Volume 13 Number 1, March 2024

International Journal of Advanced Smart Convergence

The Institute of Internet, Broadcasting and Communication

International Journal of Advanced Smart Convergence Volume 13 Number 1, March 2024 The Institute of Internet, Broadcasting and Communication

Volume 13 Number 1, March 2024

KCI Indexed Journal
ISSN 2288-2847(Print)
ISSN 2288-2855(Online)

International Journal of Advanced Smart Convergence

IIBC The Institute of Internet, Broadcasting and Communication
http://www.iibc.kr

International Journal of Advanced Smart Convergence Vol.13 No.1 85-91 (2024)
http://dx.doi.org/10.7236/IJASC.2024.13.1.85

IJASC 24-1-9

Toon Image Generation of Main Characters in a Comic from Object Diagram via Natural Language Based Requirement Specifications

[1]Janghwan Kim, [2]Jihoon Kong, [3]Hee-Do Heo, [4]Sam-Hyun Chun, [5]R. Young Chul Kim

[1,2] PhD. Candidate, Dept. of Software and Communication Engineering, Hongik University, Korea
[2]Director, Toonsquare, Korea
[3] CEO, NHN Crossent, Korea
[4] Professor, Dept. of Law, Soongsil University, Korea
[5] Professor, Dept. of Software and Communication Engineering, Hongik University, Korea

[1]lentoconstante@hongik.ac.kr, [2]go400@naver.com, [3]hd-dream@hanmail.net,
[4]shchun@ssu.ac.kr [5]bob@hongik.ac.kr

Abstract

Currently, generative artificial intelligence is a hot topic around the world. Generative artificial intelligence creates various images, art, video clips, advertisements, etc. The problem is that it is very difficult to verify the internal work of artificial intelligence. As a requirements engineer, I attempt to create a toon image by applying linguistic mechanisms to the current issue. This is combined with the UML object model through the semantic role analysis technique of linguists Chomsky and Fillmore. Then, the derived properties are linked to the toon creation template. This is to ensure productivity based on reusability rather than creativity in toon engineering. In the future, we plan to increase toon image productivity by incorporating software development processes and reusability.

Keywords: *Toon Engineering, Software Engineering, Unified Modeling Language, Natural Language Analysis*

1. Introduction

Recently, interest in generative artificial intelligence based on deep learning is increasing. These artificial intelligence tools are developing by generating sentences, writing, calculations, criticism, and images through natural language and voice queries. In particular, the growth of text-based image creation tools (e.g. Dall-E3, Midjourney, ChatGPT) is notable. These tools identify the meaning of natural language queries and generate or synthesize various images [1]. However, these tools produce inconsistent results even with the same text input [2].

In order to improve this problem, this paper proposes a method of deriving results by applying linguistic analysis and software engineering design techniques through natural language queries, compared to methods

Manuscript Received: January. 23, 2024 / Revised: January. 28, 2024 / Accepted: February. 5, 2024
Corresponding Author: bob@hongik.ac.kr
Tel: +82-44-860-2477, Fax: +82-44-865-0460
Professor, Department of Software and Communication Engineering, Hongik University, Republic of Korea

of generating images using artificial intelligence by inputting existing natural language sentences. . As an applied example, class diagrams and object diagrams among UML diagrams are applied to show the step-by-step process required to create images from natural language sentences.

Chapter 2 related research mentions prompts, and sentence analysis methods applied to generative artificial intelligence tools in natural language. Chapter 3 mentions the mechanism for generating class diagrams through natural language sentence analysis and generating images based on them. Chapter 4 mentions applied works based on our mechanism, and Chapter 5 discusses conclusions and future research.

2. Related Works

2.1 Class Diagram from Requirement Specification

Class diagrams are a core element of UML that represents the static structure of a system. This diagram represents the various classes, their properties, operations, and relationships between classes. It plays an important role in object-oriented analysis and design, providing a detailed description of the classes of systems, their interrelationships, and how they cooperate. Class diagrams are widely used in software engineering to visualize and document class structures [3]. Class diagrams also enable developers to model inheritance, associations, and dependencies between classes, aiding in the creation of robust and maintainable software. By offering a clear depiction of the relationships within a system, class diagrams facilitate effective communication among team members, making them a vital tool for software engineers throughout the development process.

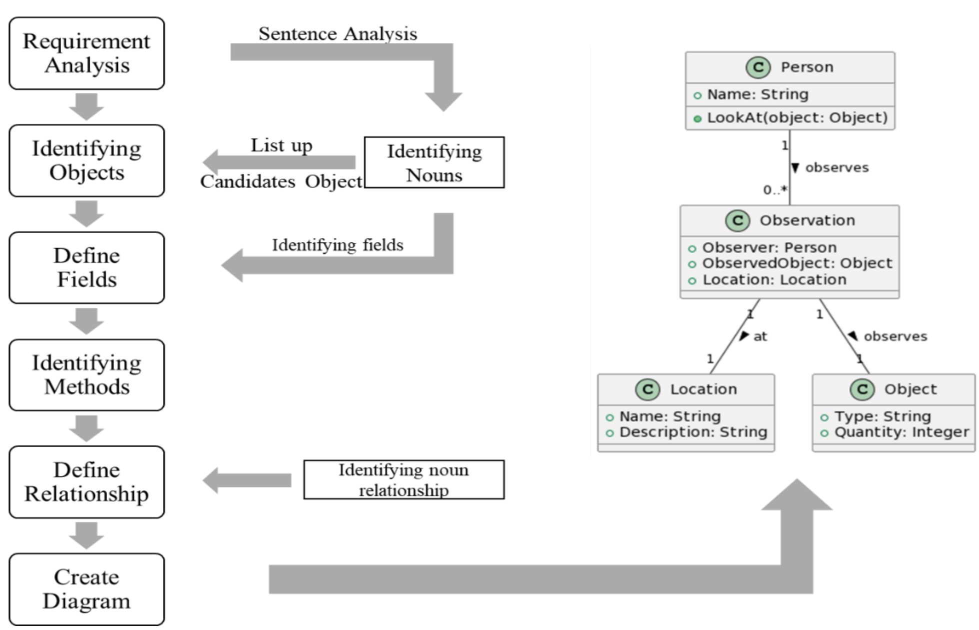

Figure 1. Requirement Analysis Mechanism to create Class Diagram

Figure 1 shows how to extract a class diagram from requirements specifications through requirements analysis. In the requirements analysis stage, sentences in the requirements specification are analyzed to identify nouns and used to define classes or fields. After all nouns are identified, entity candidates are selected from among them to finally define the entity. The identified noun is used as the class name, and the remaining nouns are used as class attributes. Methods are defined based on the behavior of an object and are often identified by a verb. After identifying entity names, properties, and methods, create a class diagram based on the derived information.

2.2 Linguistic Sentence Structure and Semantic Analysis Mechanism

The various linguistic analysis methods used to analyze sentences are largely divided into structural analysis methods and semantic analysis methods [4]. In particular, Fillmore's Case Grammar is a semantic analysis method of language that focuses on the relationship between the predicate, subject, and object in an English sentence [5-6]. Table 1 presents Fillmore's definition of Original Case Grammar in a table [6].

Table 1. Fillmore's Original Case Grammar Approach

Case	Description
Agent	The entity performing an action
Instrumental	The tool or means used to perform an action
Dative	The entity that receives or benefits from an action
Objective	The result or product of an action
Factitive	The location or direction of an action
Locative	The entity that is acted upon or affected by an action

Fillmore's Case Grammar, a seminal approach in linguistic theory, revolutionizes the understanding of sentence structure by emphasizing the dynamic roles of words. It categorizes elements based on their semantic roles in sentences, such as Agent, Instrument, and Experiencer, offering a more nuanced interpretation of language beyond traditional syntax. This framework particularly highlights the significance of context in determining the meaning, underscoring the interplay between syntax and semantics. By elucidating these relationships, Case Grammar provides crucial insights for computational linguistics, aiding in the development of more sophisticated natural language processing algorithms.

3. Requirements Analysis Process based on Linguistics Mechanisms

In this paper, we apply a linguistic approach to extract class diagrams from requirements analysis, integrating structural and semantic analysis. In this process, the sentence structure is first identified through structural analysis, and potential nouns are identified as classes. This structural analysis involves identifying the components of a sentence, such as nouns, verbs, and adjectives, and analyzing them according to the syntactic rules of the language. Semantic analysis is then applied to understand the contextual relationships and roles of the identified nouns. This method helps to comprehensively extract important information within the requirements, and based on this, create accurate and representative class diagrams. This approach not only helps accurately identify classes and their properties, but also improves the overall understanding of system requirements, enabling efficient analysis.

- Step 1: Analyze a structural analysis of requirements
  Start by structurally analyzing the requirements. In this step, the structure of the sentence is identified and potential nouns are identified as classes.

- Step 2: Identify nouns as class candidates
  Through structural analysis, nouns that can be expressed as classes within the system are identified.

- Step 3: Apply semantic analysis to identified nouns
  Apply semantic analysis to identified nouns to understand their role and contextual meaning.

- Step 4: Define properties and methods
  Define properties and methods for each candidate class. This step converts the abstract noun into a concrete class expression.

- Step 5: Establish inter-class relationships and hierarchy
  Determine relationships and hierarchies between classes. It includes inheritance structures and associations and is a key element of a class diagram.

- Step 6: Create a class diagram
  Create a class diagram based on all the collected information. This diagram visually represents classes, their properties, methods, and the relationships between classes and provides a clear overview of the system structure.

4. Best practice of Requirements Analysis Process based on Linguistics Mechanisms

Natural Language Requirement:
There is a house in the forest. Harry is looking at some mushrooms in front of the house.

Figure 2. Natural Language Requirement Prompt Example

Figure 2 shows the natural language requirements application case input. The parts of speech of natural language sentences extracted through Chomsky's syntactic-structure are re-analyzed through Fillmore's case

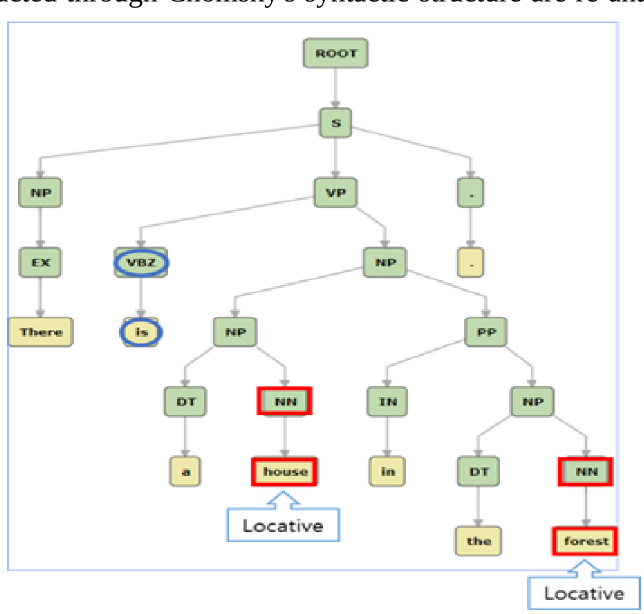

Figure 3. Syntactic Structure Analysis of Prompt

grammar.

Figure 3 shows analysis of the input sentence using the Stanford parser. Stanford Parser applies the phrase structure analysis method to the input sentence and displays the part-of-speech information of each element of the sentence [7]. Among the extracted parts of speech, nouns are classified and the meaning of verbs and nouns is identified through Fillmore's Case Grammar.

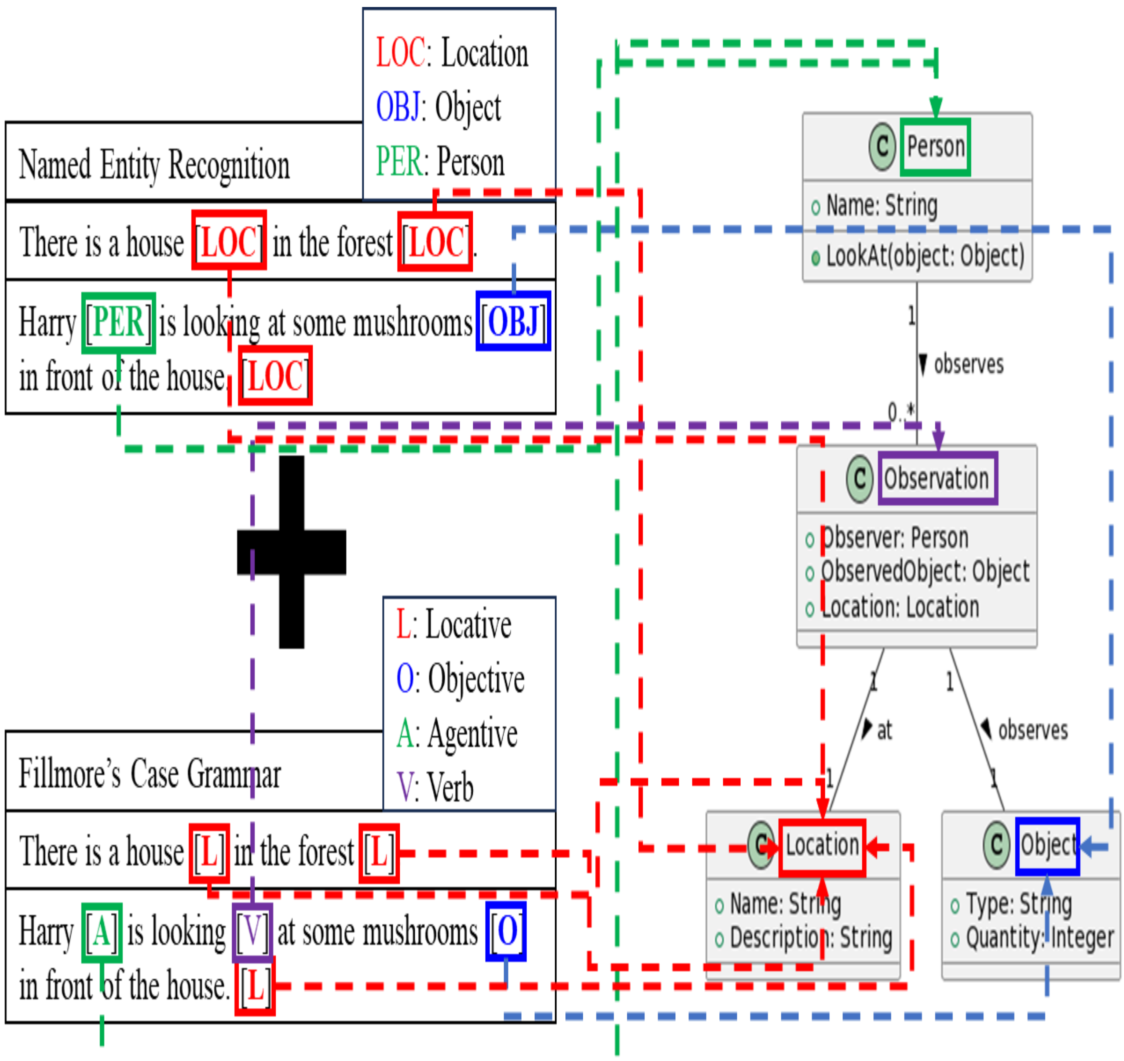

Figure 4. Creating Class Diagram from Linguistic Analysis

Figure 4 shows the derivation of a class diagram from the analysis results using Fillmore's case grammar method. Agentive cases are connected to the Person class, and Locative cases are all classified into the Location class. Since most of the input sentences describe toon images, they are connected to Location classes such as place, background, etc. Other objects are mapped to the Object class and their properties are set as fields with names and values. Once the class diagram is created, an object diagram with the properties of the class diagram is created to create a toon image.

Figure 5. Image Generation with Generative A.I Tools

Figure 5 shows various images that are generated by generative A.I tools. We compare images between tools and our software Engineering Approach with same input.

5. Conclusion

In this paper, we propose a methodology for producing cartoon images, which incorporates the use of linguistic analysis alongside the development process of software engineering. This approach enhances reproducibility by making a process of the conversion of natural language statements into visual outputs, employing both linguistic assessments and software engineering. A key aim of this methodology is to augment the efficiency and productivity of cartoon image generation, with a strong emphasis on the reusability aspect within the software development cycle. By integrating the software development procedures into the core architecture of our proposed method, we lay down a solid groundwork for future innovations in automated image creation. However, we acknowledge a significant constraint associated with this methodology: the generation of consistent pattern images necessitates a substantial collection of similar graphical assets. Despite this limitation, our approach offers a distinct advantage over generative AI tools, which produce images without revealing the internal operational details. Our method not only defines and executes intermediate stages but also makes the process transparent, offering insights into the underlying mechanisms of Toon image generation.

Acknowledgement

This study was conducted with the support of the Korea Creative Content Agency (Task Name: Artificial Intelligence-based User Interactive Storytelling 3D Scene Authoring Technology Development, Task Number: RS-2023-00227917, Contribution Rate: 70%) and the Basic Research Project of the Korea Research Foundation (Task Name: Non-defective Research through automatic refactoring based on the NLP BERT Model, Task Number: No.2021R1I1A3050407, Contribution Rate: 15%) with the support of the Korea Creative Content Agency (Task Name: No.2021R1A3050407, Contribution Rate: 15%).

References

[1] H. Park, A Case Study On Application Of Text To Image Generator AI DALL. E. The Treatise on The Plastic Media, 26(1), 104.

[2] E. Creely, “The possibilities, limitations, and dangers of generative AI in language learning and literacy practices.” *International Graduate Research Symposium 2023. Proceedings.* University of languages and International Studies (ULIS). Hanoi, Vietnam.

[3] M. Ibrahim and R. Ahmad, "Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques," 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 2010, pp. 200-204, doi: 10.1109/ICCRD.2010.71.

[4] Chomsky, Noam. Syntactic structures. Mouton de Gruyter, 2002.

[5] M. Ahmed, I. Sameer, and M. Heeti. "The Semantic Roles Which Cause Difficulty To The Fourth Year Students Of English At The University Of Anbar." Opción: Revista de Ciencias Humanas y Sociales 89 2019.

[6] C. J. Fillmore, (1967). “The case for case.”

[7] S. Schuster and C. D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In LREC 2016.