinan Si (2003). Activity Diagrams. Learning UML (pp. 156 - 164). Sebastopol, CA.: O'Reilly.

[3] Yue, T, Briand L. C. and Labiche Y. (2010). An Automated Approach to Transform Use Cases into Activity Diagrams. Proceeding of ECMFA'10 Proceedings of the 6th European conference on Modeling Foundations and Applications, pp. 337-353.

[4] Khan, Y. A. and El-Attar M. (2012). Automated Transformation of Use Case Maps to UML Activity Diagrams. ICSOFT 2012, pp. 184-189.

[5] Rodriguez J. G., Nebut C., Cuaresma M. E., Risoto M. M. and Roman I. R. (2008). Visualization of Use Cases Through Automatically Generated Activity Diagrams. MoDELS'08: 11th International Conference on Model Driven Engineering Languages and Systems, Toulouse, France. ACM/IEEE, Model Driven Engineering Languages and Systems (5301), pp.83-96.

[6] Arman N. and Jabbarin S. (2014). Generating Use Case Models from Arabic User Requirements in a Semiautomated Approach Using a Natural Language Processing Tool. Journal of Intelligent Systems.

[7] Nizar Habash, Owen Rambow and Ryan Roth (2010). MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. Center for Computational Learning Systems, Columbia University, USA.

[8] Kevin Daimi (2001). Identifying Syntactic Ambiguities in Single-Parse Arabic Sentence. Computers and the Humanities 35: 333–349.

[9] أبو المكارم، ع. (2007). الجملة الفعلية. القاهرة - مؤسسة المختار للنشر والتوزيع

[10] Karin C. Ryding, (2005). A Reference Grammar of Modern Standard Arabic, (Reference Grammars) Paperback, Cambridge University Pres.

[11] أبو المكارم، ع. (2007). التراكيب الاسنادية (الجمل: الظرفية - الوصفية - الشرطية)
القاهرة - مؤسسة المختار للنشر والتوزيع

DOI : 02.IJSE.2015.1.6