Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 6, December 2024, pp. 4378∼4387
ISSN: 2302-9285, DOI: 10.11591/eei.v13i6.7290

A novel agile method for user stories' XMI model generation via NLP and MDA

Nassim Kharmoum[1,2], Sara Retal[3], Mouna Hajjaj[4], Souad Najoua Lagmiri[5], Yassine Rhazali[6]
[1]National Center for Scientific and Technical Research (CNRST), Rabat, Morocco
[2]Intelligent Processing and Security of Systems (IPSS) Research Team, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco
[3]Computer Science, Artificial Intelligence, and Cybersecurity (2IACS) Laboratory, Higher Normal School of Technical Education (ENSET), Hassan II University of Casablanca, Mohammedia, Morocco
[4]Research Laboratory in Management and Development (LMRD), Department of Management, Faculty of Economics and Management, Hassan First University of Settat, Settat, Morocco
[5]Information Technology, Networking, Security and Management (IRSM) Laboratory, Higher Institute of Management Administration and Computer Engineering, Rabat, Morocco
[6]Information and Communication Systems Engineering (ISIC) Research Team of Higher School of Technology, Mathematical and Computer Modeling Laboratory of ENSAM, Moulay Ismail University, Meknes, Morocco

Article Info

Article history:

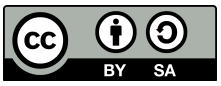
Received Jul 30, 2023
Revised Apr 14, 2024
Accepted Jun 1, 2024

Keywords:

Automatic generation
Meta-model
Model driven architecture
Natural language processing
User stories
XML metadata interchange

ABSTRACT

Agile software development methodologies have grown in popularity during the past few years. One of the key components of agile development is the use of user stories to describe software requirements. However, creating and managing user stories can be time-consuming and error-prone. In this paper, we present a novel method to generating user stories' XMI model using natural language processing (NLP) and model-driven architecture (MDA) approach. We developed a method that uses NLP to extract key information from user stories and then applies MDA techniques to generate an XMI model conforming to its proposed meta-model. We conducted a case study to illustrate and validate our method, and we analyze and discuss the studied-related work with our proposal. As a result, our method has the potential to make user stories' models and their meta-models the focus of software development. This will help to streamline the development process by making it easier to construct and transform models in an agile environment with the MDA approach.



Corresponding Author:

Nassim Kharmoum
National Center for Scientific and Technical Research (CNRST)
Rabat, Morocco
Email: nkharmoum@gmail.com

1. INTRODUCTION

Agile software development methodologies have become increasingly popular in recent years due to their flexibility and adaptability. Manifesto [1], a set of values and principles for agile development, emphasizes the importance of responding to change, delivering working software frequently, and collaborating with customers and stakeholders to meet their needs [2]. One of the key components of agile development is the use of user stories to describe the software requirements.

User stories are a simple yet powerful way to describe software requirements from the user's perspective. A user story typically consists of three parts: a brief description of the user role, the specific functionality

or feature required, and the business value for the user. User stories are usually written on index cards or sticky notes, which are then organized and prioritized in a product backlog. While user stories are effective in capturing the user's needs and goals, the process of creating and managing them can be time-consuming and error-prone. Furthermore, as the number of user stories increases, it can become challenging to manage them effectively. To address these challenges, several approaches have been proposed, including the use of templates, tools, and techniques such as acceptance test-driven development (ATDD) [3] and behavior-driven development (BDD) [4].

In this paper, we present a novel method to generate user stories' XMI model [5] using natural language processing (NLP) [6] and model-driven architecture (MDA) approach [7]. Our method aims to automate the process of creating and managing user stories, while conforming to its proposed meta-model [8]. We focus on the MDA approach for our method because it is supported by the object management group (OMG) and is the most commonly used approach to model driven engineering (MDE) [9]. MDE is a software engineering paradigm that considers models and their meta-models as the main entities in the software systems development process. MDA is also aligned with the Agile manifesto, which emphasizes iterative and incremental development. MDA is regarded as a technological advancement for the software industry's future, with the aim of automating the development process of any information system. MDA provides its attributes, as evidenced by its advocacy for adhering to multiple guidelines and standards [10]. Thus, we put the focus on the generation of the XMI model format, as it is an industry-standard format for exchanging and sharing metadata between software tools and systems [5]. XMI is based on eXtensible markup language (XML) [11] and is used primarily in the context of MDA approach, and XMI model format plays a crucial role in simplifying the exchange of model information between software development tools, enabling efficient collaboration, and promoting MDA practices.

We also, based on the NLP which is a branch of artificial intelligence and computational linguistics [12] that focuses on the interaction between computers and human language. NLP is the study and creation of models and algorithms that allow computers to meaningfully read, comprehend, and produce human language. NLP techniques involve a combination of linguistics, statistics, and machine learning. Some common approaches used in NLP include text analysis to involve the application of various NLP techniques to analyze and extract meaningful information from textual data.

To delve into the core of our study, the remainder of the paper is structured as follows. The next section provides a summary of the related work. Then, we describe the system architecture, the NLP and MDA techniques used, and the XMI model generated. We then we present a case study conducted to validate and illustrate the proposed method. After that, we analyze and discuss all obtained results. Finally, we conclude the paper by discussing the implications of our method and identifying future research directions.

2. RELATED WORK

Few studies have been proposed for automating the process of generating and managing user stories in agile software development projects, dealing with NLP language and MDA approach. Some of the most relevant studies are discussed below. First, Elallaouia *et al.* [13] propose a process of transforming the unified modeling language (UML) use case diagram from user stories according to the MDA approach. The transformation is done automatically, using the Tree Tagger parser's NLP technique. In other work, Allala *et al.* [14] present a method to automatically transform user requirements in XMI format (XML metadata interchange) to test cases using model-driven engineering and NLP to ensure the automatic transformation. Baïna *et al.* [15], suggests enhancing the agile iterative business process design methodology by integrating a model-driven approach with an automation assistant. This assistant would convert user stories and textual requirements into a business process model and notation (BPMN) [16], the automation is done dealing with their XPDL2MDE Eclipse plug-in.

Moreover, Georges in [17], extracts the feature and variability from Agile specifications expressed by the epics and user stories and their related source code for software product line migration based on relational concept analysis and formal concept analysis. This method is implemented using the NLP techniques. Gupta *et al.* [18] propose a BDD approach that automates the generation of a simplified version of UML diagrams: class diagram, use case diagram, State machine diagrams, and activity diagram from the user stories and their acceptance criteria. The generation is done automatically with Python language and an NLP-based tool. Nasiri *et al.* [19] propose an automated method to extract UML activity diagram represented by business process

models from requirements, which are presented as acceptance criteria of the user stories. The extraction is done automatically by dealing with languages Prolog, Python, and NLP using Stanford coreNLP tool.

After studying the different works done. We have that all papers based on the agile user stories as an input of their methods to generate automatically new out that can be a graphical or a textual model. But few studied papers deal with MDA approach to drive the out generation. Therefore, all approaches rely on a case study to assess their methodologies. So, in this paper, we propose a new method that automatically generate a new structured and standardized representation of user stories in XMI format from a textual one, the objective is to support the use of user stories in the MDA approach.

3. PROPOSED METHOD

The proposed method will be part of our Agile user stories' driven method project, which aims to automate the generation of new models in different MDA's abstraction level form user stories' XMI model. To do so, in this contribution, we generate user stories' XMI model using a combination of NLP techniques and MDA approach. The method streamlines the process of creating and managing user stories in agile software development projects, reducing manual effort and improving the quality of requirements representation. We emphasize that our method will benefit from the experience we gained while generating models and proposing a novel user stories meta-model [20], [21] in the MDA approach and agile context.

3.1. User stories' meta-model

In this method, we based on a part of our proposed user stories' meta-model in the MDA approach [8] Figure 1, which is considered as the backbone of this contribution. The main three parts of user stories will be the main elements for generating user stories' XMI models in compliance with the proposed meta-model Figure 1, we have as classes: "UserStoryModel", "Backlog", "Epic", "UserStory", "UserStoryDescription", "UserRole", "Function", and "BusinessValue".

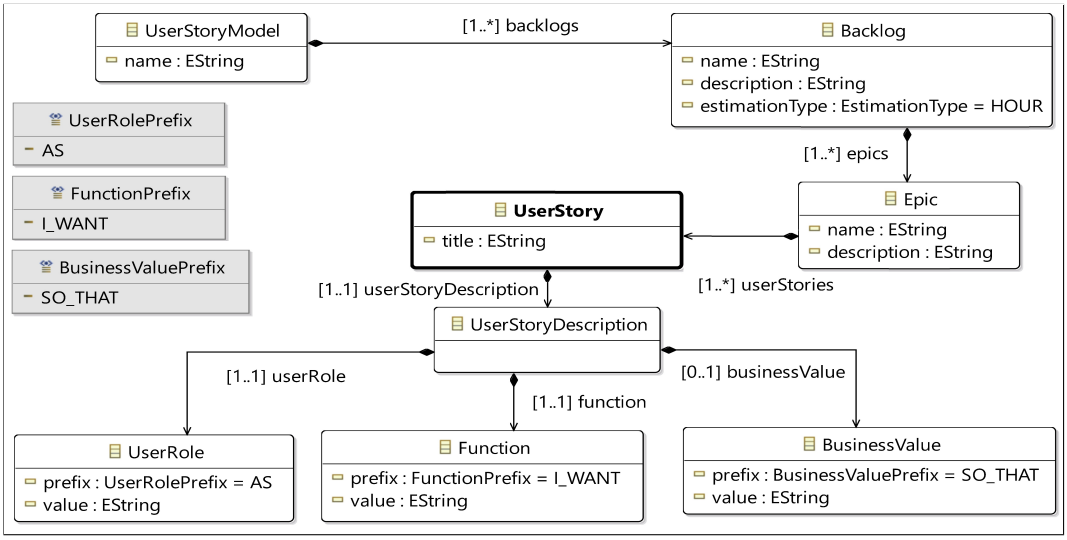


Figure 1. The proposed meta-model

3.2. Input user stories textual model

The input module receives user stories in natural language format. User stories typically follow a structured format that includes the "user role", "function", and "business value". The input module handles the ingestion of these user stories for further processing (Figure 2(a)).

3.3. User stories' validation, extraction, and generation with NLP

The NLP module employs various techniques to extract the essential information from the user stories [22]. We deal with spaCy, which is an open-source Python library for NLP [23]. Spacy library includes tokenization, part-of-speech tagging, lemmatization, and dependency parsing. Tokenization breaks the text into individual words or tokens, while part-of-speech tagging assigns grammatical labels to each word. Named

entity recognition identifies important terms like proper nouns or technical terms, and dependency parsing analyzes the relationships between words in the sentence (Figure 2(b)). For our study we use the "en_core_web_sm" which is a pre-trained English language model in the spaCy library. So, the steps to generate our user stories' XMI model are:

- User stories validation: the first step is to validate the user stories input model structure. We check for the correctness of the following mandatory parts "User Role" and "Function" and their prefix; like "As" for "User Role" and "I Want" or "I can" for the "Function", after that we check the third option part which is "Business Value" and their prefix like "So that". To make our method generic, we propose a list of the most commonly used prefixes for each part of user stories. Also, for our proposal, we take just affirmative user stories.
- User stories extraction: in this step, we deal with just the validated user stories with affirmative form. we use the spaCy's matchers to keep the "User role", "Function", and "Business value" based on their "prefixes".
- User stories'XMI generation: in this step an XMI file will be created and conforms to the proposed method, to do so, the three extracted user stories parts will be successively at the XMI node: "UserRole", "Function", and "BusinessValue". Then, we group all user stories by their "UserRole" as an epic. At the end, all epics are grouped in one "Backlog" that has the name of the studied system (Figure 2(c)).

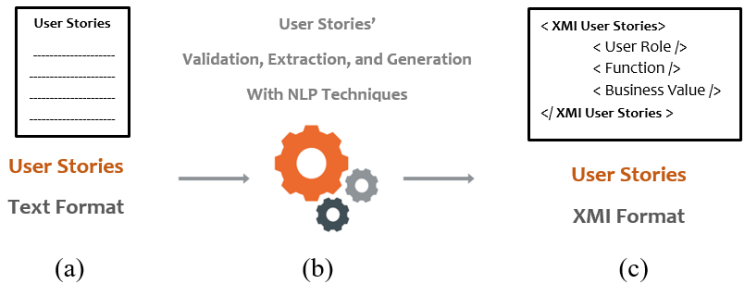


Figure 2. Our proposed method; (a) source user stories' text format, (b) user stories' validation, extraction, and generation, and (c) generated user stories' XMI format

3.4. Generated user stories' XMI model based on MDA approach

The generated user stories' XMI model it conforms to its proposed meta-model and represents the user stories in a structured and standardized format, making it easier to understand and analyze the requirements. The MDA techniques include employing the UML to define the structure and relationships between the user stories, using the object constraint language (OCL) to establish constraints and rules, and employing the model transformation language (MTL) to transform the extracted information into the XMI model.

3.5. Proposed user stories processing system

To facilitate the automation of the user stories' XMI model, we have proposed a user-friendly and ergonomic system using the Python streamline library [24] which is a data driven web application library based on Python. The purpose is to design and enhance the management and visualization of user stories in agile software development projects in a simple web interface. Leveraging the power of the generated XMI model, the system provides an intuitive and interactive interface for stakeholders, product owners, and development teams to view, analyze, and collaborate on the user stories. With this web interface, teams can efficiently track the progress of user stories, visualize backlog items, and ensure alignment with stakeholder needs, ultimately enhancing the efficiency and effectiveness of the agile development process.

In summary, the proposed method offers several advantages. Firstly, it auto-mates the extraction of key information from user stories, reducing the risk of errors and inconsistencies that may occur during manual processing. Secondly, it employs MDA techniques to generate a standardized and structured XMI model, enhancing the clarity and understanding of the requirements. Lastly, the method has a user-friendly and ergonomic web interface, and it is flexible and adaptable, making it suitable for various agile software development projects. By automating the process of generating user stories' XMI model using NLP and MDA, the proposed method significantly reduces the time and effort required, while improving the quality of the requirements' representation in agile software development projects.

4. CASE STUDY

To validate the proposed method, we conducted a case study in a real-world software development project. The case study aimed to illustrate all the proposed method process to generate the user stories' XMI from a simple user stories text file. The case study involved an e-commerce website, and the team consisted of "new user", "customer", and "webmaster" involved in the requirements gathering and management process. The new user can create an account on the website to be a new customer. A customer can search for products on the website, add and purchase items to the shopping cart, also he can track the status of any order and has a comprehensive analytics dashboard. However, the webmaster can manage and ensure all the website functionalities and content.

In the experimental approach, our system was used to automatically extract key information from user stories and generate the XMI model. The team members entered the user stories in natural language format, and the system applied NLP techniques to extract the relevant information. The MDA module then transformed the extracted information into an XMI model, which was displayed in a human-readable format.

4.1. User stories' text

In this case study, we propose 10 validated user stories in a textual format for the e-commerce website (see Table 1). These stories outline various users interactions and scenarios aimed at improving the website's functionality and their Business values. We selected the e-commerce platform as a simple example, allowing a better understanding of the proposed method.

Table 1. Input user stories in textual format

Id	User stories
1	As a new user, I want to create an account on the website, so that I can access personalized features and information
2	As a customer, I want to search for products on the website, so that I can find items that meet my specific requirements
3	As a customer, I want to add products to my shopping cart, so that I can keep track of the items I want to purchase
4	As a customer, I want to complete the purchase of the items in my shopping cart, so that I can receive the products and make the payment
5	As a customer, I want to track the status of my order, so that I can know when to expect the delivery
6	As a webmaster, I want to have a comprehensive analytics dashboard, so that I can track and analyze website performance and user behavior
7	As a webmaster, I want to be notified of broken links on the website, so that I can fix them promptly and ensure a smooth user experience
8	As a webmaster, I want to have a streamlined content publishing workflow, so that I can manage the creation, review, and publication of the website content
9	As a webmaster, I want to ensure that the website meets accessibility standards, so that it can be accessed and used by all users
10	As a webmaster, I want a seamless deployment process and version control system, so that I can easily manage and deploy updates to the website

4.2. Generated XMI user stories

The generated user stories' XMI, as illustrated in Figure 3, presents the target model. This XMI format facilitates the presentation of user stories through an XML-based UML metadata information exchange, adhering to the standards set forth by the OMG. By utilizing this standardized approach, the exchange of information between systems becomes more seamless and interoperable, promoting efficiency and consistency in software development processes.

4.3. Proposed user stories processing system

We also develop a web interface with Streamlit library to facilitate the automation of the user stories' XMI model. The proposed interface allows us to select the text file that contains the list of user stories, also allows us to select the prefixes for "User Role", "Function", and "Business Value", thus via this web interface we insert the metadata of the generated XMI model like "XMI Name Space" and "XMI MetaModel Schema Location". During the user stories processing, this interface visualizes the validate and the non-validate user stories also allows us to visualize and download the generated user stories' XMI model (Figure 4).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<usmm:UserStoryModel xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:usmm="www.usmm.com" xmi:version="2.0"
xsi:schemaLocation="www.usmm.com ../metaModel/userStoriesMetaModel" name="E-Commerce Model">
  <backlogs name="E-Commerce backlog" description="E-Commerce web application">
    <epics name="new user" description="User Stories of User Role : new user">
      <userStories title="create an account on the website">
        <userStoryDescription>
          <userRole value="new user"/>
          <function value="create an account on the website"/>
          <businessValue value="i can access personalized features and informationrn"/>
        </userStoryDescription>
      </userStories>
    </epics>
    <epics name="customer" description="User Stories of User Role : customer">
      <userStories title="search for products on the website">
        <userStoryDescription>
          <userRole value="customer"/>
          <function value="search for products on the website"/>
          <businessValue value="i can find items that meet my specific requirementsrn"/>
        </userStoryDescription>
      </userStories>
      <userStories title="add products to my shopping cart">
      <userStories title="complete the purchase of the items in my shopping cart">
      <userStories title="track the status of my order">
    </epics>
    <epics name="webmaster" description="User Stories of User Role : webmaster">
      <userStories title="have a comprehensive analytics dashboard">
        <userStoryDescription>
          <userRole value="webmaster"/>
          <function value="have a comprehensive analytics dashboard"/>
          <businessValue value="i can track and analyze website performance and user
          behaviorrn"/>
        </userStoryDescription>
      </userStories>
      <userStories title="be notified of broken links on the website">
      <userStories title="have a streamlined content publishing workflow">
      <userStories title="ensure that the website meets accessibility standards">
      <userStories title="a seamless deployment process and version control system">
    </epics>
  </backlogs>
</usmm:UserStoryModel>
```

Figure 3. Generated user stories in XMI format

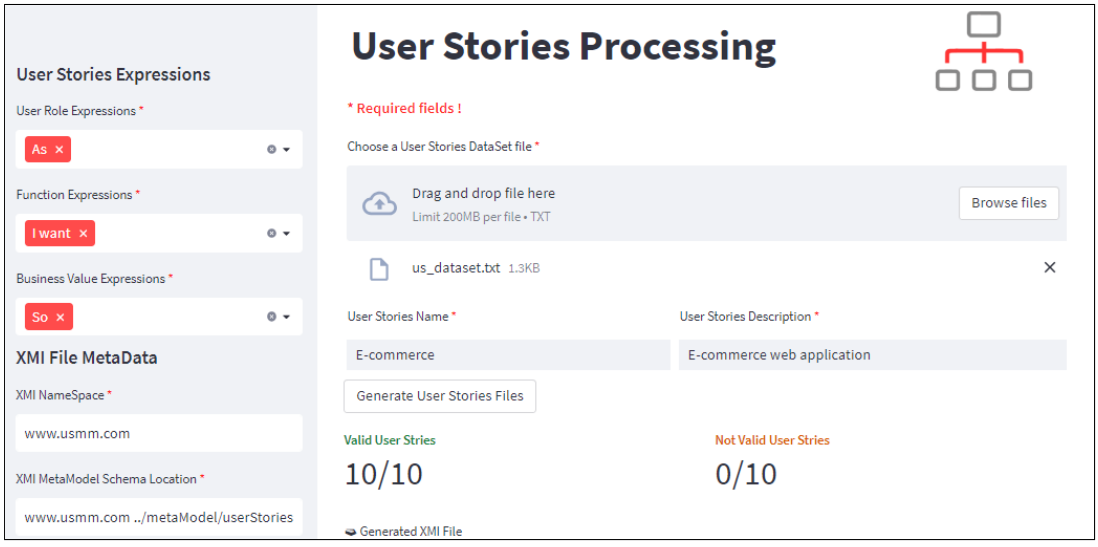


Figure 4. Proposed user stories processing system

5. RESULTS AND DISCUSSION

In this section, we analyze, discuss, and evaluate our proposed method. We have reviewed the related work and identified six methods that are relevant to our research. To define our evaluation criteria, we focus on

the OMG guidance and subsequent work [25]. Our final criteria are based on the input and generated model construction, the model alignment, the NLP and MDA coverage, and the use of assessment methodology. Tables 2–5 shows our evaluation of these methods, as well as our own proposal.

Table 2. Studied papers comparison (input model)

Studied papers	Agile approach	Type	Representation	NLP coverage	Assessment method
[13]	Yes	Textual representation	User stories	Yes	Case study
[14]	Yes	Textual representation	XMI format of user requirements user stories and use case	Yes	Case study
[15]	Yes	Textual representation	User stories		Case study
[17]	Yes	Textual representation	Epic and user stories	Yes	Case study
[18]	Yes	Textual representation	User stories and acceptable criteria	Yes	Case study
[19]	Yes	Textual representation	Acceptable criteria	Yes	Case study
Proposed method	Yes	Textual representation	User stories	Yes	Case study

Table 3. Studied papers comparison (generated model)

Studied papers	Agile approach	Type	Representation	NLP coverage	Assessment method
[13]	Yes	Graphical representation	UML use case diagram	Yes	Case study
[14]	Yes	Textual representation	Test case code	Yes	Case study
[15]	Yes	Graphical representation	BPMN		Case study
[17]	Yes	Textual representation	Source code	Yes	Case study
[18]	Yes	Graphical representation	Simplified version of UML diagrams (class, use case, state machine, and activity)	Yes	Case study
[19]	Yes	Graphical representation	UML activity diagram	Yes	Case study
Proposed method	Yes	Textual representation	XMI format of user stories	Yes	Case study

Table 4. Studied papers comparison (model alignment)

Studied papers	Agile approach	Type	Representation	NLP coverage	Assessment method
[13]	Yes	Automatic	NLP & Java	Yes	Case study
[14]	Yes	Automatic	NLP (stanford CoreNLP)	Yes	Case study
[15]	Yes	Automatic	Eclipse plug-in XPDL2MDE		Case study
[17]	Yes	Automatic	NLP	Yes	Case study
[18]	Yes	Automatic	NLP-based tool BDD	Yes	Case study
[19]	Yes	Automatic	NLP (stanford CoreNLP)	Yes	Case study
Proposed method	Yes	Automatic	NLP (Python)	Yes	Case study

As a result, all the studied papers have a common focus on the Agile approach for their proposed methods. All input models are textual in nature to express the users' requirement in a structured format, the majority of studied method as well as our proposal based on the Agile user stories except the authors of [19] based on the acceptance criteria for their input model. However, for the models' generation some authors generate a graphical representation for their models like Elallaoui *et al.* [13], they generate a UML use case diagram. Baïna *et al.* [15] generate a BPMN model, for [18] generate a simplified version of the UML diagrams (class, use case, state machine, and activity), also on [19] generate a UML activity diagram. Always on the generated model, the rest of authors generate their models with a textual representation, [14] generate a test case code, and [17] generates a source code. For our approach we generate an XMI format of user stories which is a standardized way to exchange user stories model information between different software tools and platforms, enabling efficient collaboration, and promoting MDA practices.

Table 5. Studied papers comparison (MDA approach)

Studied papers	Agile approach	MDA coverage	Meta-model coverage	NLP coverage	Assessment method
[13]	Yes			Yes	Case study
[14]	Yes	Yes	Yes	Yes	Case study
[15]	Yes	Yes	Yes		Case study
[17]	Yes	Yes	Yes	Yes	Case study
[18]	Yes			Yes	Case study
[19]	Yes			Yes	Case study
Proposed method	Yes	Yes	Yes	Yes	Case study

In addition, all the studied methods generate models automatically, which aligns with the criteria. For the model alignment implementation, all authors use the NLP techniques to automate their model alignment, except in [15] the authors based on Eclipse plug-in XPDL2MDE to generate the BPMN model. Besides, just [14], [15], [17], and our proposed method deal with an MDA approach and based on meta-models for representing and aligning our models. Furthermore, all methods use a case study to evaluate their proposal.

6. CONCLUSION

In conclusion, this paper introduced a novel method for generating user stories' XMI model using NLP techniques and MDA approach in the context of agile software development. The method aimed to streamline the process of validating, extracting, and generation of user stories format in human language in a meaningful, structured, and standardized one. The approach offers substantial advantages in terms of time and effort reduction, and enhanced comprehension and communication. By automating and standardizing the process of managing user stories, the proposed approach has the potential to significantly advance the field of agile requirements management and contribute to the successful delivery of high-quality software products. Our challenge in future contributions is to generate different UML diagrams automatically from the generated user stories XMI model, while respecting the MDA approach in the Agile context.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the IT, Networking, Security, and Management (IRSM) Laboratory at the Higher Institute of Management Administration and Computer Engineering, Rabat, Morocco, for their generous sponsorship and financial support. This research was made possible through their invaluable contribution.

REFERENCES

[1] A. Manifesto, "Agile manifesto," *Haettu*, vol. 14, p. 2012, 2001

[2] M. Hajjaj and H. Lechheb, "Co-construction of a new management approach in a public research funding agency through the contextualization of agile thinking," *Organizational Cultures: An International Journal*, vol. 21, no. 1, pp. 21–34, 2021, doi: 10.18848/2327-8013/CGP/v21i01/21-34.

[3] M. Gärtner, *ATDD by example: a practical guide to acceptance test-driven development*. Addison-Wesley, 2013.

[4] M. Soeken, R. Wille, and R. Drechsler, "Assisted behavior driven development using natural language processing," in *Objects, Models, Components, Patterns: 50th International Conference*, TOOLS 2012, Springer, Prague, Czech Republic, 29-31 May 2012, pp. 269–287.

[5] OMG-XMI, XML Metadata Interchange version 2.5.1. OMG, 2015. [Online]. Available: https://www.omg.org/spec/XMI/2.5.1/PDF. (Accessed: 18 Jan, 2024).

[6] K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020, doi: 10.1007/978-81-322-3972-7_19.

[7] OMG-MDA, MDA Guide version 2.0. OMG, 2014. [Online]. Available: https://www.omg.org/mda/specs.htm. (Accessed: 18 Jan, 2024).

[8] N. Kharmoum *et al.*, "Agile user stories' driven method: A novel users stories meta-model in the MDA approach," in *International Conference on Advanced Intelligent Systems for Sustainable Development*, Springer, 2022, pp. 145–154, doi: 10.1007/978-3-031-26384-2_13.

[9] J. Bézivin, "On the basic principles of model engineering (in French: *Sur les principes de base de l'ingénierie des modèles*)," *Obj. Logiciel Base données Réseaux*, vol. 10, no. 4, pp. 145–157, 2004.

[10] X. Blanc and O. Salvatori, *MDA in action: Model-driven software engineering* (in French: *MDA en action: Ingénierie logicielle guidée par les modèles*), EYROLLES, 2011.

[11] B. Marchal, *XML by Example*. Que Publishing, 2002.

[12] C. L. Paris, W. R. Swartout, and W. C. Mann, *Natural language generation in artificial intelligence and computational linguistics*, Springer Science & Business Media, 2013, vol. 119.

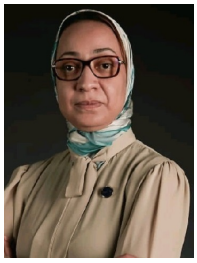
[13] M. Elallaoui, K. Nafil, and R. Touahni, "Automatic transformation of user stories into uml use case diagrams using nlp techniques," *Procedia Computer Science*, vol. 130, pp. 42–49, 2018, doi: 10.1016/j.procs.2018.04.010.
[14] S. C. Allala, J. P. Sotomayor, D. Santiago, T. M. King, and P. J. Clarke, "Towards transforming user requirements to test cases using mde and nlp," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Milwaukee, WI, USA, 2019, pp. 350-355, doi: 10.1109/COMPSAC.2019.10231.
[15] K. Baïna, M. El Hamlaoui, and H. Kabbaj, "Business process modelling augmented: Model driven transformation of user stories to processes," in *SITA'20: Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*, 2020, pp. 1–6, doi: 10.1145/3419604.3419793.
[16] OMG-BPMN, Business Process Model and Notation version 2.0.2. OMG, 2013. [Online]. Available: https://www.omg.org/spec/BPMN/2.0.2/PDF. (Accessed: 18 Jan, 2024).
[17] T. Georges, "Feature and variability extraction from agile specifications and their related source code for software product line migration," *SPLC '22: Proceedings of the 26th ACM International Systems and Software Product Line Conference*, 2022, Volume B, pp. 29–33, doi: 10.1145/3503229.3547065.
[18] A. Gupta, G. Poels, and P. Bera, "Generating multiple conceptual models from behavior-driven development scenarios," *Data & Knowledge Engineering*, vol. 145, p. 102141, 2023, doi: 10.1016/j.datak.2023.102141.
[19] S. Nasiri, A. Adadi, and M. Lahmer, "Automatic generation of business process models from user stories," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 809-822, 2023, doi: 10.11591/ijece.v13i1.pp809-822.
[20] N. Kharmoum, K. El Bouchti, N. Laaz, W. Rhalem, and Y. Rhazali, "Transformations' study between requirements models and business process models in mda approach," *Procedia Computer Science*, vol. 170, pp. 819–824, 2020, doi: 10.1016/j.procs.2020.03.150.
[21] K. El Bouchti, S. Ziti, F. Omary, and N. Kharmoum, "A new database encryption model based on encryption classes," *Journal of Computer Science*, vol. 15, no. 6, pp. 844–854, 2019, doi: 10.3844/jcssp.2019.844.854.
[22] L. Ajallouda, F. Z. Fagroud, A. Zellou, El h. Benlahmar, "Automatic keyphrases extraction: an overview of deep learning approaches," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 303–313, 2023, doi: 10.11591/eei.v12i1.4130.
[23] Y. Vasiliev, *Natural language processing with Python and spaCy: A practical introduction*, No Starch Press, 2020.
[24] T. Richards, *Getting Started with Streamlit for Data Science: Create and Deploy StreamlitWeb Applications from Scratch in Python*, Packt Publishing Limited, 2021.
[25] A. Sajji, Y. Rhazali, and Y. Hadi, "A methodology for transforming bpmn to ifml into mda," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 5, pp. 2773–2782, 2022, doi: 10.11591/eei.v11i5.3973.

BIOGRAPHIES OF AUTHORS

Nassim Kharmoum Ph.D. and University Professor. He received his diploma of doctor degree in computer science from the Mohammed V University in Rabat. Currently, he is Professor at the National Center for Scientific and Technical Research in Morocco. He proposed different validated methods in scientific articles published in international journals, books, and conferences. His research interest focuses on software engineering, artificial intelligence, data science, analysis and conceptual modeling, software requirements, multi-agent systems modeling, and modernization of legacy systems. He can be contacted at email: nkharmoum@gmail.com.

Sara Retal received her Ph.D. in computer science at the Faculty of Sciences of Rabat and is now an assistant professor of computer science and researcher at the University Hassan II Mohammedia, ENSET. She participated in the ICT Maghreb Technology Platform project co-funded by the European Community and in other research projects at Aalto University. Her research interests include applied research and applications of artificial intelligence and uses of AI systems for solving tasks in software engineering, digital health, networks, and so on. She can be contacted at email: retal@enset-media.ac.ma.

Mouna Hajjaj received her Ph.D. degree in Management from Ibn Tofail University. With extensive experience in the higher education sector, she held several senior positions before assuming her current role as Assistant Professor of management at the Faculty of Economics and Management at University Hassan I. Her research interests span a wide range of topics including agile project management, organization sciences, human resources, and business administration. She can be contacted at email: mouna.hajjaj@uhp.ac.ma.



Souad Najoua Lagmiri received her diploma for doctor degree from the Mohammadia School of Engineers Rabat. Currently Director of the Higher Institute of Management, Administration and Computer Engineering Rabat Morocco. She has published several scientific articles in international journals and conferences. His field of research focuses on encryption algorithms, cryptography, and information security based on complex mathematical equations. She can be contacted at email: najoua.lagmiri@gmail.com.

Yassine Rhazali is a Moroccan professor, he receive his Ph.D. in computer science at Ibn Tofail University, Kenitra, Morocco, more specifically in software engineering, his specialty is the model driven engineering, he proposed several approaches in model engineering, and his approaches are validated in various scientific papers published at international scientific conferences and international scientific journals. He is now a professor researcher, reviewer, editorial board member, and scientific committee member in several international scientific journals and at various international scientific conferences. He has taught computer science in varied universities and institutes. He can be contacted at email: dr.yassine.rhazali@gmail.com.