

# Generating Sequence Diagrams from Arabic User Requirements using MADA+TOKAN Tool

Nermeen Alami, Nabil Arman, and Faisal Khamayseh

Department of Computer Science, Palestine Polytechnic University, Palestine

**Abstract:** A new semi-automated approach for generating sequence diagrams from Arabic user requirements is presented. In this novel approach, the Arabic user requirements are parsed using a natural language processing tool called MADA+TOKAN to generate the Part Of Speech (POS) tags of the parsed user requirements, then a set of heuristics are applied on the resulted tags to obtain the sequence diagram components; objects, messages and work flow transitions (messages). The generated sequence diagram is expressed using Extensible Markup Language (XMI) to be drawn using sequence diagrams drawing tools. Our approach achieves better results than students in generating sequence diagrams. It also has better accuracy in generating the participants and less accuracy in generating messages exchanged between participants. The proposed approach is validated using a set of experiments involving a set of real cases evaluated by a group of software engineers and a group of graduate students who are familiar with sequence diagrams.

**Keywords:** UML, automated software engineering, sequence diagram, Arabic user requirements.

Received May 18, 2017; accepted December 6, 2018

<https://doi.org/10.34028/iajit/17/1/8>

## 1. Introduction

The sequence diagram shows how processes in a software system interact with each other based on time. In analysis phase, it is used to illustrate the objects that participate in the use case and the messages passed between them over time while in design phase the sequence diagram is used to distribute the use case behaviour to classes [5, 6].

This paper addresses the problem of generating sequence diagram from a set of user requirements, written in Arabic, in a semi-automated approach using a natural language processing tool, namely MADA+TOKAN. The process of transforming user requirements into the Unified Modelling Language (UML) diagrams is normally performed by human analysts. This process is time-and-money consuming and error prone because user requirements are usually written in a natural language and some analysts may be unfamiliar with sequence diagrams. The human analyst may make mistakes during reading large number of natural language user requirements, which may produce an incorrect model. In addition, if a change of a model is needed, a great effort, money and time are wasted during the modification process in order to accommodate the needed changes. So, the need for automated approaches is of great value to software engineers and software engineering community [2, 9]. The new approach starts with identifying the participants, receivers and the messages from the Arabic requirements and generates the sequence diagram [1].

The rest of the paper is organized as follows: section 2 presents the literature review and related works.

Section 3 describes the research methodology used in generating the sequence diagram models from Arabic user requirements. Section 4 presents the implementation and the validation of our proposed approach, and finally, sections 5 and 6 present the discussion, conclusions and future works.

## 2. Related Works

Generating sequence diagrams can be fully-automated or semi-automated. In many approaches, the generation of sequence diagrams depends on other UML diagrams such as class diagrams and use cases. To our knowledge, there are no approaches to generate sequence diagrams directly from Arabic user requirements or without using other UML diagrams.

### 2.1. Semi-Automated Approach for Generating Sequence Diagrams

Recent studies presented semi-automated approaches for generating sequence diagrams using use cases or other UML diagrams [4, 7, 10, 17]. A semi-automatic approach to translate the use case descriptions into sequence diagrams is presented in [20]. The study also presented a set of rules for writing and rewriting the descriptions of use case diagrams that can be understood and is helpful for both developers and experts which can be transformed and translated to build the sequence diagrams. There are other approaches that have been proposed to improve auto-

generation of UML models using user requirements written in Arabic language [12].

## 2.2. Automated Approaches for Generating Sequence Diagrams

An approach that uses the Use Case Specifications (UCS) in generating a sequence diagram is presented by Mason and Suprisupachai [11]. The generation of sequence diagrams was based on UCS written in Spanish language. Yue *et al.* [22] proposed an approach to automatically generate the sequence diagrams based on corresponding use case specifications in which the objects are identified using a set of heuristics [21].

## 2.3. Generating UML Diagrams from User Requirements

Some approaches used for generating use case and activity diagrams from Arabic user requirements are presented in [2, 13]. The first approach used a semi-automated algorithm for generating activity diagram from Arabic user requirements using MADA+TOKAN Natural Language Processing (NLP) tool, in which the elements of the activity diagram have been extracted from Arabic user requirements. In the second study, researchers generated use case diagrams from user requirements written in Arabic language, in which a set of heuristic rules were proposed to obtain use cases. Other recent studies considered generating sequence diagrams from user stories written in English natural language [4, 8, 18, 19]. The first study used an algorithm that read a text file of user stories. For each user story, an XMI file is generated, which is later transformed into a sequence diagram using UML2 tool. The second study discussed generating behavioural diagrams (sequence and activity diagrams) by transforming the statements of the requirements into a structured representation (intermediary structured using frames). The frames were then translated into UML models. Researchers in this study used grammatical knowledge patterns along with lexical and syntactic analysis to analyse the requirements in order to get the frames for the corresponding requirements statements. By using the knowledge patterns in the resulted frames, the activity and sequence diagrams are generated. This study was presented using a set of performed case-studies. Sequence diagrams differ from other UML diagrams since they cannot be mapped easily to graphical diagrams. For example, lifelines are represented using vertical lines whereas the nodes are usually circles or boxes. Another reason is the connection points that are usually placed on one side of the node for incoming and outgoing connections, whereas in sequence diagrams messages are placed over the vertical line horizontally [2].

## 3. The Research Approach

The principal measure to find out the success of the software system is by measuring how much the output system meets the pre-set purpose and for what it is intended to do. To have good results, we should have good requirements that should have a set of characteristics based on IEEE standards for Software Requirements Specifications (SRS). Those characteristics force the user requirements to be correct, unambiguous, verifiable, traceable, complete and consistent. It is assumed that the requirements are good in the sense implied by the IEEE good requirements assumptions [14, 15, 16].

In this section, the proposed approach methodology is discussed. Figure 1 describes the general research methodology in which the first step is developing a set of heuristics to extract the sequence diagram key parts based on Arabic language grammar rules and sequence diagrams constructing rules. The textual Arabic user requirements are parsed using an NLP tool called MADA+TOKAN to get a set of tokens subject to a set of analysis phases including the application of the pre-developed heuristics to define the key parts of the sequence diagram in the processed textual user requirements scenario. The results are represented as a set of messages extracted from a set of senders and a set of receivers. The previous results are then placed on time order and transformed into Extensible Markup Language (XMI) to be drawn using UML drawing tools.

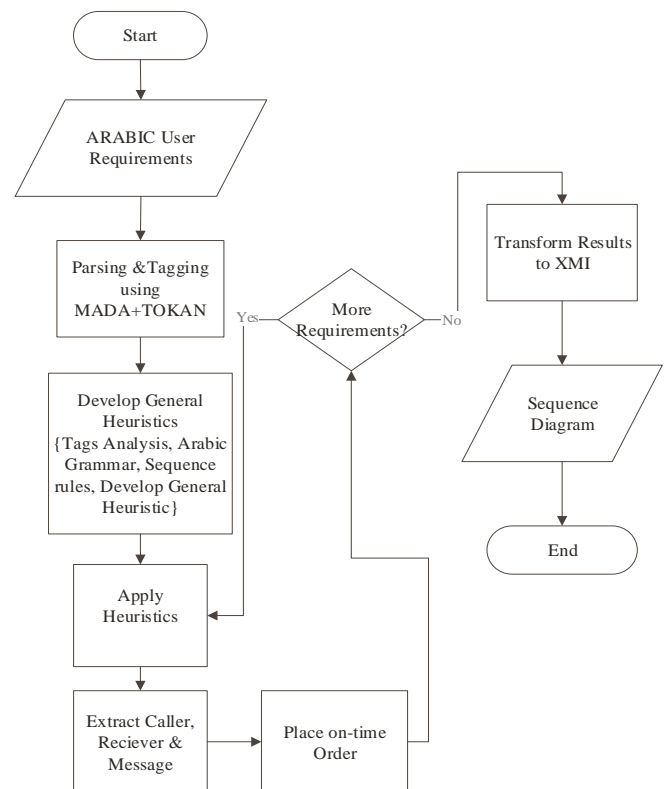


Figure 1. The Methodology of proposed approach.

MADA+TOKAN is a Toolkit for Arabic Tokenization, Discretization, Morphological Disambiguation, Part Of Speech (POS) Tagging, Stemming and Lemmatization [22]. MADA+TOKAN is a free tool, and is very customizable and versatile toolkit for NLP Arabic applications. It is used for the purpose of extracting morphological and contextual information from the raw Arabic text in order to be used for other applications. It mainly consists of two main components: MADA and TOKAN. MADA is the service of giving new Arabic text by adding morphological and lexical information, while the TOKAN is the utility of generating segmentation (Tokenization) based on the information produced from the MADA process in order to identify the stem of the words. Having the two utilities together (MADA+TOKAN) provide a powerful tool for pre-processing for the applications of NLP such as Automatic Speech Recognition (ASR) [3, 7]. Some user requirements cases of real system scenarios from Isra Computer and Programming Company written in Arabic were given and used in this research. A simple example is the following ATM system:

يقوم الزبون بإدخال البطاقة الى الصراف حيث يقوم الصراف بطلب الرقم السري من الزبون، يدخل الزبون الرقم السري الى الصراف ليقيم الصراف بإرسال المعلومات الى البنك للتحقق منها، يقوم البنك بإعادة النتيجة الى الصراف، ثم يقوم الصراف بعرض الخيارات للزبون، ويقوم الزبون بإدخال طلبه الى الصراف، يرسل البنك رسالة نجاح العملية الى الصراف، ويقوم الصراف بإرسال اشعار تنفيذ العملية للزبون.

MADA+TOKAN produces a tree to represent analysis and order of the processed words, as presented in Figure 2 where each word has a number representing tree level used heavily in analysis and building the heuristics phases in addition to a set of tags for each word as presented in Tables 1 and 2.

Table 1. MADA+TOKAN word grammar tags.

POS tag	Tag abbreviation	Word Type
Verb	VRB	فعل معلوم
Passive Verb	VRB-pass	فعل مجهول
Nominal	NOM	اسم
Particle	PRT	حرف/ أداة
Punctuation	PNX	علامة ترقيم
Proper Noun	PROP	اسم علم
Error	ERR	خطأ
Unknown		غير معروف

Table 2. MADA+TOKAN POS tags.

Dependency	Tag	Word Grammar
Subject	SBJ	فاعل/ نائب فاعل/ مبتدأ/ اسم كان/ اسم ان/ اسم كاد/ اسم فاعل/ اسم مفعول
Object	OBJ	مفعول لفاعل/ مفعول لاسم/ اسم مفعول/ مصدر/ اسم مجرور
Predicate	PRD	خبر لمبتدأ/ خبر ان/ خبر كان
Topic	TPC	مبتدأ
Idafa	IDF	مضاف اليه
Tamyiyz	TMZ	تمييز
Modifier	MOD	صفة/ حال/ ظرف
Flat	-----	اسم علم/ اسم اعجمي/ رقم/ علامات ترقيم مكررة

The first tag from Table 1 is the word type (verb, noun, punctuation, particle, etc.), while the second tag from Table 2 is the word parsing (verb, subject, object, etc.). These tags are used to determine participants and messages. Before start applying the proposed approach, the following conditions should be met:

- Each tag for each POS is expressed using the following set (Word, Word Type, Level, POS Tag).
- Each requirement is a verbal sentence, and each verbal sentence is an action while for each action there is a subject and sometimes an object.
- The tag <PNX> means the end of the sentence indicated by comma (,) or full stop (.), but in this phase sentences or user requirements statements will be separated not based on <PNX> tag - which represents the dot (.) or comma (,) in order to analyze verbal sentences. Rather, when we say requirement or sentence, we mean a set of tags between two consecutive <VRB> tags.
- In UML terms, subjects are called senders, objects are called receivers, both subjects and objects are called participants and actions are called messages.

This approach is applied in two phases, the first phase includes the scanning of the resulted tags from MADA+TOKAN to define the subjects set, which is a subset of participants set. At the beginning of applying the proposed approach, the following sets should be declared: Subjects= {}, Receivers = {} and Participants = {}. Subjects is a set of distinct words that has a POS tag of <SBJ> in the results of parsing a specific scenario. Subjects are used in subsequent phases to find the callers of that scenario, Receivers set is the set of the distinct receivers of the actions in the same scenario while the Participants set is the set of distinct callers and receivers. Subjects are defined at the first phase of scanning the parsing results while Receivers and Participants are defined and updated during the second phase of constructing sequence diagram key parts. Furthermore, a sequence table should be constructed and updated during the approach analysis process with the structure shown in Table 3.

Table 3. Sequence diagram table structure.

Statement #	Sender	Receiver	Message
-	-	-	-

### 3.1. Participants Identification

The participants of sequence diagram include:

Sender/ Caller, Main Actor and Receiver. By applying the first phase of scanning MADA+TOKAN parsing results on ATM system scenario, the produced subjects are as:  
Subjects = {البنك, الصراف, الزبون}

#### 3.1.1. Sender Identification

The sender or the caller for each statement represents

the subject of the action of that statement, which means senders are identified based on subject tags.

For each statement the sender of that statement is the subject of <SBJ > tag.

Example: يقوم الزبون بإدخال البطاقة الى الصراف

Using MADA+TOKAN tool, the results of statement parsing are shown in Table 4.

The main subject is <SBJ > tag is the word <الزبون>.

Table 4. MADA+TOKAN example POS tags.

Word	Word Type	Level	Grammar
يقوم	VRB	0	
الزبون	NOM	1	SBJ
+ ب	PRT	1	MOD
إدخال	NOM	3	OBJ
البطاقة	NOM	4	IDF
الى	PRT	4	MOD
الصراف	NOM	6	OBJ
حيث	NOM	7	MOD

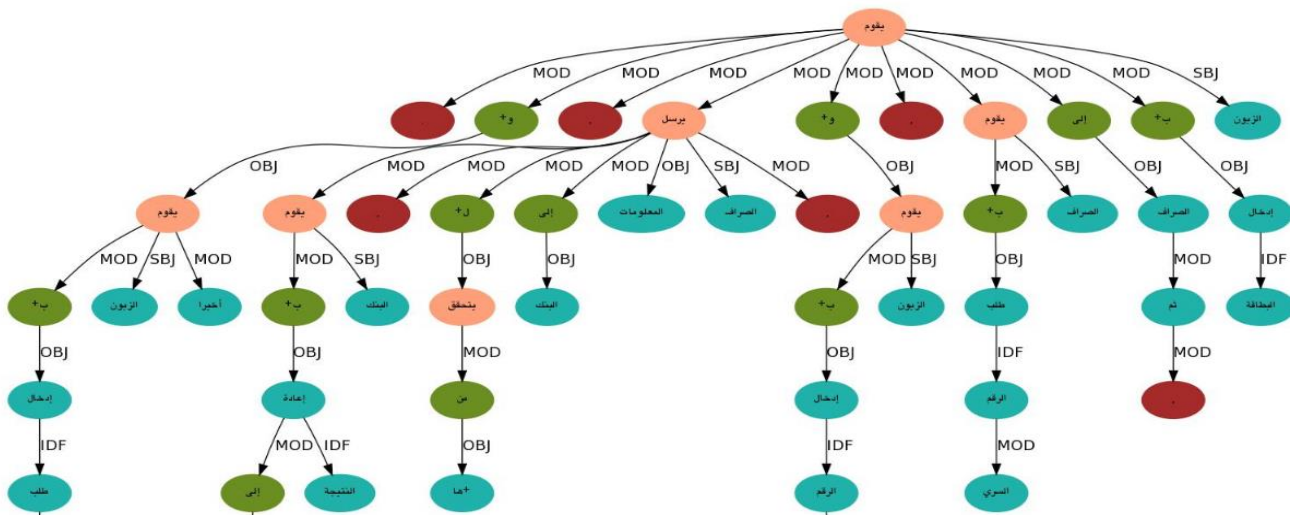


Figure 2. The Resulted tree from MADA+TOKAN ATM example.

**3.1.2. Main Actor Identification**

Main actor for any system is the first subject in that system. Based on resulted tree bank of MADA+TOKAN tagger, the first subject should have the level number of (1) after a verb of level (0) which is the root.

- *Generalization:* Rule P2: To find the main actor, search for the subject <Word, NOM, 1, SBJ> tag of the level (1) in all resulted MADA+TOKAN POS tags. Then add it to Participants set, if it does not exist in it. If it exists, then just mark it as main actor.

**3.1.3. Receiver Identification**

The receiver for each statement represents the object for the action of that statement. So, receivers are identified based on objects tags. The authors assumed, objects tags can be receiver or message.

To find the receiver for each requirement statement: find all < Word, NOM, level, OBJ > tags within each

- *Generalization:* Rule P1: For each user requirement statement with the following set of POS tags: <Word, NOM, level, VRB> <Word, NOM, level, SBJ> or <Word, NOM, level, SBJ> <Word, NOM, level, VRB>, find the <Word, NOM, level, SBJ> tag to determine the sender or the caller of that statement. Then the found subject should be added to the Participants group. If there is no <SBJ> tag, there is no sender and no message which means discarding the full statement.

statement and find the object belonging to subjects group, because each object in different point should be a subject (sender). If no objects exist within subjects group then last object of that statement is a receiver.

Update Participants set by adding the found receiver to it. Referring to Table 4, we can find that we have two objects in this statement: OBJ = {<إدخال> NOM 3 OBJ >, <الصراف> NOM 6 OBJ >}

- The first object <إدخال> does not belong to subject's group, and then it's not the receiver.
- The second object <الصراف> does not belong to subjects group then it is the receiver.
- Update Participants group and sequence table:

- *Generalization:* Rule P3: To find the receiver for each statement, apply the following rules on all <OBJ> tags within a statement:

*For each <OBJ> tag of <Word, Word- Type, level, OBJ>:  
check if the Word-Type is NOM then check if the object belongs to Subjects group  
If yes, then this object is a receiver, else it is a*

message (in the next section)  
 else, the last object in this statement is the receiver and all other objects are messages between the same sender and receiver within this statement.  
 check if Word-Type is VRB, then discards it.

### 3.2. Message Identification

To find the message within user requirements statements, we have to find the <OBJ> tag that is not the receiver. This tag is followed by an idafa <IDF> or modifier <MOD > tags to construct the message between two participants. Referring to Table IV, we can find that we have two objects in this statement:

- The first object <إدخال> does not belong to subjects group then it is a message:
- The next tag for this is <البطاقة NOM 4 IDF >so the full message is <إدخال NOM 3 OBJ > <البطاقة NOM 4 IDF >, “إدخال البطاقة”.
- The second object is <الصراف>, based on rule P2 it is a receiver and not a message.
- Update sequence table.
- *Generalization:* Rule M1: To find the message for each statement, apply the following rules on all <OBJ> tags that are not within a subject statement:

If the object tag is <Word, NOM, Level\_NO, OBJ> and it does not belong to the receivers group then it is a message. To find the message:

If the next tags is <Word, NOM, Level\_NO, IDF> then the message is <Word, NOM, Level\_NO, OBJ> + <Word, NOM, Level\_NO, IDF>

else if the next tag is <Word, NOM, Level\_NO, MOD> then the message is <NOM, Level\_NO, OBJ>+ <Word, NOM, Level\_NO, MOD>

else message is <Word, NOM, Level\_NO, OBJ>.

Update sequence table.

### 3.3. Algorithmic Approach of Heuristics

The algorithm for applying heuristics tags from parsing user requirements in MADA+TOKAN is presented as:

*Algorithm 1 ApplyHeuristics(Requirements)*

*Subjects = {}, Receivers = {}, Participants = {}, Sequence\_*  
*table [ ] [ ], Subjects= All <SBJ> tags*

*Find main actor based on Rule P2*

*Add main actor to participants & mark it as initiator*

*for all Arabic user requirements statements do*

*{Apply Rule P1 to find the sender*

*Update Participants and Sequence table*

*Apply Rule P3 to find the receiver*

*Update Receivers, Participants and Sequence table*

*Apply Rule M1 to find the message*

*Update Sequence table }*

*return sequence\_table*

The results of ATM system are: *Participants = { البنك، الزبون، الصراف، }*, *Subjects = {البنك، الصراف، الزبون، }*, *Receivers = {البنك، الصراف، الزبون، }*

The final sequence table is shown in Table 5. We can see that statement number 5 has been discarded based on Rule P1. The next required step is transforming the results for each row in final sequence table (message, sender and receiver) into XMI to be drawn using UML drawing tools.

Table 5. Final sequence table.

Statement #	Sender	Receiver	Message
1	الزبون	الصراف	إدخال البطاقة
2	الصراف	الزبون	طلب الرقم
3	الزبون	الصراف	الرقم السري
4	الصراف	البنك	المعلومات الى
5	البنك	الصراف	ارسال النتيجة
6	الصراف	الزبون	عرض الخيارات
7	الزبون	الصراف	ادخال طلب
8	الزبون	الصراف	ه الى
	البنك	الصراف	رسالة نجاح
9	الصراف	الزبون	ارسال اشعار

### 4. Evaluation and Validation

The proposed approach has been evaluated based on a set of experiments in which the sequence diagram for a set of cases is generated. By software engineers, and undergraduate students. The results have been compared and the accuracy measurements have been calculated. Used benchmark for testing consists of three examples: ATM, E-Registration and E-ticket systems. The used scenarios for real systems were developed in ISRA' SOFTWARE and COMPUTER COMPANY in Nablus-Palestine. The three systems' scenarios and their sequence diagrams are considered as the benchmark. Three software engineers from TAMKEEN SOFTWARE and COMPUTER COMPANY and 10 master students in Masters of Informatics programme in Palestine Polytechnic University have generated the sequence diagrams for the benchmark examples. Finally, the proposed approach has been applied on the same cases. Results of generating sequence diagrams were analysed to define the accuracy of the proposed approach in generating sequence diagrams in comparison with software engineer's solutions. The average of correct and incorrect number of messages, participants, and the order of messages between participants determines the system accuracy [10].

To facilitate the evaluation process, the final sequence table will be used, in which it consists of the sender, the receiver and the message for each action in the sequence diagram. So, for each case in the benchmark, to unify the comparison based, the generated sequence diagram by software engineers or by students will be mapped into sequence table format. Then it will be compared with the generated sequence table using the proposed approach.

The message is considered correct if it has a match of sender, receiver, and message action within the sequence table which maintains the order of the messages. Figure 3 describes the results of generating



the sequence diagram of ATM system by ten graduate students; in which each student construct the ATM sequence diagram from the given scenario. The ten students generated ten different sequence diagrams for the same ATM system scenarios where the results have been analysed as shown in Figure 3. The same process has been repeated for the experts as shown in Figure 4. Table 6 shows the results of generating sequence diagram for ATM system by graduate students, experts and proposed approach.

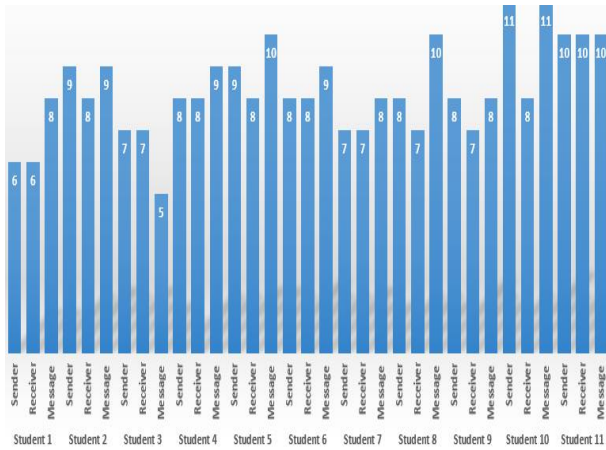


Figure 3. Students' evaluation results for ATM system.

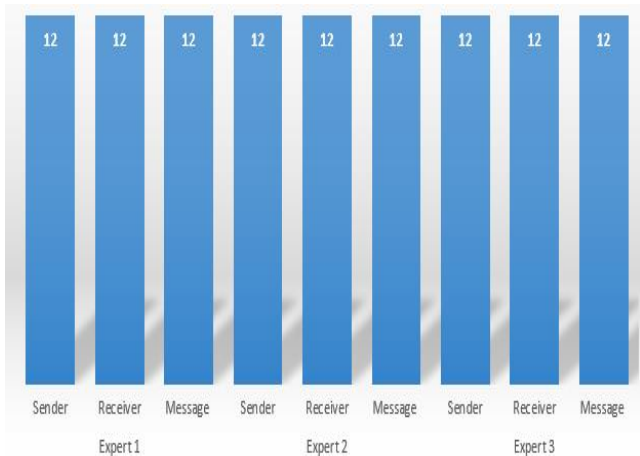


Figure 4. Experts' evaluation results for ATM system.

Same evaluation process has also been repeated for the other two case studies in the benchmark (E-registration and E-ticket systems). Tables 7 and 8 show results of evaluation accuracy for generating sequence diagrams for E-registration and E-ticket systems.

Table 6. Sequence diagram for ATM system evaluation accuracy.

	Benchmark	Experts	Students	Approach
# of messages	100%	94%	85%	91%
correct messages	100%	94%	79%	91%
correct senders	100%	94%	75%	91%
correct receivers	100%	94%	75%	91%
Participants	100%	100%	98%	100%

Table 7. Evaluation accuracy of E-registration sequence diagram.

	Benchmark	Experts	Students	Approach
# of messages	100%	92%	72%	75%
correct messages	100%	100%	73%	75%
correct senders	100%	100%	80%	83%
correct receivers	100%	100%	80%	83%
Participants	100%	100%	98%	100%

Table 8. Evaluation accuracy of generating E-tickets sequence diagram.

	Benchmark	Experts	Students	Approach
# of messages	100%	95%	71%	74%
correct messages	100%	100%	72%	75%
correct senders	100%	100%	79%	80%
correct receivers	100%	100%	81%	83%
participants	100%	100%	98%	100%

Table 9 shows the total accuracy of the proposed approach in generating sequence diagram based on three different scenarios for three different systems.

Table 9. Total accuracy of proposed approach.

	ATM System	E-Registration System	E-Tickets System	AVG
# of messages	91%	75%	74%	80%
correct messages	91%	75%	75%	80%
correct senders	91%	83%	80%	85%
correct receivers	91%	83%	83%	86%
participants	100%	100%	100%	100%

The proposed approach usually achieves better results than the students in generating sequence diagram and less than experts. It also has better accuracy in generating the participants and less accuracy in generating messages exchanged between participants. This makes sense because messages are more complicated than participants.

### 5. Discussion

A semi-automated approach to generate sequence diagrams from Arabic user requirements using an NLP tool by making use of user requirements to generate sequence diagrams directly. Our approach does not require any rewriting of user statements if the sender and the receiver within each statement are written explicitly. Other approaches require the user requirements to be in specific format. We proposed an approach that stores the textual representation of requirements in an intermediate form that can accept changes (optional) from the user too. However, the accuracy is limited by the correctness of the results provided by the Tagger and the Parser. Nevertheless, the results are satisfactory, especially in the case of sequence diagrams. Our approach improves software requirements analysis and leads to improved software development. Further research may update the proposed heuristics on complex scenarios and

automated generation of sequence diagrams using complex Arabic sentences.

## 6. Conclusions

In this paper, a new semi-automated approach for generating UML sequence diagrams from Arabic user requirements was proposed and evaluated. The proposed approach is essential in object-oriented applications, requirements analysis and software design especially in generating UML sequence diagrams from Arabic user requirements. The proposed approach has the main advantage of dealing with text written in Arabic language. Furthermore, a set of heuristics were developed and applied on a set of tokens resulted from an NLP tool called MADA+TOKAN to obtain sequence diagrams' key parts which include (participants and messages).

## References

- [1] Alami N., Arman N., and Khamayseh F., "A Semi-automated Approach for Generating Sequence Diagrams from Arabic User Requirements Using a Natural Language Processing Tool," in *Proceedings of International Conference on Information Technology*, Amman, 2017.
- [2] Arman N. and Jabbarin S., "Generating Use Case Models from Arabic User Requirements in a Semiautomated Approach Using a Natural Language Processing Tool," *Journal of Intelligent Systems*, vol. 24, no. 2, pp. 277-286, 2015.
- [3] Díaz I., Moreno L., Fuentes I., and Pastor O., in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, pp. 560-571, 2005.
- [4] Elallaoui M., Nafil K., and Touahni R., "Automatic Generation of UML Sequence Diagrams from User Stories in Scrum Process," in *Proceedings of International Conference on Intelligent Systems: Theories and Applications*, Rabat, pp. 1-6, 2015.
- [5] Gegentana X., "A Systematic Review of Automated Software Engineering," Master of Science Thesis in Program Software Engineering and Management, University of Gothenburg, 2011.
- [6] Harmon P. and Watson M., *Understanding UML-The Developers Guide*, Morgan Kaufmann Publishers, 2005.
- [7] Habash N., Rambow O., and Roth R., "MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization," in *Proceedings of the 5<sup>th</sup> International Conference on Language Resources and Evaluation*, Genoa, pp. 102-109, 2006.
- [8] Ilieva M. and Ormandjieva O., "Models Derived from Automatically Analyzed Textual User Requirements," in *Proceedings of the 4<sup>th</sup> International Conference on Software Engineering Research, Management and Applications*, Seattle, pp. 13-21, 2006.
- [9] Jabbarin S. and Arman N., "Constructing Use Case Models from Arabic User Requirements in A Semi-Automated Approach," in *Proceedings of the World Congress on Computer Applications and Information Systems*, Hammamet, pp. 1-4, 2014.
- [10] Li L., "Translating use Cases to Sequence Diagrams Automated Software Engineering," in *Proceedings of the 50<sup>th</sup> IEEE International Conference on*, Grenoble, pp. 293-296, 2000.
- [11] Mason P. and Suprisupachai S., "Paraphrasing use Case Descriptions and Sequence Diagrams: An Approach with Tool Support," in *Proceedings of the 6<sup>th</sup> International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Pattaya, pp. 722-725, 2009.
- [12] Nassar I. and Khamayseh F., "Constructing Activity Diagrams from Arabic User Requirements using Natural Language Processing Tool," in *Proceedings of the Conference on Information and Communication Systems*, Amman, pp. 50-54, 2015.
- [13] Nassar I. and Khamayseh F., "A Semi-Automated Generation of Activity Diagrams from Arabic User Requirements," *NGIT International Journal on Software Engineering*, vol. 2, pp. 1-5, 2015.
- [14] Nuseibeh B. and Easterbrook S., "Requirements Engineering: A Roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, Limerick, pp. 35-46, 2000.
- [15] Popescu D., Rugaber S., Medvidovic N., and Daniel M., "Reducing Ambiguities in Requirements Specifications via Automatically Created Object-Oriented Models Innovations for Requirement Analysis From Stakeholders Needs to Formal Designs," in *Proceedings of the Monterey Workshop*, Monterey, pp. 103-124, 2008.
- [16] Pressman R., *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Computer Society, 1998.
- [17] Samarasinghe N. and Stephane S., "Generating a Domain Model from a Use Case Model," in *Proceedings of the ISCA 14<sup>th</sup> International Conference on Intelligent and Adaptive Systems and Software Engineering*, Natural Sciences and Engineering Research Council of Canada, Toronto, pp. 23-29, 2005.

- [18] Sharma R., Gulia S., and Biswas K., "Automated Generation of Activity and Sequence Diagrams From Natural Language Requirements," in *Proceedings of the 9<sup>th</sup> International Conference on Evaluation of Novel Approaches to Software Engineering*, Lisbon, pp. 1-9, 2014.
- [19] Shatnawi A. and Shatnaw R., "Generating a Language-Independent Graphical User Interfaces from UML Models," *The International Arab Journal of Information Technology*, vol. 13, no. 6, pp. 291-295, 2016.
- [20] Souza F. and Giorno F., "Automatic Generation of Sequence Diagrams and Updating Domain Model from Use Cases," in *Proceedings of The 1<sup>st</sup> International Conference on Advances and Trends in Software Engineering*, Barcelona, pp. 85-92, 2015.
- [21] Thakur J. and Gupta A., "Automatic generation of Sequence Diagram from Use Case Specification," in *Proceedings of the 7<sup>th</sup> India Software Engineering Conference*, Chennai, pp. 20, 2014.
- [22] Yue T., Briand L., and Labiche Y., "Automatically Deriving UML Sequence Diagrams from Use Cases," Technical Report 2010-04, Carleton University, 2010.



University.

**Nermeen Alami** received her BS in Information Technology and MS in Informatics from Palestine Polytechnic University in 2016. She worked as a teaching assistant at department of Computer Science and IT, Palestine Polytechnic



University of Washington, DC USA in 1997, and his PhD from George Mason University, Virginia, USA in 2000. Dr. Arman is interested in Database and Knowledge-Base Systems, Algorithms, and Automated Software Engineering.

**Nabil Arman** is a Computer Science professor at Palestine Polytechnic University. He received his BS in Computer Science with high honours from Yarmouk University, Jordan in 1990 and an MS in Computer Science from The American



University of Washington, DC USA in 1997, and his PhD from George Mason University, Virginia, USA in 2000. Dr. Arman is interested in Database and Knowledge-Base Systems, Algorithms, and Automated Software Engineering.

**Faisal Khamayseh** is a Computer Science associate professor at Palestine Polytechnic University. He received his BS and MS in Computer Science from Southern Illinois University, USA 1995 and his PhD in Computers and Information Systems from the College of Computers and Information, Helwan University, Egypt, in 2009. He is a faculty member at the Department of Computer Science and IT. He is interested in Computer Algorithms and Software Engineering.

**Faisal Khamayseh** is a Computer Science associate professor at Palestine Polytechnic University. He received his BS and MS in Computer Science from Southern Illinois University, USA 1995 and his PhD in Computers and Information Systems from the College of Computers and Information, Helwan University, Egypt, in 2009. He is a faculty member at the Department of Computer Science and IT. He is interested in Computer Algorithms and Software Engineering.