Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

AR2B: FORMALIZATION OF ARABIC TEXTS WITH EVENT-B

Kheira-Zineb Bousmaha Ossoukine[1] and Lamia Belguith Hadrich[2]

(Received: 11-Nov.-2019, Revised: 31-Dec.-2019, Accepted: 3-Feb.-2020)

ABSTRACT

Transforming natural software requirements into a more formal specification is difficult and may be an excellent application for natural language processing. This problem is not recent. It aroused and still arouses great interest, because it gives rise to many challenges in various scientific fields, such as automatic language processing, requirements engineering, knowledge representation and formal verification. This paper proposes a platform and a strategy to transform software requirements specified to formal specification with event-B. The texts used are those of Arabic language, which is really a challenge. The Ar2B system is built and the experiments showed good results with an accuracy of 70%.

KEYWORDS

Arabic natural language processing, Information system, Information extraction, Formals' specification, Event-B.

1. INTRODUCTION

One of the challenges of Natural Language Processing (NLP) is the understanding of texts and their interpretation. To theorize the meaning of a text and automatically reach a level of understanding is notoriously difficult. This ambitious objective has been regularly adjourned to more local and less complex tasks. One of its axes is the formalization of the text describing the specifications of the requirements of the information system (IS).

Conceptual modeling of data is a very important phase in any development of the IS. Current design methods are based on models and data processing using various formalisms (graphs, entity / relation, UML diagrams, algorithmic notation, object representation…etc.). They are a factor in reducing costs and delays. The choice of a representation model that is sufficiently formal, precise and expressive to represent the semantics of natural language specifications allows an automated transition to formal specifications. The semi-formal and formal models often coexist in the same project, because they are complementary and each of them compensates for the disadvantages of the other, as they allow for better distribution and automation of tasks.

Modeling platforms will now be able to be used to make code generation or formal verification as well as moving back and forth between the code and the model without loss of information [1]. Automating the design and formalization has become a considerable activity which gives rise to many challenges in different scientific fields, such as requirements engineering, automatic language processing, information retrieval, representation and engineering of knowledge.

Several works have been interested in this topic which continues to attract much interest in more recent research, aiming to process more specifications in a shorter time and less subjective than an expert who relies solely on his knowledge and skills [2]. Our objective is to propose an approach to design a platform and develop a strategy to formalize the functional specification text to event-B. The originality of our research lies in the choice of the language proposed for study, the Arabic language, to which no work on this theme has been devoted. We offer assistance that can help in the processes of formalization and conceptual modeling based on reliable methods and tools. We propose a platform Ar2B (Figure 1) dedicated to Arabic Natural Language Processing (ANLP). We proceed to a linguistic treatment, conceptualization and formalization of text, oriented towards the conceptual modeling of information

1. K. Z Bousmaha Ossoukine is with Department of Computer Science, RIIR laboratory, University Oran1 Algeria. Member of ANLP-RG (Arabic Natural Language Processing – Reseach Group), Emails: kzbousmaha@univ-oran1.dz; kzbousmaha@yahoo.fr
2. L. Belguith Hadrich is with Departmant of Computer Science, Faculty of Economics and Management (FSEGS), University of Sfax,Tunisia, Head of ANLP-RG, Emails: l.belguith@fsegs.rnu.tn; lamia.belguith@gmail.com

systems (ISs). We provide a set of tools and techniques for the transition from the informal to the formal, knowing that no work in that direction has been ever done with Arabic text.

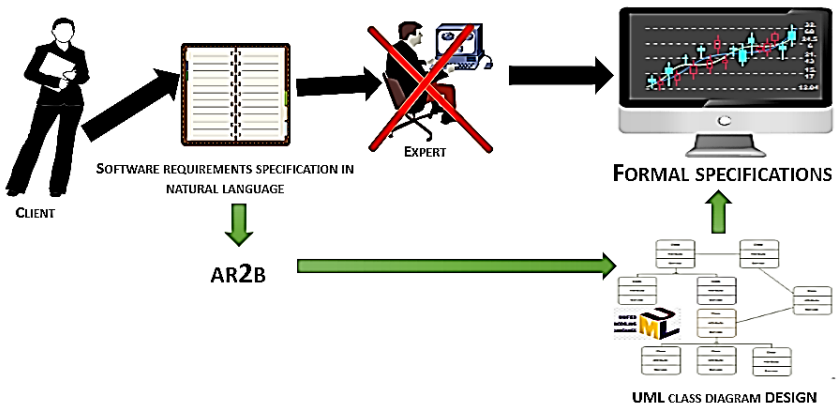


Figure 1. Automation of formalizing requirements.

An example of part of a specification text treated by Ar2B:

الدرس معرف باسم ورقم ومدة زمنية. يمكن لهذا الدرس أن يدرس خلال سنة في عدة دورات. الدورة مميزة برقم وتاريخ البداية وثمن وعدد أيام الدورة. في غالب الأحيان الدورة مؤمنة من طرف عدّة منشطين. الدورة موضوعة تحت مسؤولية المنشط.

للمنشط رقم تسلسلي فريد ولقب ومرتب وعنوان شخصي. فأما عن المنشط، فيستطيع التدخل في عدة دورات في السنة على الأكثر، فلذلك نرغب في تسجيل عدد الساعات التي يدرسها كل منشط. وأما عن الدورة، فإنها متبوعة بعدد من المشاركين الذين يتميز كل منهم باسم وعنوان ورقم هاتف، كما يمكن للمشارك أن يكون موظفاً أو شخصاً. يعتبر المنشط إما منشطاً رئيسياً أو ثانوياً. يجب أن يكون سن المنشط أكبر من أو يساوي 18 سنة.

The rest of this paper is organized as follows: Section 2 deals with related work. We present our platform approach in Section 3. We show the experiments that we conducted and the first results of the first version obtained in Section 4. Finally, a conclusion and some perspectives for this work are given in Section 5.

2. LITERATURE REVIEW

Several research works have been aimed at automating the development of semi-formal models based on requirements' specifications in French or English language. The proposed approaches most often use NLP techniques. We can cite the works of [3]-[6] and the list is not exhaustive.

[7] proposed NL2Alloy combining a succession of tools allowing passing from constraints written in natural language to SBVR (NL2SBVR) rules, then towards UML/OCL (SBVR2OCL). They use automatic language processing methods and semantic technologies to generate UML models from natural language requirements. Manual interactions with the designer are then inevitable leading to semi-automatic approaches. On the other hand, several works have focused on the transition from UML to formal languages, such as B [8]; the Z language [9], which uses conceptual graphs as a pivotal model; or the language Maude [10], [1].

[2] used an ontology as a pivotal model; the formal language VDM and VDM ++ [11], …etc. The application of Artificial Intelligence techniques to requirements engineering [12] suggests software to be developed faster and better [13].

For all these research works, the results are satisfactory and their f-measure exceeds 90%. However, few studies are related to the state-of-the-art. These studies proposed only semi-formalizing Arabic user requirements and generating UML diagrams from them. They used algorithms for generating use case [14], sequence diagrams [15] and activity diagrams [16].

[14] and [15] generated diagrams from user requirements written in Arabic language, in which a set of heuristic rules were proposed.

[16] used a semi-automated algorithm for generating activity diagrams using MADA+TOKAN NLP tool, in which the elements of the activity diagrams have been extracted.

3. THE AR2B PLATFORM

Modeling of natural language became an issue of particular importance. It encouraged researchers to develop a variety of linguistic models that could solve practical problems. Linguistic models deal with statements as they are used to express meanings. They involve a body of meanings and a vocabulary to express meanings, as well as a mechanism to construct statements that can define new meanings based on the initial ones. The conceptual model represents 'concepts' (entities) and relationships between them. It plays an important role in the overall system development life-cycle. It is clear that if the conceptual model is not fully developed, the execution of fundamental system properties may not be implemented properly, giving way to future problems or system shortfalls, such as lack of user input, incomplete or unclear requirements and changing requirements. The concepts of the conceptual model can be mapped into physical design or implementation constructs using either manual or automated code generation approaches. To remove ambiguity and improve precision, to verify that the requirements have been met, to reason about the requirements/designs, to test for consistency, to explore consequences, to check automatically the properties; …etc., we need formalization. The formal model is based on rigorous methods and formats. Moving from the conceptual model to the formal model seems interesting.

Event-B is a formal model; it's an extension of the B-method (J-R. Abrial). It is devoted to system engineering (both hardware and software) and to specifying and reasoning about complex systems: concurrent and reactive systems. Event-B models are organized in terms of the two basic constructs: contexts and machines. Contexts specify the static part of a model, whereas machines specify the dynamic part. The role of the contexts is to isolate the parameters of a formal model and their properties, which are assumed to hold for all instances. A machine encapsulates a transition system with the state being specified by a set of variables and transitions modelled by a set of guarded events. Event-B allows models to be developed gradually *via* mechanisms, such as context extension and machine refinement. These techniques enable users to develop target systems from their abstract specifications and subsequently introduce more implementation details. More importantly, properties that are proved at the abstract level are maintained through refinement and hence are guaranteed to be satisfied also by later refinement. As a result, correctness proofs of systems are broken down and distributed amongst different levels of abstraction, which are easier to manage [17]. Event-B comes with a new modelling framework called Rodin (like Atelier B tool for the classical B). The Rodin platform is an eclipse-based open and extensible tool for B model specification and verification. It integrates various plug-ins: B Model editors, proof-obligation, generators, provers, model-checkers, UML transformers, …etc.

Our platform allows conducting linguistic pretreatment, modeling and formal validation activities for text in Arabic language. We note, through the state-of-the-art, that a direct transition from informal specifications to formal specifications is not possible [18]. The common solution would be the transition to a pivotal intermediate representation that would reduce the gap between the two types of specifications [2].

To conceive Ar2B, two questions arose:

1. How is the problem of linguistic pretreatment to be solved with all the difficulties of treatment that the Arabic language knows?
2. How are the specifications written in natural language to be formalized?

The solution that we adopted in response to these questions is to conceive three models: linguistic, conceptual and semi-formal models in order to finally lead to formalization. It can be summarized as follows:

1. It is necessary to treat the text by various linguistic analyses by integrating them into a platform in order to annotate them and to represent them by an intermediate model that can serve as a pivot for conceptualization: **Linguistic model** represented by XML.
2. It is imperative to use an intermediate representation to move from this linguistic model to a semi-formal specification: **Conceptual model** represented by semantic networks.

3. It is necessary to transform this conceptual model by means of conceptualization rules and to represent it by a rigorous representation model to make it relevant at the level of formal specification. It is represented by UML class diagram (semi-formal) then by event-B: **Formal model.**

As shown in Figure 2, the text constitutes the basis of modeling. From it, it is possible to extract a linguistic model that will contain the elements expressed in the text. The conceptual model corresponds to a modeling process whose automation is realized only by a linguistic model. The choices of modeling are important at this level. They strongly depend on the granularity of the desired description and the objectives of the modeling. The semi-formal model is based on the exploitation of the representation and formalization potential of the UML language. The lack of formal semantics from which UML suffers can lead to serious modeling problems [1], generating inconsistencies in the models developed. In addition, its simplicity has as a price, which is lack of precision. This led us to make a transition to the formal model in event-B. A formalization of the conceptual modeling is thus generated.

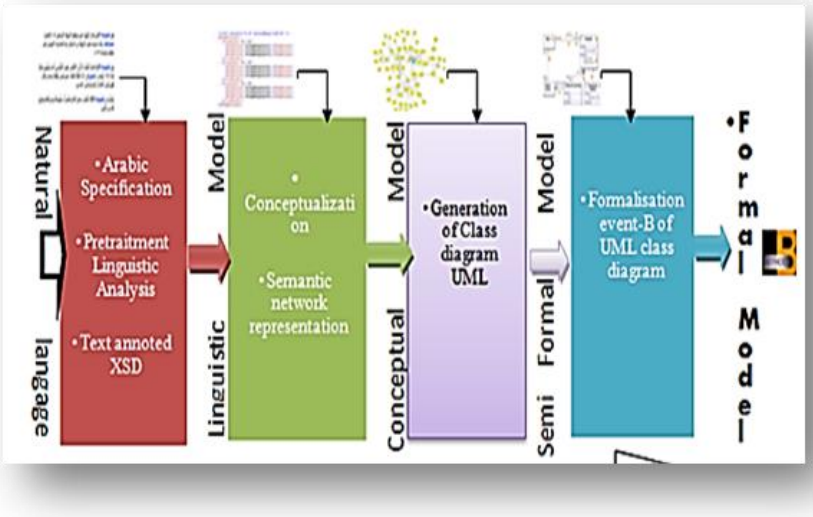


Figure 2. Models of the platform Ar2B.

3.1 The Linguistic Model

To establish the linguistic model and proceed with the pretreatment (preprocessing) of the text of specification, we have designed the tool Alkhalil+ [19]. It does a considerable amount of work on MSA (Modern Standard Arabic), segmentation of the text in sentences, tokenization, morphological analysis, lemmatization, part-of-speech tagging (POS tag), disambiguation and diacritization. After the segmentation of the text into sentences, we use Alkhalil Morpho Sys (Alkhalil Morpho Sys, Version 1.3, http://sourceforge.net/projects/alkhalil/), a morphological online analyzer for Standard Arabic text. It is based in one part on the modeling of a large set of Arabic morphological rules and on the other part on the integration of linguistic resources that are useful to the other analysis. Next, we implemented a set of grammar rules as probabilistic contextual free grammar (PCFG) with a set of 87 ATNs (Augmented Transition Networks) applied to the pre-labelled text. Once the associated grammatical label for each word, a second disambiguation, is made, we apply a method based on decision theory to filter all successful applicant patterns and determine the final POS tag and diacritics of this word. The experiments have carried a disambiguation rate that is above 92.33%. The output is an XML file.

The XML schema consists of a set of sentences. Each word is described by its position in the sentence, its name, its lemma, its tag, a class of the verb, its multiplicity and its type.

The utilized tag set comprises the collapsed tags available in the Arabic TreeBank distribution {CC, CD, CONJ+NEG PART,DT, FW, IN, JJ, NN, NNP, NNPS, NNS, NO FUNC, NUMERIC,COMMA, PRP, PRP$, PUNC, RB, UH, VBD,VBN, VBP, WP, WRB} and we use 18 morpho-syntactic tags of Alkhalil {فا, مف,مفا, آ,زمك, نك,إض,فض, وش,أ ص,صم,صر, صه, صن,عانس, جانعا} to complete the description of verbs and nouns.

We have established a typology of verbs to determine the meaning of the sentence. We noted that the meaning of the verb gives us an indication of the semantics of the relationship in the future conceptual model. This typology is specific to the design of the IS. It is called class of verb.

Class of verb of state: includes the verbs that describe a state; these verbs express a hyponymy. Example: تنقسم, تتجزأ, تتفرع

Class of possession verb: introduces the concept of structural description. Example: يملك،يحوز،يتميز

Class of constraint verb: includes the verbs that are likely to define the constraints on attributes. Example: تفوق, تتجاوز, تنحصر, تتعدى

Class of action verb: includes the verbs that define an action; this action can be interpreted by a relationship. Example: يرسل، يكتب، يشارك، يزور

Class of composition verb: includes the verbs expressing the meronymy. Example: تحتوي، تتركب، تتشكل

Class of verb NULL (or sentiment): includes the verbs that signify no important action in the sentence. Example:.....يستطيع، يمكن، يقدر, يجب

About multiplicity, it is specific to the noun; it depends on its morphological nature; multiplicity = '2' (dual), multiplicity = '*' (plural). If the number is not specified in the text, it is equal to 1 by default.

Considering the type, it is an attribute to particles. It takes the value: NULL or NOT_NULL depending on whether or not these particles have a semantic meaning and a role in the design of our future IS. For example, particles ل' /lem' and في' fi' can in some sentences play an important role in the design; for example, in the sentence: للمعلم اسم و لقب و رقم شخصي / the teacher is characterized by a name, a surname and a personal number /, the particle 'ل' plays the role of a verb of possession. On the other hand, in the sentence:يدرس المعلم للطلاب مادتين إثنتين , / The teacher teaches two subjects to the students / particle 'ل' plays no role; its type will be set to NULL.

3.2 The Conceptual Model

The proposed approach starts by extracting terms and compound terms from the annotated XML file. The second step is the design of the chunker; we have defined a list of categories of chunks that were necessary for the classification of the sentences in order to extract the meaning; only the chunks pertinent to the structural description of the future IS are selected and we attribute to them different roles. Then, we proceed with the classification of these sentences according to sentence patterns that we have already determined. A semantic network represents the extracted information. A set of design patterns are applied, hence generating the corresponding UML class diagram.

3.2.1 Extraction of Simple and Compound Terms

To extract simple terms, our approach is based on weight calculation. We have not assigned a weight to every word in the text; it is only calculated for these whose tag is (NN, DT NN, NNS, NNP, NNPS), because there could be classes or potential attributes in the design of the future IS. This weight can be critical for the recognition of the nature of this concept. We calculated the frequency of the term using the lemma; the weight is calculated with the formula tf-idf. A list of term candidates is so extracted.

To extract word pairs, we have used a hybrid method. We have defined linguistic patterns to determine couples of candidates and then we have filtered them by using a statistical method based on mutual information (MI) in order to keep the couples of pertinent words. A third filtering is performed during the validation of the semantic network. If the chosen pair consists of two terms that have been identified as an entity, the couple is then rejected as a compound word and another treatment will be assigned to it.

The linguistic pattern. For syntactic patterns, we have adopted the research work of [20]. We focus here on collocations consisting of two lexical units and respecting the following schemes: NN + JJ; (DT+NN) + (DT+JJ); NN + (DT+NN); NN + (DT+JJ); NN + NNP; NNP + NN……. (NN: indefinite noun; JJ: adjective; DT: definite noun).

Example: المدرسة الابتدائية ,جامعة العلوم ,الطالب الجامعي ,طالب متفوق

The filtering calculation of mutual information (MI). For each pair of words learned previously, we calculate the MI. The MI is used to determine whether two words are closely related or not. Given two words designated by the variables x and y, the MI is calculated using the following formula (1):

$$MI(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)} \quad (1)$$

where, p (x) and p (y) are the probabilities of observation of respectively words x and y; and p (x, y) is the probability to observe them together. Once the IM is calculated for each couple, we experimentally set a threshold for preferred pairs with strong cohesion. Figure 3 shows the compound terms extracted by Ar2B.

N°	Terme1	Lemme1	Terme2	Lemme2	Nbr. occ.	Inf. mut.
1	الأمراض	مَرَض	الخطيرة	خَطِير	1	170.5
2	نمو	نُمُوّ	الخلايا	خَلِيَّة	1	5.6833
3	غير	غَيْر	الطبيعية	طَبِيعِيّ	5	56.8333
4	أنواع	نَوْع	الخلايا	خَلِيَّة	1	3.2476
5	بسرعة	سُرْعَة	أكبر	أَكْبَر	1	341.0
6	الخلايا	خَلِيَّة	الطبيعية	طَبِيعِيّ	1	3.7888
7	أورام	وَرَم	حميدة	حَمِيدَة	2	48.7142
8	الخلايا	خَلِيَّة	السرطانية	سَرَطَانِيَّة	2	22.7333
9	أورام	وَرَم	جديدة	جَدِيد	1	24.3571
10	أجزاء	جُزْء	أخرى	أُخْرَى	2	113.6666
11	أنواع	نَوْع	السرطان	سَرَطَان	3	6.354
12	نمّاى	نِمَاى	السيطرة	سَيْطَرَة	1	341.0
13	سرطان	سَرَطَان	الجلد	جِلْد	2	5.9304
14	سرطان	سَرَطَان	الثدي	ثَدْي	5	12.355
15	جزء	جُزْء	آخر	آخَر	1	113.6666
16	خلايا	خَلِيَّة	الجلد	جِلْد	1	4.5466
17	الخلايا	خَلِيَّة	الجديدة	جَدِيد	1	11.3666
18	الثدي	ثَدْي	المنتشر	مُنْتَشِر	1	28.4166
19	أنسجة	نَسِيج	الجسم	جِسْم	1	14.2083
20	الجسم	جِسْم	والثاني	ثَانِي	1	56.8333
21	أنواع	نَوْع	مختلفة	مُخْتَلِف	1	24.3571

Figure 3. Extraction of compound terms by Ar2B.

3.2.2 Base Phrase Chunking

Free-order language complicates the grammar construction. The basic order of Arabic words in a sentence is Verb–Subject-Object (VSO). However, other orders are possible: SVO and VOS. The idea of chunking was the solution to i) reduce the number of rules and thus improve the performance of the relationship extraction system and ii) eliminate the treatment of temporal and behavioral syntagm types in the structural description of the class diagram. Its use in the semantic analysis has made our system much more efficient in the classification of the sentences according to the established patterns of sentences. For this task, we use a setup similar to that of [21], with the BIO annotation representation: "Beginning", "Inside" and "Outside" the chunk. Ten types of chunked phrases are recognized: {VP, NP, ADJP, PP, ADVP, CONJP, INTJP, PREDP, PRTP and SBARP}. We have added two other types: CD that indicates the cardinality necessary to determine in the design of the future IS and CARD to define static constraint. Chunking is introduced by the presence of words as indicated in Table 1.

Table 1. The chunk added by Ar2B.

Type of Chunk	**Begin of the chunk**	
CD	جميع، كل، وحيد، فقط.....	number
CARD	أكبر من، يساوي، ما بين، أقل ، أصغر من، أصغر من أو يساوي، أقل من، على الأقل، في أحد، واحد من.....	constraint verb type

An example of chunking is given in Figure 4.

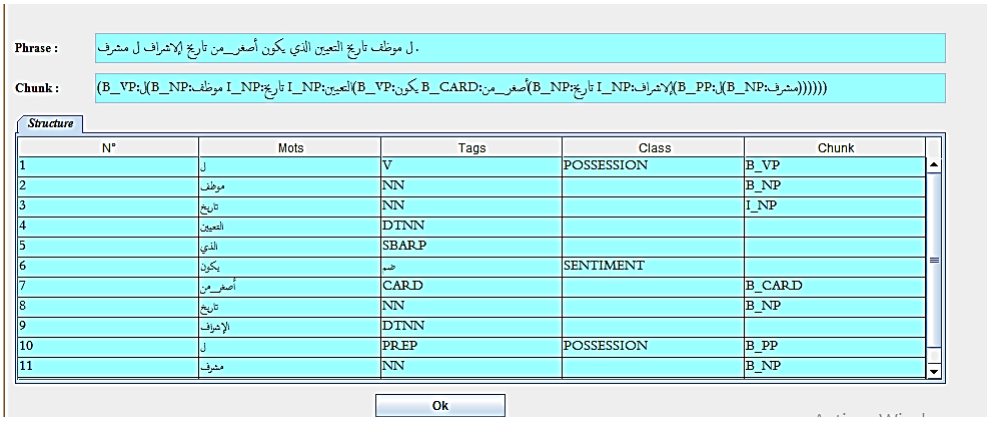


Figure 4. Chunking by Ar2B.

3.2.3 Semantic Analysis

For semantic analysis, our approach is hybrid. It is based on Fillmore's case theory, combined with the verb-based and pattern-based approach. As shown in Figure 5, we begin with the assignment of roles to the various components in the chunks of the sentence. Then, we proceed with classifying these sentences into patterns. A semantic network is generated as output in order to represent the whole extracted information.

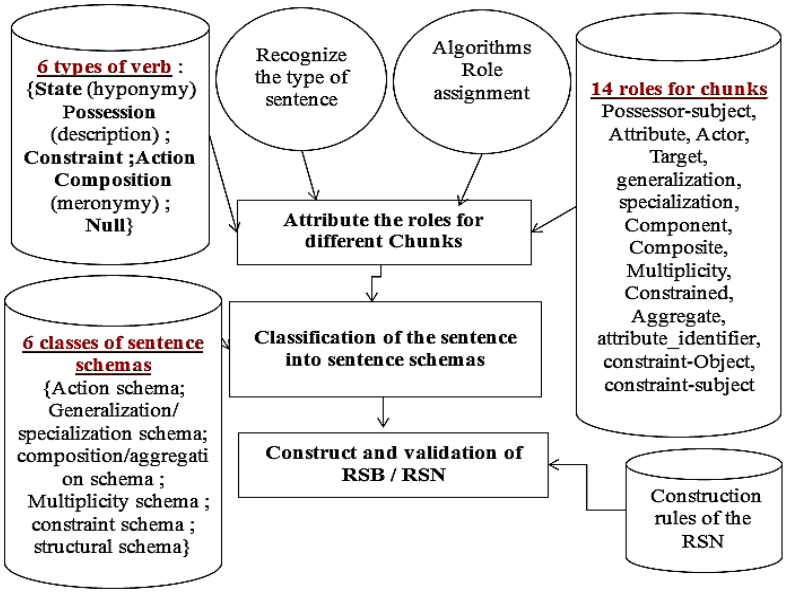


Figure 5. Semantic analyzer design approach.

Step 1/ Recognition the type of sentence and attribution of roles: Fillmore accords that the verb is the central component of the sentence. He schematizes the sentence as distinction between modality (M) and proposal (P). The modality contains information about negation, time, mode and appearance. V is the verb. Each Ci is the name of a case (role) that will represent a name related to the verb by semantical case Ci.

$$S_V = \mathrm{M} + VP + C1 + C2 + \cdots . + Cn \quad (2)$$

We have extended this definition to take into account the nominal sentence, knowing that it is non-existent in other languages, such as English and French. The formulae will then be written as:

$$S_N = \mathrm{M} + \mathrm{Pivot} + C1 + C2 + \cdots . + Cn \quad (3)$$

$$Sv = M + P\,,\;\; PV + C1 + C2 + \cdots . + Cn \quad (4)$$

For our design, Ci indicates the semantic case that binds a *chunk* with *verbal chunk (VP)* or with pivot.

We have identified 14 roles (*Ci)* (Figure 2). We were inspired to define our roles to those defined in Fillmore's causal theory. We adapted them to the Arabic language and to the purpose of our conception.

For S_V processing, it is recognized by the presence of a verbal chunk (VP) and not a verb. We were inspired by the research of [22] for detecting it. Our VP is recognized by the identification of a verb or verbal noun/ ((مصدر (صم, صر, صه, صأ)), active participle / ((فا) اسم فاعل) or passive participle /((اسم مفعول مف)) or the particle 'ل' (the particle of possession/لام الامتلاك)**.** We look for the verb class and the identification of the semantic relations that link the different chunks to the verbal chunk, in order to assign coherent roles to the different words and chunks composing this sentence.

We have defined for each class of verb an algorithm for assigning roles, except for the class of verb null considered as a stop word.

Algorithm *Class of action verb*

```
Begin
For i=1 to n do //n: number of sentences in text//
 Find_chunk_verbal (”VP”,i,exist,class) ;//research chunk VP (verbal chunk) //
  If exist then
   If class=”action” then
    If Find_chunk (i,”NP”) then  //research chunk NP (nominal chunk)//
    Role_chunk_NP ← “actor”; //For each term in this chunk NP attribute role “Actor”//
     While Find_chunk (i,”NP”) do //research another chunk NP//
       Role_chunk_NP ←”target” //For each term in this chunk NP, attribute role “target”//
       End;  End;
     While Find_chunk (i,”PP”) do     // PP: prepositional chunk//
       Role_chunk_PP ←”target”; end;
     While Find_chunk (i,”CD”) do  // CD: Cardinal Chunk//
            Role_chunk_PP ←”multiplicity”; end;
    end;end;end;
```

In Figure 6, an example is given of a sentence in which *Class of verb is State*. The algorithm assigns chunk roles.

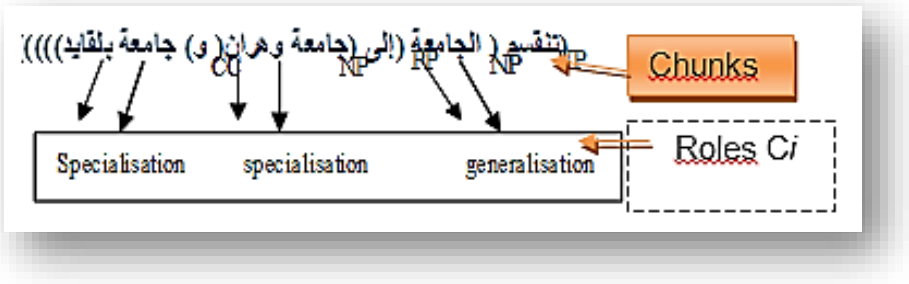


Figure 6. Example 'State' class verb=' ينقسم '.

Remark: جامعة وهران و جامعة بلقايد have been determined as compound words by our tool.

For S_N processing, usually the nominal sentence describes either an association relationship or an inheritance relationship. The relationship is deduced by a prepositional chunk (PP), nominal chunk (NP) or adjectival chunk (ADJP). It may even be implicit, in which case we look for a pivot element (the subject of the action) in the nominal chunk, identified by its Pos tag and by its position in this chunk. We have established algorithms that assign roles for each chunk of the sentence.

Step 2/ Classification of the sentence into sentence schemas: The sentence patterns allow us to stereotype the sentences; we have classified them according to six schemas (Figure 5). This classification determines the first interpretation of the specification text.

For example, all sentences resulting from one of this combination are classified as Structural, Action or

Generalization/ specialization schema.

Structural schema
Verbal Chunk in possession class + Role possessor subject+ Role Attribute+ constraint Role.

Action Schema
Verbal Chunk in Action class + actor Role+ target Role +multiplicity Role

Generalization/ specialization schema
Verbal Chunk in State class + Generalization Role+ specialization Role +multiplicity Role

The sentence shown in Figure 7 is in the structural schema:

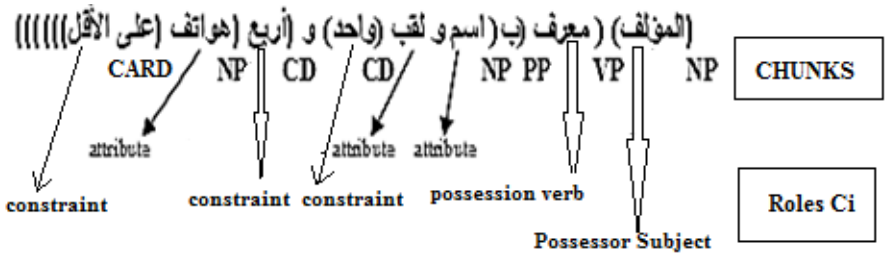


Figure 7. Example of classification in structural schema.

The constraint is recognized by the presence of a chunk CARD in the sentence. Once this chunk is detected, the processing of the constraint passes priority over the treatment of the different types of sentences.

Step 3/ Construction and validation of semantic network (RSB/RSN): After classifying the sentences, we applied specific algorithms to extract the relevant information to represent them by a rigorous model to represent knowledge, which is the **S**emantic **N**etwork (RSN) in order to make it accessible.

The **r**aw **s**emantic **n**etwork (RSB) is the first network built; the RSN is the validated version of the RSB with pattern design. The RSN is characterized by a set of nodes and arcs. We have defined a total of 7 types of nodes and 10 types of arcs as shown in Figure 8, which allowed us to represent all relevant information in our corpus.

The nodes of the RSB: {entity, action, multiplicity, constraint, value, negation};
The nodes of RSN = RSB U {operation};
The arcs of the RSB = {poss, acti, mult, is-a, comp / arg, cti, not, val};
The arcs of RSN = RSB U {op, id}.

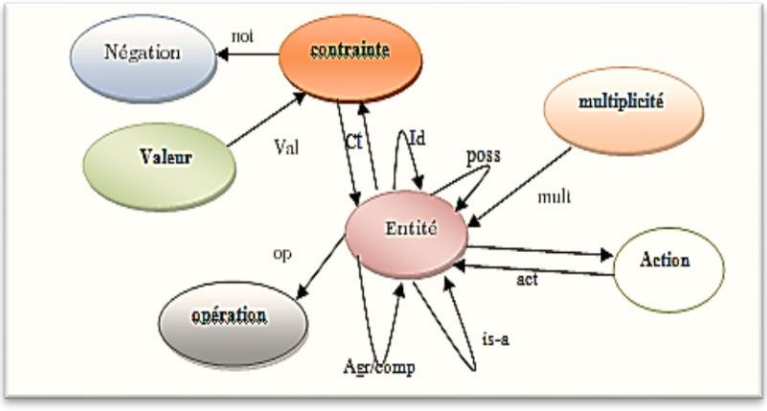


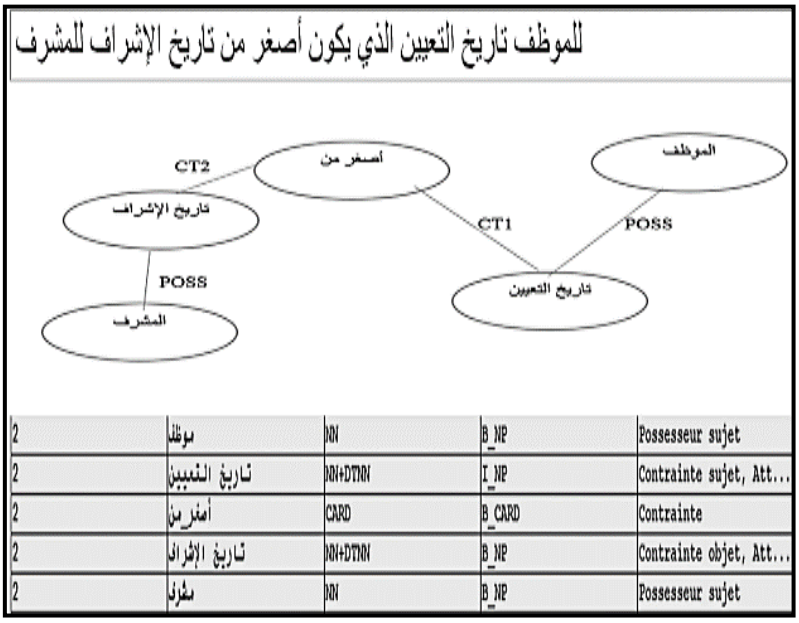
Figure 8. The meta-model of RSB / RSN.

For each sentence schema, we have applied specific algorithms for processing nodes and arcs.

An example of rule of structural schema:

- All terms having as Role “possessor_subject” and role “attribute” will all be transformed into entity nodes.
- They will be linked by an arc 'poss'. The source of the arc will be the term whose role is "possessor_subject".

In Figure 9, Ar2B treats a sentence. The second column contains and affichs the simple terms found: مشرف, موظف and compound terms: تاريخ التعيين, تاريخ الإشراف. The third column contains the Pos tag. In the fourth one, it presents the results of chunking. The roles assigned are displayed in the fifth column and the corresponding RSN is generate. We note that "poss" links were deducted automatically between تاريخ التعيين and موظف ; تاريخ الإشراف and مشرف. Although they were not specified as such in the sentence. Our approach allows for the detection of implicit arcs and nodes using Chunks role assignment algorithms and sentence classification algorithms.



2	موظف	NN	B_NP	Possesseur sujet
2	تاريخ التعيين	NN+DTNN	I_NP	Contrainte sujet, Att...
2	أصغر من	CARD	B_CARD	Contrainte
2	تاريخ الإشراف	NN+DTNN	B_NP	Contrainte objet, Att...
2	مشرف	NN	B_NP	Possesseur sujet

Figure 9. RSN of constraint schema sentence treated by Ar2B.

Once the network is generated, we proceed with its normalization for verification of the consistency of the network. To do that, we have established normalization rules inspired by design patterns of software engineering. These rules will help us remove some nodes and arcs and create others to validate coherence and compliance. Rule 5 is an example of normalization rules. Figure 10 shows the application of this rule to the specification text given at the top (المنشط الرئيسي والمنشط الثانوي).

Classe source	Classe cible	Type de lien	Liens	Association	Cardinalité
الدرس	الدورة	Association	يدرس		0..n/1..n
الدورة	المنشط	Association	مؤمنة	المنشط	0..n/1..1
الدورة	المنشط الرئيسي	Association	مسؤولية	الدرس	1..n/1..1
الدورة	المشارك	Association	متبوعة	المنشط	0..1/0..n
المنشط	الدورة	Association	تدخل	الدورة	0..1/1..1
المنشط الرئيسي	المنشط	Heritage		الدورة	
المنشط الثانوي	المنشط	Heritage			
الشخص	المشارك	Heritage			
الموظف	المشارك	Heritage			

Figure 10. Application of rule 5 by Ar2B.

Rule 5: Transformation in the entity node

Any entity node containing a compound word will be converted into 2 entity nodes connected by a link 'is_a' if one of the terms (or both) is a node entity.

Figure 11 represents the RSB of the specification given in Section 1.

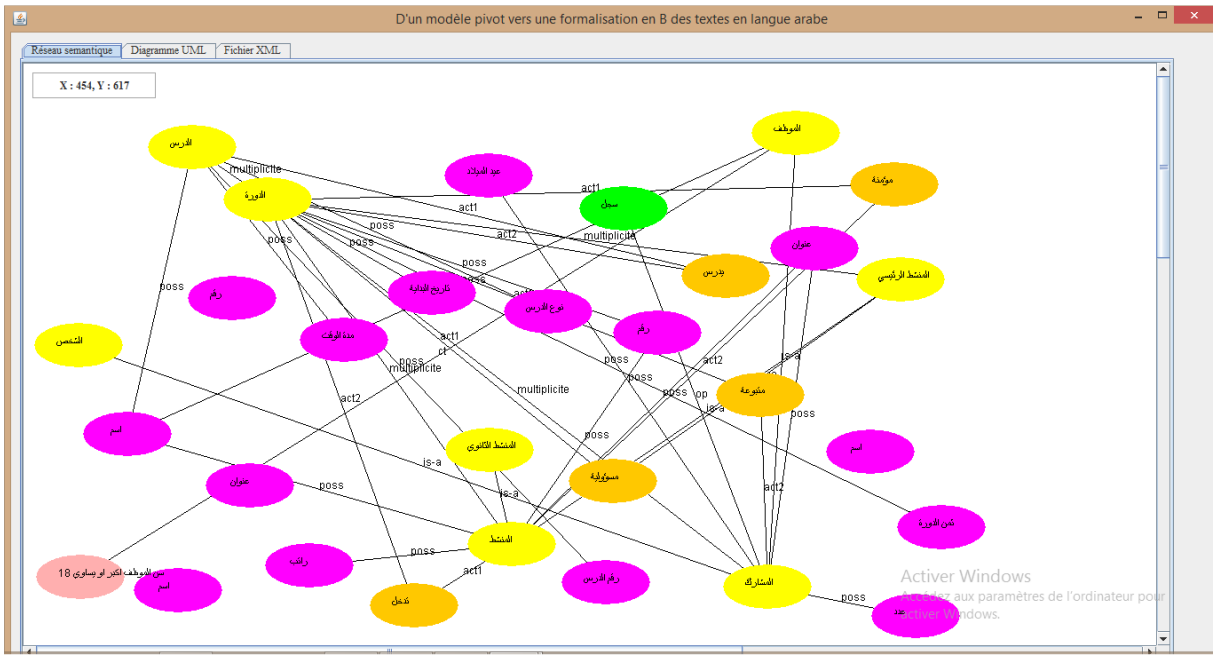


Figure 11. Generation of RSB by Ar2B.

3.3 The Formal Model

Ar2B generates the corresponding class diagram and then transforms it into an XML file for better interoperability. A second passage from UML to formal language Event-B is done. This formal method performs formal refinement in several steps until the final refinement contains enough details for an implementation.

Since we are the pioneers in the B-language formalization of the Arabic language, we were obliged to make an adaptation of the plug-ins for this language for the Rodin platform.
In the meantime, we have studied the specification of a class diagram under the RODIN platform and its implementation in EVENT-B. Machine and context will then be translated into XML (.bum: machine) and (.buc: context) files. To do this, we have deduced rules for switching from UML to EVENT-B.

3.3.1 Extraction of the Concepts of Class Diagram: Semi-formal Model

We have defined a rule database that we have applied to the standard semantic network (RSB) in order to extract the classes, the relations, the cardinalities, the operations and the static constraints of the class diagram. Once the model is generated, we apply to it a set of design patterns: creation design patterns, structure design patterns, in order to verify its conformity, consistency and completeness. Example of rules of transformation from RSN to class diagram

R1: Class concept identification

All the nodes of type "entity" sources of the arc "poss" will be transformed into classes.

R2: Identification of the relationship concept

- All nodes of type "action" will be transformed into relations. The role will be the name of the "action" node.
- All source/target "entity" nodes of an "act" arc will be classes linked by this relation.
- All the nodes of type "entity" linked by an arc "agr/comp" will become classes linked by a relation of composition / aggregation whose component is the target of the arc "agr /comp".
- All target "entity" nodes and "is-a" arc sources will be transformed into an inheritance relationship. The source nodes will be generic class and the target nodes the specialized class.

Figure 12 shows the class diagram generate by Ar2B of the specification given in Section 1.

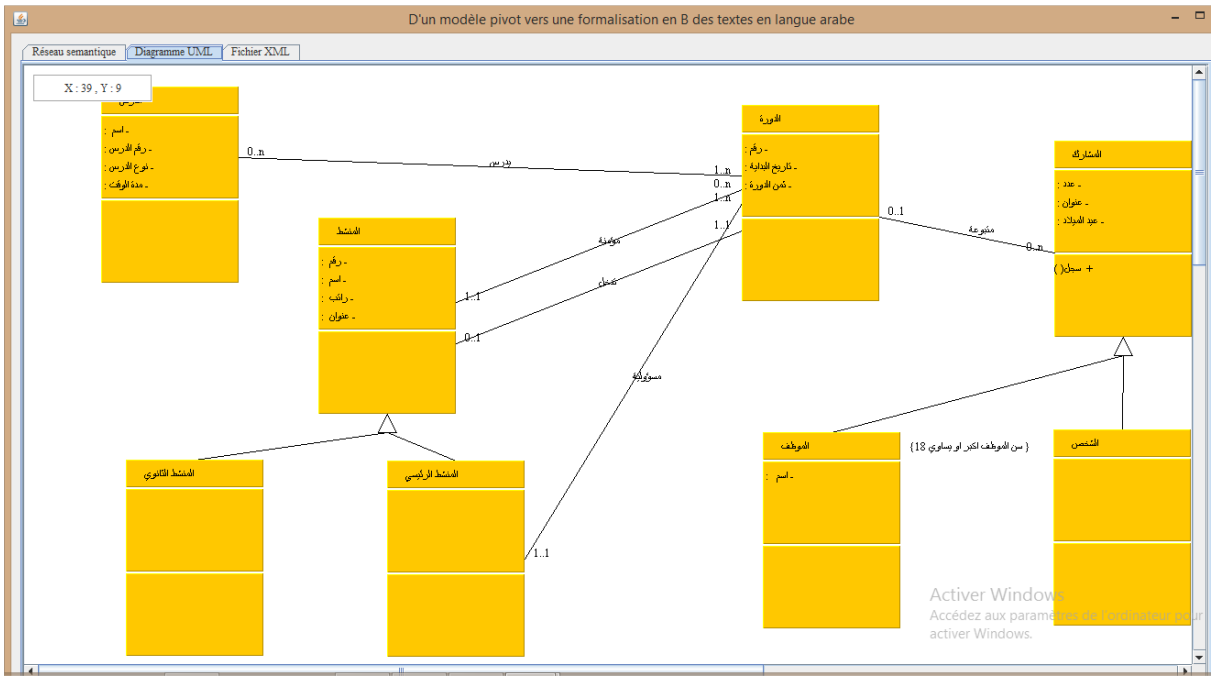


Figure 12. Class diagram generate by Ar2B.

3.3.2 Transformation of the UML Class Diagram to XML Schema

We built the XML schema from the UML model through the concept of Meta-data XML Interchange (XMI) specification, which defines a rigorous approach for generating an XML DTD from a meta-model definition expressed by UML to XML Schema. The transformation rules used in the mapping process are described as follows in [23]. Figure 13 shows the XML file generated by Ar2B of the specification given in Section 1.

Structure RDF | Fichier XML

```
<diagram xmlns="x-schema:classDiagram.xsd">
<class name="الدرس">
<instance-variable>اسم الدرس</instance-variable>
<instance-variable>رقم الدرس</instance-variable>
<instance-variable>نوع الدرس</instance-variable>
<instance-variable>مدة الوقت</instance-variable>
<association multiplicity="one" role-name="يدرس">
<class-name>الدرس</class-name>
</association>
<association multiplicity="many">
<class-name>الدورة</class-name>
</association>
<association multiplicity="one" role-name="مسؤولية">
```

Figure 13. XML file generated by Ar2B.

3.3.3 Extraction of Formal Specification with Event-B

We try to apply the plug-in (XSLT Orange volt) Eclipse to translate our XML through the transformation rules proposed by XSLT and from research work [24] to automatically produce Event-B specifications (.bum / .buc) under the Rodin demonstrator.

Event-B is an extension of the B-method (J-R. Abrial). It uses set theory and logic, is relatively simple and has an extensive tool support. It comes with a new modelling framework called Rodin (like Atelier B tool for the classical B). The Rodin platform is an eclipse-based open and extensible tool for B-model specification and verification.

Unfortunately, forced by the transformation of our UML class diagram into a SOA pattern design diagram and the non-finalization of the transformation rules proposed by the authors, our choice then turned to the UML-B software. These specifications can also produce SQL or JAVA source code after a series of proven refinements proposed by the plug-in UML-B.

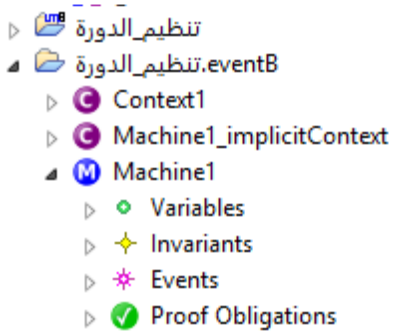
Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

Table 2 summarizes the correspondence between a UML-B (context) specification and that of event-B. It is generated automatically.

Table 2. Correspondence UML-B/event-B.

UML-B	Event-B
- Classe (variable-instances) - Class (fixed-instances) - Class (variable inst and has super class) - Class (fixed inst and has super class)	- Variable ⊆ Set - Set - Variable ⊆SuperClass - Constant ⊆SuperClass
- Attribute (card 0...n -1...1) - Attribute (card 0...n -0...1) - Attribute (card 0...n-0...n) - Etc (try other cardinalities in UML-B)	- Variable ∈ Class →type - Variable ∈ Class ⇸type - Variable ∈ Class ↔type - Etc
- Associations	- As Attribute but Type is another class
- Class Event	- Event (sellf) WHEN self ∈ Class
- Class Constructor	- Event (self) WHEN self ∈ SET\ Class
- Class Invariant	- ∀self. ((self∈ Class) ⇒ Class invariant

Figure 14 shows the formal specification with event_B of the text in Section 1.



MACHINE
Machine1
SEES
Machine1_implicitContext
VARIABLES
منشط // *class instances*
مشارك // *class instances*
دورة // *class instances*
درس // *class instances*
شخص // *class instances*
موظف // *class instances*
رئيسي_منشط // *class instances*
ثانوي_منشط // *class instances*
يتدخل // *attribute of* منشط
المنشط_اسم // *attribute of* منشط
المنشط_رقم // *attribute of* منشط
المنشط_راتب // *attribute of* منشط
المنشط_عنوان // *attribute of* منشط
المشارك_اسم // *attribute of* مشارك
المشارك_رقم // *attribute of* مشارك
المشارك_عنوان // *attribute of* مشارك
الميلاد_عيد // *attribute of* مشارك

Annexe

مسؤولية // *attribute of* دورة
متبوعة // *attribute of* دورة
مؤمنة // *attribute of* دورة
الدورة_رقم // *attribute of* دورة
البداية_تاريخ // *attribute of* دورة
الدورة_ثمن // *attribute of* دورة
يدرس // *attribute of* درس
الدرس_اسم // *attribute of* درس
الدرس_رقم // *attribute of* درس
الدرس_نوع // *attribute of* درس
الوقت_مدة // *attribute of* درس

INVARIANTS

منشط _type : منشط ∈ ℙ (منشط _SET)
مشارك _type : مشارك ∈ ℙ (مشارك _SET)
دورة _type : دورة ∈ ℙ (دورة _SET)
درس _type : درس ∈ ℙ (دروس)
شخص _type : شخص ∈ ℙ (مشارك)
موظف _type : موظف ∈ ℙ (مشارك)
رئيسي_منشط _type : رئيسي_منشط ∈ ℙ (منشط)
ثانوي_منشط _type : ثانوي_منشط ∈ ℙ (منشط)
يتدخل _type : يتدخل ∋ منشط ↔ دورة
المنشط_اسم _type : المنشط_اسم ∋ منشط ⟶ منشطين
المنشط_رقم _type : المنشط_رقم ∋ منشط ⟶ منشطين
المنشط_راتب _type : المنشط_راتب ∋ منشط ⟶ ℕ
المنشط_عنوان _type : المنشط_عنوان ∋ منشط ⟶ منشطين
المشارك_اسم _type : المشارك_اسم ∋ مشارك ⟶ مشاركين
المشارك_رقم _type : المشارك_رقم ∋ مشارك ⟶ ℕ
المشارك_عنوان _type : المشارك_عنوان ∋ مشارك ⟶ مشاركين
الميلاد_عيد _type : الميلاد_عيد ∋ مشارك ⟶ ℕ
مسؤولية _type : مسؤولية ∋ دورة ↔ رئيسي_منشط
متبوعة _type : متبوعة ∋ دورة ↔ مشارك

مؤمنة _type : منشط ↔ دورة ∋ مؤمنة
الدورة_رقم _type : دورات ← دورة ∋ الدورة_رقم
البداية_تاريخ _type : ℕ ← دورة ∋ البداية_تاريخ
الدورة_ثمن _type : ℕ ← دورة ∋ الدورة_ثمن
يدرس _type : دورة ↔ درس ∋ يدرس
الدرس_اسم _type : دروس ← درس ∋ الدرس_اسم
الدرس_رقم _type : دروس ← درس ∋ الدرس_رقم
الدرس_نوع _type : دروس ← درس ∋ الدرس_نوع
الوقت_مدة _type : دروس ← درس ∋ الوقت_مدة

EVENTS
INITIALIZATION ≙
STATUS
ordinary
BEGIN
منشط _init : ∅ =: منشط
مشارك _init : ∅ =: مشارك
دورة _init : ∅ =: دورة
درس _init : ∅ =: درس
شخص _init : ∅ =: شخص
موظف _init : ∅ =: موظف
رئيسي_منشط _init : ∅ =: رئيسي_منشط
ثانوي_منشط _init : ∅ =: ثانوي_منشط
يتدخل _init : ∅ =: يتدخل
المنشط_اسم _init : ∅ =: المنشط_اسم
المنشط_رقم _init : ∅ =: المنشط_رقم

Annexe

المنشط_راتب _init : ∅ =: المنشط_راتب
المنشط_عنوان _init : ∅ =: المنشط_عنوان
المشارك_اسم _init : ∅ =: المشارك_اسم
المشارك_رقم _init : ∅ =: المشارك_رقم
المشارك_عنوان _init : ∅ =: المشارك_عنوان
الميلاد_عيد _init : ∅ =: الميلاد_عيد
مسؤولية _init : ∅ =: مسؤولية
متبوعة _init : ∅ =: متبوعة
مؤمنة _init : ∅ =: مؤمنة
الدورة_رقم _init : ∅ =: الدورة_رقم
البداية_تاريخ _init : ∅ =: البداية_تاريخ
الدورة_ثمن _init : ∅ =: الدورة_ثمن
يدرس _init : ∅ =: يدرس
الدرس_اسم _init : ∅ =: الدرس_اسم
الدرس_رقم _init : ∅ =: الدرس_رقم
الدرس_نوع _init : ∅ =: الدرس_نوع
الوقت_مدة _init : ∅ =: الوقت_مدة

END
END

Figure 14. Specification with event_B of class diagram obtained.

4. EXPERIMENTS AND RESULTS

In order to evaluate Ar2B, experiments were performed on a corpus consisting of a collection of texts in Arabic by maintaining existing potential diacritics. The corpus contains about 51404 words, including 81 Arabic text, 899 paragraphs, 3871 sentences and 29188 words. The sentence can contain upto 25 words. We have taken texts from the practical exercises of the 'software engineering' course taught to 3rd year students in computer science of our university. We also translated specification texts that we took from other universities, websites and books. A list of text is available at: https://sites.google.com/site/kheirazinebbousmeha/corpus.

We have put our first results on the graph in Figure 15. We have, for a text comprising only simple sentences, an f-measure greater than 93 % in the generation of the class diagram. The f-measures of each concept were in the order of 95 % for the extraction of class and more than 92 % for the extraction of attribute, operation and relation. These f-measures would decrease as the sentence became more complicated, containing negative forms (f-measure=63.3825%), anaphoras and ellipses (f-measure=41.3825 %) or a complex formulation.

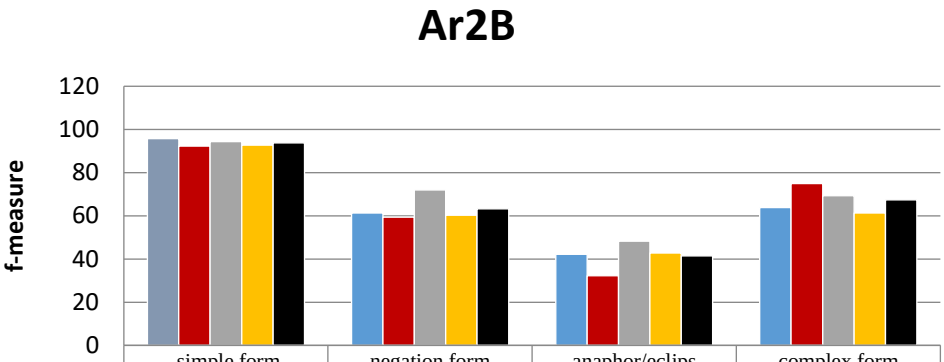
The greatest values were observed in the extraction of classes and attributes, because we used, in addition to the linguistic rules, statistical measures. The lowest rate was that of the generalization and specialization relation. This type of relationship has a schema similar to the action-type schema. It can be referred to in nominal and in verbal sentences. Sometimes, the verb type is ambiguous. This problem has been circumvented in other languages (English and recently in French) by the use of verbnet lexicon, where it is possible to use a syntactic construct to match an argument of a verb to semantic roles [25].

We use the confusion matrix in order to analyze the error rate of each concept. According to the values reported in this table, the overall error rate is:

$$E = 1 - (\sum_{i=1}^{m} n_{ii} / \sum_{i=1}^{m} n_{ij}) = 23.38\%. \quad (5)$$

This matrix reveals the distribution of the error for each concept of the diagram on each rate found. We note that for the Attribute-identifier concept for example, 2.5% is erroneously classified as classes and 9.1% are classified, also by mistake, as attributes. This can be explained by a bad formulation of the

rules concerning the extraction of the identifier attribute. This measurement allowed us to locate the error and review certain points of the design.



	simple form	negation form	anaphor/eclips	complex form
class	95.75	61.25	42.24	63.85
attribute	92.33	59.38	32.25	75
operation	94.43	72.03	48.26	69.32
relation	92.68	60.23	42.78	61.35
f-measure	93.7975	63.2225	41.3825	67.38

Figure 15. The f-measures obtained by Ar2B for each type of text.

5. CONCLUSIONS

We presented a platform for ANLP devoted to a text language processing treating the functional specifications and more specifically in the general functional specifications (GFSs).

Ar2B encompasses a set of coherent modules and an automatic event-B formalization of conceptual modeling based on hybrid approaches and reliable tools. It processes a large number of specifications in a shorter time and in a less subjective way than an expert.

The realization of the conceptual phase by "chunking" simplifies sentences. The idea of classifying sentences using Fillmore's case theory allowed us to disambiguate the different interpretations that a sentence may contain. The proposed hybrid approach based on linguistic rules and statistical methods allowed us to generate the relevant concepts of the future semi-formal model.

As for semi-formalization, the proposed approach makes it possible to take into account, through a set of heuristics and then design patterns, the automatic passage of the text represented by the standardized semantic network into a UML class diagram.

We used semantic networks for unambiguous semantic interpretation of specifications. The use of ontologies significantly improves the quality of the specified requirements. Their use as an intermediate representation in a process of automatic formalization of natural language specifications has been explored only recently [2]. The lack or absence of domain ontology devoted to the Arabic language has led us to use semantic networks that respond well to the specificities of our field of application.

Regarding the formalization in event-B and since there is no work on this formalization in Arabic, many difficulties were encountered among which we quote the installation of the platform Rodin and UML-B software as well as the adaptation of the plug-ins for the Arabic language.

We chose to expand our platform environment of the open source platform Rodin Version 3.2.0-ecacdcb; an IDE based on Eclipse for event-B provides an effective support for refinement and mathematical proof. The platform contributes through Eclipse and can be extended with plug-ins.

The results obtained are promising and have reached f-measures of around 70% for all types of sentences and up to 93% for the treatment of simple sentences. For the treatment of anaphoras and ellipses, we plan to integrate the research of our ANLP-RG and take into account the treatment of synonymy by using Ontology AWN as well as to complete the UML model by the OCL constraint expression language for more processing constraints.

ACKNOWLEDGEMENTS

The authors would like to thank the ANLP-RG research team as well as the master students from the Computer Science Department of the University Oran1 for their efforts in designing the Ar2B platform. They would also like to thank Mr. MEZIANE Farid, Professor at the University of Salford Manchester (United Kingdom) for his wise advice. He is the pioneer in the formalization of information system specifications from English texts. His great experience has been very useful in carrying out this work. It is noted through the examples given that the name of the application is different, because each student group of our masters had the programming of a module of the platform.

REFERENCES

[1] W. Chama, R. Elmansouri and A. Chaoui, "Modeling and Verification Approach Based on Graph Transformation," Lecture Notes on Software Engineering, vol. 1, no. 1, pp. 39-43, 2013.

[2] D. Sadoun, Des Spécifications en Langage Naturel aux Spécifications Formelles *vi*a une Ontologie Comme Modèle Pivot, Ph.D. Thesis, Diss. Université, Paris, Sud-Paris XI, 2014.

[3] M. Ilieva and H. Boley, "Representing Textual Requirements as Graphical Natural Language for UML Diagram Generation," Proc. of the International Conference on Software Engineering and Knowledge Engineering (SEKE'08), pp. 478–483, 2008.

[4] L. Kof, "Requirements Analysis: Concept Extraction and Translation of Textual Specifications to Executable Models," Proc. of the International Conference on Application of Natural Language to Information Systems, pp. 79-90, 2009.

[5] I. S. Bajwa, B. L. Bordbar and G. Mark, "OCL Constraints Generation from Natural Language Specification," Proc. of the 14th IEEE International Conference on Enterprise Distributed Object Computing (EDOC), pp. 204-213, 2010.

[6] O. Keszocze, M. Soeken, E. Kuksa and R. Drechsler, "Lips: An IDE for Model-driven Engineering Based on Natural Language Processing," Proc. of the 1st International IEEE Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE), San Francisco, CA, USA, 2013.

[7] W. F. Tichy and S. J. Koerner, "Text to Software: Developing Tools to Close the Gaps in Software Engineering," Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, pp. 379-384, 2010.

[8] R. Laleau and A. Mammar, "An Automatic Generation of B Specifications from Well-defined UML Notations for Database Applications," Proc. of the International Symposium on Programming and Systems (ISPS), Algers, Algérie, 2001.

[9] A. J. Fougéres and P. Trigano, "Rédaction de Spécifications Formelles: élaboration à Partir des Spécifications écrites en Langage Naturel," Cognito-Cahiers Romans de Sci. Cognitives, pp. 29-36, 1997.

[10] F. Mokhtari and M. Badri, "Generating Maude Specifications From UML Use Case Diagrams," Journal of Object Technology, vol. 8, no. 2, pp. 119–136, 2009.

[11] E. Mit, Developing VDM++ Operations From UML Diagrams, Ph.D. Thesis, School of Computing, Science and Engineering University of Salford, U.K, 2007.

[12] F. Meziane and S. Vadera, "Artificial Intelligence in Software Engineering: Current Developments and Future Prospects," Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects, pp. 24-29, 2010.

[13] H. H. Ammar, W. Abdelmoez and M. S. Hamdi, "Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems," Proc. of the IEEE International Conference on Communications and Information Technology, Al-Madinah Al-Munawwarah, Saudi Arabia, pp. 24-29, 2012.

[14] N. Arman and S. Jabbarin, "Generating Use Case Models from Arabic User Requirements in a Semi-automated Approach Using a Natural Language Processing Tool," Journal of Intelligent Systems, vol. 24, no. 2, pp. 277-286, 2015.

[15] N. Alami, N. Arman and F. Khamyseh. "A Semi-automated Approach for Generating Sequence Diagrams from Arabic User Requirements Using a Natural Language Processing Tool," Proc. of the 8th IEEE International Conference on Information Technology (ICIT), Amman, Jordan, 2017.

[16] I. Nassar and F. Khamayseh, "A Semi-automated Generation of Activity Diagrams from Arabic User Requirements," NNGT International Journal on Software Engineering, vol. 2, 2015.

[17] T. S. Hoang, "An Introduction to the Event-B Modelling Method," In Book: Industrial Deployment of System Engineering Methods, Publisher: Springer-Verlag, Editors: Alexander Romanovsky and Martyn Thomas, 2013.

[18] A. Guissé, F. Lévy and A. Nazarenko, "From Regulatory Text to BRMS: How to Guide the Acquisition of Business Rules," Proc. of the International Workshop on Rules and Rule Markup Languages for the Semantic Web, Springer, pp. 77-91, 2012.

[19] K. Z. Bousmaha, M. K. Rahmouni, B. Kouninef and L. Belguith Hadrich, "A Hybrid Approach for the Morpho-Lexical Disambiguation of Arabic," Journal of Information Processing Systems (JIPS), vol. 12, no. 3, pp. 358-380, 2016.

[20] S. Boulaknadel, B. Daille and D. Aboutajdine, "A Multi-word Term Extraction Program for Arabic Language," Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 1485-1488, Morocco, 2008.

[21] M. Diab, "Second Generation AMIRA Tools for Arabic Processing Fast and Robust Tokenization, POS tagging and Base Phrase Chunking," Proc. of the 2nd International Conference on Arabic Language Resources and Tools, pp. 285-288, Egypt, 2009.

[22] F. Z. Belkredim and A. El Sebai, "An Ontology-based Formalism for the Arabic Language Using Verbs and Their Derivatoion," Communications of the IBIMA, vol. 11, pp. 44-52, 2009.

[23] J. Singh, Mapping UML Diagrams to XML, Doctoral Dissertation, University New Delhi, 2003.

[24] I. Tounsi, H. Zied, M. H. Kacem, A. H. Kacem and K. Drira, "Using SOAml Models and Event-B Specifications for Modeling SOA Design Patterns," Proc. of the International Conference on Enterprise Information Systems (ICEIS), 2013.

[25] L. Danlos, T. Nakamura and Q. Pradet, "Vers la Création d'un Verbnet du Français," Proc. of the 21ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), Atelier Fondamen TAL, 2014.

ملخص البحث:

إن تحويل متطلبات البرمجيات الطبيعية إلى مواصفات أكثر رسمية أمر ليس بالسهل، ويمكن أن يكون تطبيقاً ممتازاً لمعالجة اللغات الطبيعية. هذه المشكلة ليست حديثة، وقد أثارت وما زالت نثير اهتماماً كبيراً؛ لأنها تفتح الباب أمام العديد من التحديات في مجالات علمية متنوعة، مثل: المعالجة الآلية للّغات، وهندسة المتطلبات، وتمثيل المعرفة، والتحقق الرسمي. تقترح هذه الورقة منصّة واستراتيجية لتحويل متطلبات البرمجيات المعينة الى مواصفات رسمية باستخدام الحدث – ب (event-B). والجدير بالذكر أن النصوص المستخدمة في هذا البحث هي نصوص باللغة العربية، الأمر الذي يُعدّ تحدّياً حقيقياً. وقد تم بناء نظام (Ar2B) واختباره، وحققت التجارب نتائج جيدة بدقة بلغت 70%.