

Automated Software Engineering and ChatGPT*

Nabil ARMAN

Department of Computer Science and Information Technology
Palestine Polytechnic University
Hebron, Palestine

Correspondence should be addressed to: Nabil ARMAN, narman@ppu.edu.ps

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

Abstract

Automated Software Engineering (ASE) has attracted a great deal of research efforts in recent years. ASE has been used in database development, some UML models/diagrams generation from textual software requirements to name just a few examples. Recently, AI tools like ChatGPT have also been used in performing several tasks that were among the main tasks and the focus of ASE. This research paper aims to help software engineers by highlighting some examples and activities that used to be performed using specialized tools and how they can be performed using ChatGPT. This represents a valuable tool and a challenge for researchers interested in ASE. In addition, the use of AI tools, although not 100% accurate, reduces the cost and time required in performing different activities and tasks in software development. In addition, ChatGPT performs some functions of Computer-Aided Software Engineering (CASE) tools.

Keywords: Automated Software Engineering, AI tools, ChatGPT, CASE tools.

Introduction

Automated Software Engineering (ASE) has attracted a great deal of research efforts in recent years. ASE and CASE tools have been used in database development, like reverse engineering of a relational database to generate the conceptual model like an Entity-Relationship model (Arman, 2006a). ASE and CASE tools have also been used in normalizing a relational database up to a certain normal form Arman, 2006b).

In addition, ASE has also been used in UML models/diagrams generation from textual software requirements. Many researchers have used ASE to generate UML use case diagrams, UML activity diagrams, UML activity diagrams, UML sequence diagrams to name just a few examples (Arman and Daghameen, 2010; Jabbarin and Arman, 2014; Arman and Jabbarin, 2015; Arman, 2015; Alami, Arman and Khamayseh, 2017; Alami, Arman and Khamayseh, 2020). Other researchers have used ASE to classify software requirements into functional and non-functional requirements. Some researchers used ASE to classify functional requirements into different types of functional requirements Khamayseh, Arman and Khamayseh, 2022; Shehadeh, Arman and Khamayseh, 2021). Some researchers also used ASE to classify non-functional requirements into different types of non-functional requirements to name just a few examples (Awad, Arman and Khamayseh, 2023; Shehadeh, Arman and Khamayseh, 2024).

Recently, with the latest advancement of Artificial Intelligence (AI), many AI tools like ChatGPT have also been used in performing several tasks that were among the main tasks and the focus of ASE and CASE tools.

This research paper aims to help software engineers by highlighting some examples that used to be performed using specialized ASE and CASE tools and how they can be performed using ChatGPT. This represents a valuable

technique and a challenge for researchers interested in ASE and CASE tools development. In addition, the use of AI tools, although not a 100% accurate, reduces the cost and time required in performing different tasks in software development. It is worth mentioning that many ASE tools are not 100% automated, but rather can be considered semi-automated in the sense that human intervention is needed in some cases. In addition, ChatGPT performs Expert System's functionality providing explanation and justification for its suggested designs/solutions. Also, ChatGPT can deal with incomplete or missing information in certain contexts, which represents a great advantage over software engineers and knowledge engineers in their endeavours to design the systems of interest.

ASE research has been documented in many international conferences and refereed journals. It is highly recommended that researchers interested in these conferences and refereed journals test AI tools, like ChatGPT, to find out its capabilities in automating different tasks in the software development process. In fact, software engineering and software development is being shaped and affected by AI and tools like ChatGPT (Valchanov, 2024). Many activities usually performed by software engineers can be performed by ChatGPT.

When ChatGPT, itself, was asked "Can ChatGPT help in automated software engineering?", its answer includes (ChatGPT, 2024):

1. Code Generation
2. Code Optimization
3. Unit Testing
4. Code Debugging
5. Documentation
6. Requirements to Code
7. DevOps Automation
8. Database Management
9. Front-end Development
10. Machine Learning & Data Science
11. Version Control & Collaboration
12. Natural Language Processing in Software
13. Custom Scripts & Automation
14. Continuous Learning and Pair Programming

And it concluded its response with the question/answer:

Do you have a specific software engineering task in mind where automation is required? I can provide tailored assistance!

This means it has a lot more to offer.

Examples of Using ChatGPT in Software Engineering

In this section, a set of examples of using ChatGPT is presented to explain the main purpose of this research article:

i) Normalization of relational database schemas is a well-defined formal process. An enquiry has been asked to ChatGPT using the text:

Normalize $R(A,B,C,D,E)$ where the PK is AB and $B \rightarrow C$ to 2NF

ChatGPT response/answer was:

To normalize the relation $R(A,B,C,D,E)$ where the primary key is AB and $B \rightarrow C$ into **Second Normal Form (2NF)**, follow these steps:

Steps for Normalization to 2NF:

1. **Understand the dependencies:**
AB is the **primary key**, meaning A and B together uniquely identify all attributes in R.
 $B \rightarrow C$ indicates that B (a non-key attribute) can determine C.
2. **Check for partial dependencies:**

A partial dependency occurs when a non-key attribute is dependent on part of the composite primary key (i.e., only A or B, not AB).

$B \rightarrow C$ is a **partial dependency**, because C is dependent on B, which is only part of the composite primary key AB.

3. Decompose the relation to remove partial dependencies:

Create two relations:

One that contains B and C (to isolate the partial dependency $B \rightarrow C$).

Another that contains the remaining attributes of R while retaining the original primary key.

Decomposed Relations

1. $R_1(B,C)$:

Includes the attributes involved in the partial dependency $B \rightarrow C$.

The primary key for R_1 is B, as B uniquely determines C.

2. $R_2(A,B,D,E)$

Includes the remaining attributes A,B,D,E, where AB remains the primary key.

D and E are now only dependent on the whole key ABABAB.

Final Schema in 2NF

1. $R_1(B,C)$, where B is the primary key.

2. $R_2(A,B,D,E)$, where AB is the primary key.

This decomposition eliminates the partial dependency, ensuring that all non-key attributes are fully functionally dependent on the entire primary key.

Another enquiry was asked to find the candidate keys.

What are the candidate keys of $R(A,B,C,D,E)$ where $A \rightarrow C$, $C \rightarrow D$, $C \rightarrow E$ and $B \rightarrow C$?

To determine the **candidate keys** of $R(A,B,C,D,E)$, we analyze the functional dependencies and the closure of attribute combinations.

Given Functional Dependencies

$A \rightarrow C$

$C \rightarrow D$

$C \rightarrow E$

$B \rightarrow C$

Step 1: Determine Attributes Fully Functionally Dependent on Others

Using the transitive dependencies:

From $A \rightarrow C$ and $C \rightarrow D$, $C \rightarrow E$, we get $A \rightarrow CDE$.

From $B \rightarrow C$ and $C \rightarrow D$, $C \rightarrow E$, we get $B \rightarrow CDE$.

Thus, both A and B can determine CDE.

Step 2: Attributes Needed for a Candidate Key

To form a candidate key, we need to identify the smallest set of attributes that can determine all attributes (A,B,C,D,E) in R.

The attributes are A,B,C,D,E.

From the given dependencies:

A determines CDE, but not B.

B determines CDE, but not A.

Thus, A and B are **mutually dependent** and must be combined to form a superkey.

Step 3: Verify Candidate Keys

AB:

AB determines all attributes:

$A \rightarrow C$, and $C \rightarrow DE$, so A gives CDE.

$B \rightarrow C$, and $C \rightarrow DE$, so B gives CDE.

AB gives ABCDE.

Therefore, AB is a **candidate key**.

Any other combination (e.g., A,B,AC,BCA, B, AC, BCA,B,AC,BC) does not determine all attributes because: A alone or AC cannot determine BBB.

B alone or BC cannot determine A.

Final Answer

The only **candidate key** for $R(A,B,C,D,E)$ is: {AB}

End of ChatGPT response.

In fact, after reviewing the ChatGPT response, one can conclude that it produces a perfect answer. In addition, it has acted as an Expert System in this domain by explaining its answers.

ii) Reverse engineering of a relational database to generate an ER diagram or UML class diagram

An enquiry was asked to ChatGPT:

Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course, reverse engineer the database to generate an entity-relationship diagram:

STUDENT(SSN, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL(SSN, Course#, Quarter, Grade)

BOOK_ADOPTION(Course#, Quarter, Book_ISBN)

TEXT(Book_ISBN, Book_Title, Publisher, Author)

The response of the ChatGPT was:

To reverse-engineer the database schema into an Entity-Relationship Diagram (ERD), we analyze the given relations and identify the entities, attributes, and relationships:

Entities and Their Attributes:

1. **STUDENT**

Primary Key: SSN

Attributes: Name, Major, Bdate

2. **COURSE**

Primary Key: Course#

Attributes: Cname, Dept

3. **TEXT**

Primary Key: Book_ISBN

Attributes: Book_Title, Publisher, Author

Relationships and Their Attributes:

1. **ENROLL**

This is a relationship between STUDENT and COURSE.

Attributes: Quarter, Grade

Composite Key: (SSN, Course#)

2. BOOK_ADOPTION

This is a relationship between COURSE and TEXT.

Attributes: Quarter

Composite Key: (Course#, Quarter)

ERD Details:

- **Entities:**

Represent STUDENT, COURSE, and TEXT as rectangles.

- **Attributes:**

Represent each attribute within the entities as ovals connected to their respective entity.

Mark primary keys (e.g., SSN, Course#, Book_ ISBN) as underlined attributes.

- **Relationships:**

ENROLL is a many-to-many relationship between STUDENT and COURSE.

BOOK_ADOPTION is a many-to-many relationship between COURSE and TEXT.

- **Cardinalities:**

ENROLL:

A STUDENT can enroll in multiple COURSES, and a COURSE can have multiple STUDENTS.

BOOK_ADOPTION:

A COURSE can adopt multiple TEXTS, and a TEXT can be adopted by multiple COURSES.

Would you like a visual representation of the ERD?

You said:

Yes

ChatGPT said:

Here is the visual representation of the Entity-Relationship Diagram (ERD) for the university database. (omitted to save space).

Again, after reviewing the ChatGPT response, one can conclude that it produces a perfect answer. In addition, it has acted as an Expert System in this domain by dealing with incomplete and missing information regarding the foreign keys. And it suggested the appropriate relationship types along with the integrity constraints, including multiplicity constraints. Although, ChatGPT didn't generate the diagram, it produces an exact description of the required ER diagram. It also suggested to help in guiding the user through creating the diagram manually using tools like **draw.io**, **Lucidchart**, or **Microsoft Word**.

UML diagrams generation

ChatGPT was given user/software requirements and was asked to draw/design different UML diagrams/models like Use case diagrams, Sequence diagrams, Activity diagrams, ...etc. and produces the required diagrams/models. These diagrams were accurate and needed minor human intervention to correct minor errors. Examples were also omitted to save space.

In addition, ChatGPT can be used to classify user/software requirements into functional or non-functional requirements. Also ChatGPT can be used to further classify functional requirements into its main categories and classify non-functional requirements into its main categories. When ChatGPT was asked: Is the requirement "The program should have interface in different languages" a functional or a non-functional requirement?

ChatGPT said:

The requirement "The program should have an interface in different languages" is a functional requirement.

It also provided a justification for its answer by writing:

Reason:

It describes a specific feature or behavior that the system must provide — namely, the ability to support multiple languages in the interface. Functional requirements define what the system should do, and in this case, providing multilingual support is a key function of the program.

Different Natural languages are supported in ChatGPT in the sense that it understands enquires and representations in different languages, including Arabic.

Summary and Discussion

Many researchers have worked on ASE and SE engineering. They used different approaches and proposed automated and semi-automated approaches to obtain different models/diagrams/components. AI tools like ChatGPT have proven its capabilities in obtaining these models/diagrams/components. In many cases, it even outperforms ASE tools by providing explanation/justification for deciding to use certain options and not others. AI tools like ChatGPT can deal with incomplete or missing information providing accurate answers or solutions to many questions.

Therefore, using AI tools like ChatGPT in software development will change the functions and jobs of programmers and software engineers. The jobs of software engineers will shift towards sophisticated and creative activities rather than focusing routine tasks. The functions of programmers will shift towards program optimization rather than writing simple code. This definitely reduces the cost and time needed to develop software and allows software engineers to focus more on optimizing and testing the developed software rather than spending/wasting time in routine activities.

Conclusion

Software Engineering and ASE have attracted a great deal of research efforts in recent years. ASE has been used in database development, some UML models/diagrams generation from textual software requirements, and many typical functions of software engineers, to name just few examples. Recently, AI tools like ChatGPT have also been used in performing several tasks that were among the main tasks and the focus of ASE. This paper aims to help and advise software engineers by highlighting some examples that used to be performed using specialized tools and techniques and how they can be performed using ChatGPT. This represents a valuable tool and a challenge for researchers interested in ASE. In addition, the use of AI tools, although not a 100% accurate, reduces the cost and time required in performing different tasks in software development.

References

- Arman, N. (2006), 'ERRDS: A Case Tool to Generate an ER Data Model from a Relational Database Schema,' Proceedings of the 7th International Arab Conference on Information Technology (ACIT'2006), December 19-21, 2006, Yarmouk University, Irbid, Jordan.
- Arman, N. (2006), 'Normalizer: A Case Tool to Normalize Relational Database Schemas,' Information Technology Journal 5(2), 329-331, ISSN: 1812-5638.
- Arman, N. and Daghameen, K. (2010), 'Requirements Based Static Class Diagram Constructor (SCDC) Case Tool,' Journal of Theoretical and Applied Information Technology 15(2), 108-114, ISSN 1992-8645.
- Jabbarin, S. and Arman, N. (2014), 'Constructing Use Case Models from Arabic User Requirements in a Semi-Automated Approach,' Proceedings of the International Conference on Requirements Engineering (ICRE'2014), January 17-19, 2014, Hammamet, Tunisia.
- Arman, N. and Jabbarin, S. (2015), 'Generating Use Case Models from Arabic User Requirements in a Semiautomated Approach Using a Natural Language Processing Tool,' Journal of Intelligent Systems (JISYS), 24(2), 277-286.
- Arman, N. (2015), 'Using MADA+TOKAN to Generate Use Case Models from Arabic User Requirements in a Semi-Automated Approach,' Proceedings of The 7th International Conference on Information Technology (ICIT 2015), May 12-15, 2015, Amman, Jordan.
- Alami, N., Arman, N. and Khamayseh, F. (2017), 'A Semi-automated Approach for Generating Sequence Diagrams from Arabic User Requirements Using a Natural Language Processing Tool,' Proceedings of The 8th International Conference on Information Technology (ICIT 2017), May 13-14, 2017, Amman, Jordan.

- Alami, N, Arman, N., and Khamayseh, F. (2020), ‘Generating Sequence Diagrams from Arabic User Requirements using MADA+TOKAN Tool,’ *International Arab Journal of Information Technology*, 17(1), 65-72.
- Khamayseh, B., Arman, N. and Khamayseh, F. (2022), ‘Security Requirements Classification into Functional and Non-functional,’ *Proceedings of The 40th International Business Information Management Association (IBIMA) Computer Science Conference*, 23-24 November 2022.
- Shehadeh, K., Arman, N. and Khamayseh, F. (2021), ‘Semi-Automated Classification of Arabic User Requirements Into Functional and Non-Functional Requirements Using NLP Tools,’ *Proceedings of The 10th International Conference on Information Technology (ICIT 2021)*, July 14-15, 2021, Jordan.
- Awad, E., Arman, N. and Khamayseh, F. (2023), ‘Semi-automated Classification of Non-functional Arabic User Requirements Using NLP Tools,’ *Artificial intelligence and Machine Learning: 41st IBIMA International Conference, IBIMA-AI 2023*, June 26–27, 2023, Granada, Spain.
- Shehadeh, K., Arman, N. and Khamayseh, F. (2024), ‘Classification of Arabic User Requirements: A Semi-Automated Approach using NLP Tools,’ *International Journal of Advances in Soft Computing and Its Applications (IJASCA)*, 16(3) 1-14.
- Valchanov, I. (2024) ‘The Future of Software Engineering + ChatGPT Use Cases for Code [2024],’ URL: <https://team-gpt.com/blog/future-of-software-engineering> (Accessed: November 15, 2024).
- ChatGPT: chatgpt.com (Accessed: November 1, 2024).