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

Volume 13 Number 1
March 2024

Table of Contents

IJASC Contents

Volume 13 Number 1
March 2024

International Journal of Advanced Smart Convergence Vol.13 No.1 92-98 (2024)
http://dx.doi.org/10.7236/IJASC.2024.13.1.92

IJASC 24-1-10

Emotion Analysis of Characters in a Comic from State Diagram via Natural Language-based Requirement Specifications

Ye Jin Jin[1], Ji Hoon Kong[2], Hyun Seung Son[3], R. Young Chul Kim[4]

[1] *M.S., Dept. of Software and Communication Engineering, Hongik University, Korea*
[2] *Ph.D. Candidate, Dept. of Software and Communication Engineering, Hongik University (Toonsquare), Korea*
[3] *Assistant Professor, Dept. of Computer Engineering, Mokpo National University, Korea*
[4] *Professor, Dept. of Software and Communication Engineering, Hongik University, Korea*

[1]yejin_jin@g.hongik.ac.kr, [2]john.tooning@toonsquare.co, [3]hson@mnu.ac.kr, [4]bob@hongik.ac.kr

Abstract

The current software industry has an emerging issue with natural language-based requirement specifications. However, the accuracy of such requirement analysis remains a concern. It is noted that most errors still occur at the requirement specification stage. Defining and analyzing requirements based on natural language has become necessary. To address this issue, the linguistic theories of Chomsky and Fillmore are applied to the analysis of natural language-based requirements. This involves identifying the semantics of morphemes and nouns. Consequently, a mechanism was proposed for extracting object state designs and automatically generating code templates. Building on this mechanism, I suggest generating natural language-based comic images. Utilizing state diagrams, I apply changes to the states of comic characters (protagonists) and extract variations in their expressions. This introduces a novel approach to comic image generation. I anticipate highly productive comic creation by applying software processes to Cartoon ART.

Keywords: *Cartoon, Emotion, UML, State Diagram, Requirements, Natural Language*

1. Introduction

The software development process typically involves requirement analysis, design, implementation, testing, and maintenance phases. Notably, 66% of software errors occur during the system specification phase[1]. While natural language provides a convenient medium for describing atypical requirements, discrepancies in interpretation between developers and clients can significantly increase the risk of software failure. Errors at the specification stage often led to systemic development failures, underscoring the need for precise requirement analysis and specification to align with the client's intentions.

This study uses the Unified Modeling Language (UML) model to minimize errors in specifying and

Manuscript Received: January. 23, 2024 / Revised: January. 29, 2024 / Accepted: February. 6, 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, Korea

designing requirements. UML offers greater visibility than natural language, facilitating more transparent communication. I integrate two linguistic approaches to derive UML from natural language descriptions automatically. Chomsky's syntactic structure analysis helps identify and dissect sentence structures at the morpheme level. Fillmore's semantic frameworks analyze the relationships between the sentence elements. This study maps linguistically extracted information and essential components for state diagram generation within the UML framework. This approach enables the extraction of state diagrams by analyzing the state changes of main characters in cartoons, as described in natural language. Subsequently, a JSON code template is constructed using Fabric.js to generate cartoon images. In existing research, Java code was generated by creating a state diagram through natural language analysis[2]. In this study, I apply existing research to cartoons to create cartoon images.

The paper is organized as follows: Chapter 2 discusses related approaches and linguistic theories. Chapter 3 outlines the proposed mechanism, and Chapter 4 presents a case study applying this mechanism. Finally, the paper concludes with discussions of the findings and directions for future research.

2. Related Works

2.1 Code Templates Generation from State Diagrams

This study presents a code template generation approach that combines software development process techniques and design thinking mechanisms [3]. Figure 1 shows part of the state diagram-related skeleton code generation process proposed in this study.

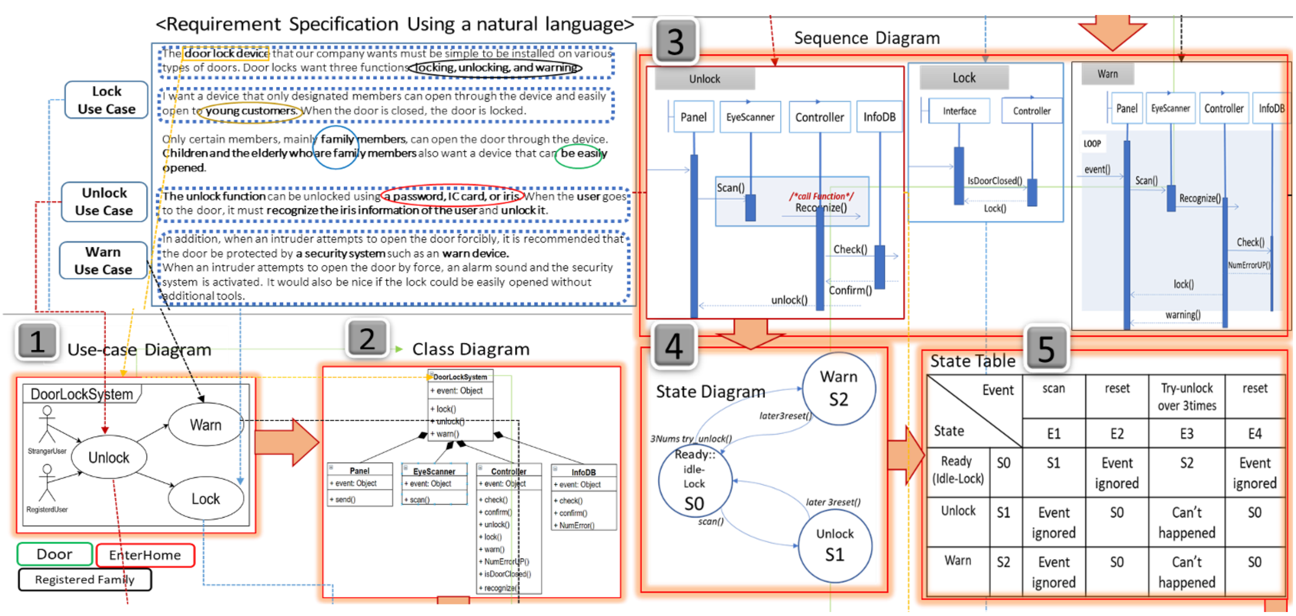

Figure 1. Software Development Process for Skeleton Code Generation

Requirements specifications are defined to extract use cases, which are used to derive state diagrams. A state diagram generates skeletal code that constructs a state table and represents the operations and events for each state within the class. This approach promotes systematic and structured software development.

In this study, I do not go through a mapping process between requirements and use cases but instead, extract state diagrams directly from language-based requirements analysis.

2.2 State Extraction Approach with Requirement Modeling

This research conducts natural language requirement modeling based on states and modes to understand system requirements [4]. They design and implement a MoSt (Modes and States) meta-model to facilitate the extraction of states and modes from natural language requirements. It is proposed that this approach will enhance the efficiency of requirement management. However, it is acknowledged that the existing model, as outlined in prior studies, mandates natural language requirements to adhere to a predefined template. This stipulation poses a limitation in accommodating the diversity of natural language requirements.

In response to this limitation, my study takes a novel approach by incorporating linguistic principles to analyze complexly structured requirements systematically. Unlike conventional templates, this linguistic analysis allows for a more flexible and adaptable approach, ensuring broader applicability to diverse forms of natural language requirements. By sidestepping the constraints of a rigid template, I aim to enhance the versatility and effectiveness of my state extraction approach, making it more adept at handling the intricacies inherent in a wide range of system requirements. This emphasis on linguistic analysis adds a layer of sophistication to my methodology, fostering a more inclusive and comprehensive understanding of natural language requirements in the context of state extraction and requirement modeling.

2.3 Chomsky's Syntactic Structure Analysis

Chomsky's contribution to syntactic structure analysis is rooted in his expertise as a linguist, enabling the systematic extraction of sentence components [5]. To conduct a comprehensive syntactic analysis, sentences are systematically divided into hierarchical components, progressing from the overall sentence to clauses, phrases, and individual parts of speech.

	Tag	Description
JJ*	JJ	Adjective
	JJR	Adjective, comparative
	JJS	Adjective, superlative
RB*	RB	Adverb
	RBR	Adverb, comparative
	RBS	Adverb, superlative
	RP	Particle
NN*	NN	Noun, singular or mass
	NNS	Noun, plural
	NNP	Proper noun, singular
	NNPS	Proper noun, plural
PRP*	PRP	Personal pronoun
	PRP$	Possessive pronoun
VB*	VB	Verb, base form
	VBD	Verb, past tense
	VBG	Verb, gerund or present participle
	VBN	Verb, past participle
	VBP	Verb, non-3rd person singular present
	VBZ	Verb, 3rd person singular present

Figure 2. Part Of Speech Tag

Figure 2 shows the frequently used parts of speech in this study. Notably, the tags assigned to both primary and derived parts of speech exhibit similar forms. To distinguish between the two categories in my research, I introduce an asterisk (*) as a suffix to denote primary parts of speech tags.

I employ the Berkeley Parser, which applies Chomsky's syntactic theory [6]. This parser facilitates the creation of tree structures, allowing for a detailed examination of the hierarchical relationships among sentence components. Through this approach, I can gain deeper insights into the structure of natural language.

2.4 Fillmore's Semantic Roles Analysis

Fillmore is a linguist who focuses on the relationship between nouns and verbs in sentences [7]. This theory analyzes syntax by studying combinations of semantic roles based on verbs.

Table 1. Part of Fillmore's Semantic Roles

Role	Definition
Agent	A person or entity causing a verb's action to be performed.
Experiencer	A person or thing affected by a verb's action, replacing the dative.
Object	An entity affected directly by a transitive verb's action
Instrument	The inanimate entity causally involved in a verb's action.

Table 1 is part of the semantic roles domain of Fillmore. The case is redefined in research that applied Fillmore's linguistic theory to software engineering, and the requirement sentence is analyzed to extract the use case [8]. Based on this, this study redefined and applied the appropriate role for state diagram extraction.

3. Extract State Diagrams from Requirements

3.1 Requirement Analysis Process

In this study, I reconstruct natural language sentences and extract state diagrams through linguistic analysis. To extract the state diagram, the semantic domain of Fillmore among the linguistics used in this study is redefined, as shown in Table 2.

Table 2. Fillmore's Semantic Roles Redefined

Role [Notation]	Definition
Agent [A]	An entity affected directly by a transitive verb's action
Object [O]	The entity that undergoes an action or change of state
Cause [C]	The entity that causes an action or event
Stimulus [S]	The entity that triggers an action or event
Instrument [I]	An entity used to perform an action or event.

The cause in Table 2 refers to the cause of a particular event, and Stimulus refers to a condition for an event to occur due to a trigger of a specific event.

3.2 Cartoon Creation Process

The process of generating cartoon images from the natural language proposed in this study is shown in

Figure 3.

When a natural language sentence is input, the double text is preprocessed into a single text. I analyze the syntax structure of preprocessed sentences through the Berkeley parser. Each analyzed component is given a redefined semantic domain. A state diagram is created based on linguistic elements. In this case, the state of the state diagram is mapped to the morpheme of the syntax structure analysis step. A state diagram is extracted based on the mapped elements. The extracted state is connected to the state of the cartoon object. The cartoon's protagonist expresses four emotions: joy, anger, sorrow, and pleasure. The protagonist's state generates an image as a facial expression according to emotion.

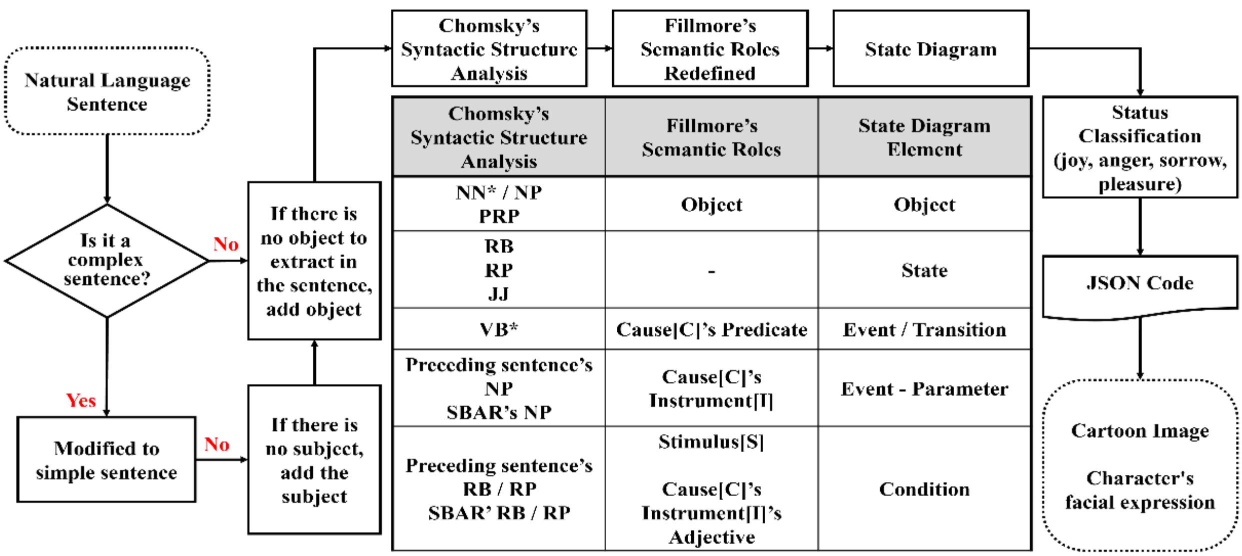

Figure 3. Process of Generating Cartoon Image from Natural Language

4. Case Study

If the mechanism proposed in this study is applied to the cartoon story, it is shown in Figure 4.

Figure 4 illustrates an example of the male protagonist, Edward, purchasing a beverage from a vending machine. In Figure 4, the state diagram and JSON code for image generation are generated by analyzing the five steps of the requirement sentences.

Step 1: Natural Language Analysis.

The natural language requirements are processed according to the existing procedure. Complex sentences are divided into simple sentences, and the missing subject is inserted to complete the sentence.

Step 2: Identifying the morpheme of a sentence through the Berkeley parser.

Preprocessed sentences undergo morpheme segmentation with the Berkeley Parser, grounded in Chomsky's theory. The parser analyzes sentences, phrases, clauses, and nouns, representing them in a tree structure. The Parts of Speech (POS) tags associated with each element facilitate the verification of their respective grammatical categories. In Figure 4, an analysis is conducted on the second sentence of Scene 1 and the fourth sentence of Scene 2, both containing expressions of emotion.

Step 3: Analysis of the role of Fillmore mapped to morphemes.

Figure 3 illustrates Chomsky's Syntactic Structure analysis mapping with Fillmore's Semantic Role. Based on this, Fillmore's role theory is applied to the elements of analyzed sentences. Roles are inferred and assigned based on the mapping table that aligns parts of speech with semantic roles. In the first sentence of Scene 1, the giving verb induces emotional change, making it the 'Cause's Predicate,' and the tool used, coke, assigns the role of 'Instrument.' A similar analysis is applied in Scene 2.

Step 4: Mapping core properties and roles in a state diagram.

Referring to the object diagram employed in previous studies for cartoon creation [9], the primary objects within cartoons are identified. Each element of a linguistically analyzed sentence is then mapped to a corresponding component of the state diagram. Elements analyzed through Fillmore's theory are utilized in the transitions of the state diagram, while emotions analyzed through Chomsky's theory are applied to the states. Based on this, a state diagram representing the main objects is generated.

Step 5: Create cartoon images by mapping state diagram properties with cartoon properties.

The states within the state diagram are categorized based on significant emotions such as joy, anger, sorrow, and pleasure. Emotion-indicative facial expression assets are retrieved from the JSON file for the objects and displayed on the canvas accordingly. The expression assets, reflecting the emotions of the main object, are loaded onto the canvas to create the cartoon images.

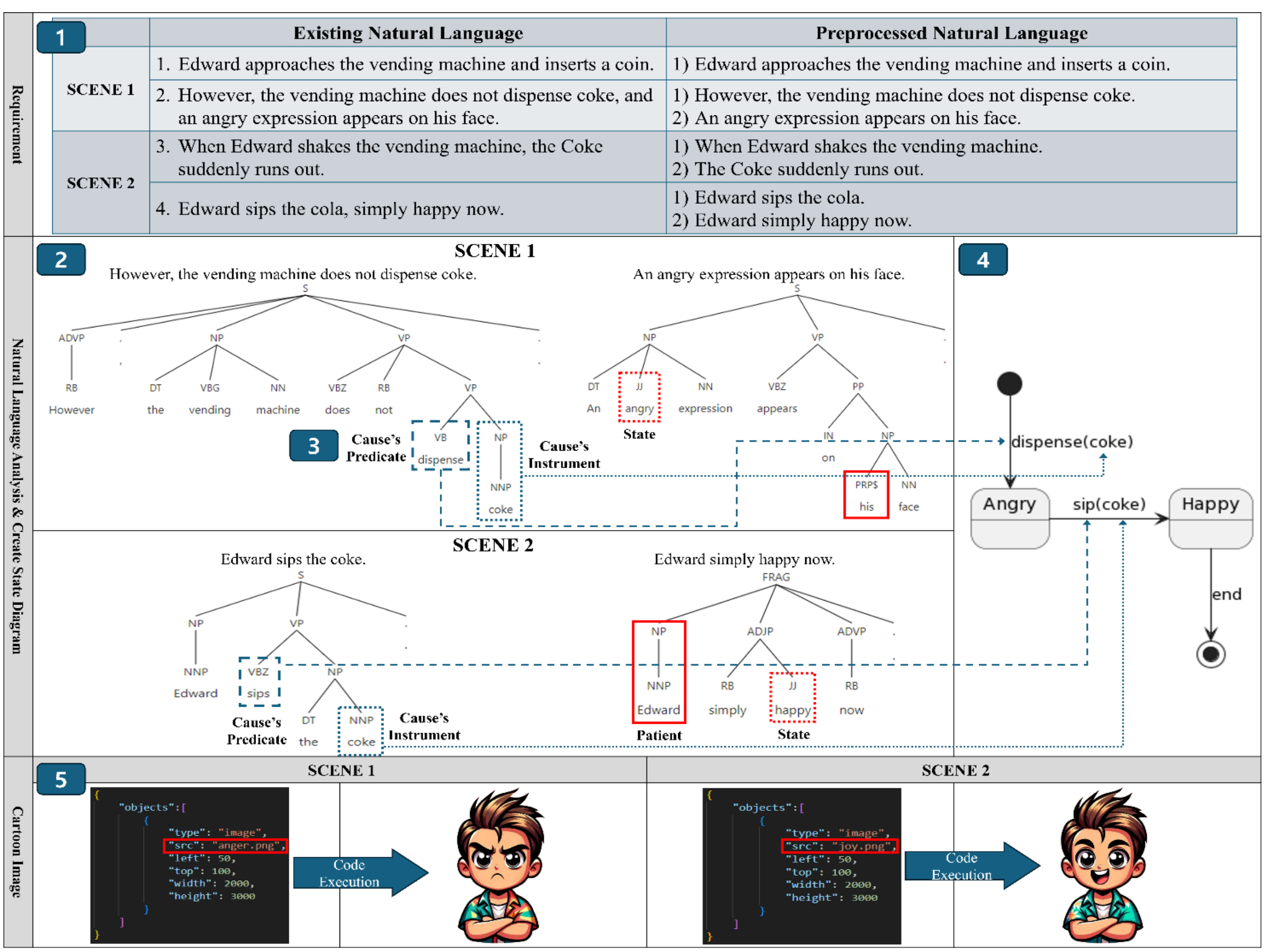

Figure 4. Example of Proposed Process Application

5. Conclusion

This study applies the linguistic theories of Fillmore and Chomsky to analyze natural language requirements. Consequently, it proposes a mechanism that converts natural language descriptions into state diagrams and cartoon images. Utilizing this proposed mechanism, it is possible to generate cartoon images that naturally reflect changes in the state of cartoon objects through systematic analysis. Moreover, the state change of the cartoon object can be clearly understood through the state diagram. However, as it represents only the state change of one central object, the detailed state of the object cannot be confirmed.

Future studies will address the state of all objects and allow for the expression of detailed state diagrams. Additionally, it aims to generate high-quality cartoon images and codes by extracting various types of diagrams and state diagrams. This advancement will enable a transition from text-based requirements to visual storytelling, offering a new dimension to the fields of cartoon production and content creation.

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] R. Kaur and J. Sengupta, "Software process models and analysis on failure of software development projects," *International Journal of Scientific & Engineering Research(IJSER)*, vol.2(2), February 2012. DOI: https://doi.org/10.48550/arXiv.1306.1068.

[2] Y. Jin and R. Y. Kim, "UML State Diagram Extraction from Natural Language based Requirements with Linguistic Analysis Mechanism," ISAAC2023, AACL22, pp.111-114, 2023.

[3] C. Seo, J. Kim, W. Jang, B. Park, R. Y. Kim, "Skeleton Code Generation based on Design Thinking," Korean Institute of Smart Media, vol.10, pp.118-191, 2021.

[4] Y. Liu, and J. Bruel, "Modeling of Natural Language Requirements based on States and Modes," *2022 IEEE 30th International Requirements Engineering Conference Workshops(REW)*, pp.190-194, 2022. DOI: https://doi.org/10.1109/REW56159.2022.00043

[5] N. Chomsky, *Syntactic structures*, Mouton de Gruyter, 2002.

[6] "Parser Demo," Berkeley Neural Parser, *https://parser.kitaev.io/*.

[7] C. J. Fillmore. *The case for case*. Universals in Linguistic Theory. E. Bach and R. Harms. New York, Holt, Rinehart, and Winston, 1968.

[8] B. Park, and R. Y. Kim, "Effort estimation approach through extracting use cases via informal requirement specifications," *Applied Sciences*, vol. 10, No. 3044, pp. 1-15, 2020. DOI: https://doi.org/10.3390/app10093044

[9] J. Kim, and R. Y. Kim, "Cartoon Extraction Mechanism via UML Model based on Natural Language Requirement Specs.," 10th Annual Conf. on Computational Science & Computational Intelligence, 2023