I/O FSM으로 명세화된 철도 신호제어용 프로토콜 검정에 관한 연구
A Study on Verification of Rail Signal Control Protocol specified in I/O FSM

서미선* 황종규** 이재호** 김성운***
Seo, Mi-Seon Hwang, Jong-Gyu Lee, Jae-Ho Kim, Sung-Un

ABSTRACT

The verification confirms a correspondence between requirements and a specification before implementing. The problem in the formal method verifying a protocol specification using model checking is that the protocol behaviors must be always specified in LTS(Label Transition System). But if Region Automata is applied to the model checking, it is enable to verify whether properties are true on specification specified in I/O FSM(Input/Output Finite State Machine) as well as LTS.

In this paper, we verify the correctness of rail signal control protocol type 1 specified in I/O FSM by using model checking method and region automata. This removes many errors and ambiguities of an informal method used in the past and saves down expenditures and times required in the protocol development. Therefore it is expected that there will be an increase in safety, reliability and efficiency in terms of the maintenance of the signaling system by using the proposed verification methods.

1. 서론

프로토콜 검정이란 사용자 요구사항과 명세와의 일치성을 제품 구현 전에 확인하는 단계로, 모든 프로토콜에 필수적인 정확성을 만족하는지를 모형검사(model checking)방법을 사용하여 자동적, 형식적으로 검정하는 기술이다[1]. 검정을 위해 과거에 사용된 비정형적 방법은 명세의 모호함과 불완전성 등의 많은 오류와 비효율성을 내포하고 있다.

LTS(Labeled Transition System)로 명세화된 철도 신호제어용 프로토콜 모델의 안전성 및 필연성 특성을 모형검사 기법에 의해 검정하기 위해 대수적 명세기법인 modal mu-calculus[2]를 사용하면 행위에 의한 순환적 정의가 가능하므로 상태폭발 문제가 해결될 뿐만 아니라 자동적이고 효율적인 검정이 가능하다. 개발된 '프로토콜 검정기[3]'는 전체 철도 신호제어 시스템의 안정성 및 신뢰성을 보장하여 준다.

그러나 현재까지의 모형검사 기법에 의한 프로토콜 검정 방법은 LTS 기반이므로, 프로토콜 동작 행위를 반드시 LTS 모델로 명세해야만 하는 한계점이 존재한다. 일반적으로 프로토콜 개발에 많이 사용되는 I/O FSM (Input/Output Finite State Machine) 또한 기반이 된다면, I/O FSM으로 명세화된 프로토콜을 다시 LTS로 모델링 해야하는 번거로움 없이 더욱 빠른 검정 및 폭넓은 기능 수행이 가능할 것이다. 본 논문에서는 LTS 뿐만 아니라 I/O FSM을 기반으로 하고 모형검사 알고리즘인 Solve를 그대로 적용하기 위해 지역 오토마타(Region Automata) 알고리즘을 제안하고, 이를 사용하여 프로토콜을 효율적으로 검정하는 방법을 보인다.

이를 위해 2장에서는 검정대상인 철도 신호제어 프로토콜 type 1을 LTS 및 I/O FSM으로 명세하고, 3장에서는 I/O FSM 검정시 발생하는 livelock을 제거하기 위한 지역 오토마타를 소개한다. 그리고 4장에서는 지역 오토마타 알고리즘을 적용하여 모형검사 방법으로 철도 신호제어 프로토콜을 검정하는 방법을 예시를 통해 보이며, 마지막으로 5장에서 본 논문의 결론과 향후 추진 사항에 대해 기술한다.

* 부경대학교 정보통신공학과, 학생회원
** 철도연구원 선임, 책임연구원, 정회원
*** 부경대학교 정보통신공학과, 정회원

2. 철도 신호제어 프로토콜 type 1과 LTS 및 I/O FSM 비교 분석

2.1 철도 신호제어 프로토콜 type 1과 LTS 및 I/O FSM 명세

철도 신호 제어 프로토콜 type 1은 역정보전송장치(LDTS : Local Data Transmission System)와 전자연동장치(EIS : Electronic Interlocking System) 간의 정보 전송 방식으로, LDTS는 EIS에게 폴링 메시지와 진로설정이나 선로전환기 등을 제어하는 메시지를 전송하고, EIS는 LDTS에게 현장 신호설비들의 상태정보 메시지 및 제어메시지에 대한 응답(ACK)을 전송하여 상호간에 통신한다. 이를 LTS와 I/O FSM으로 명세화한 모델은 다음 그림 1, 2와 같다.

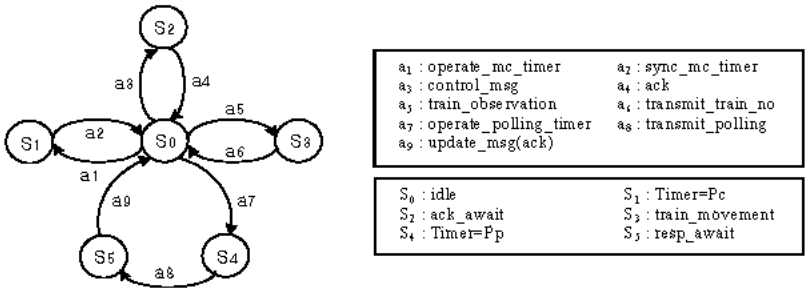
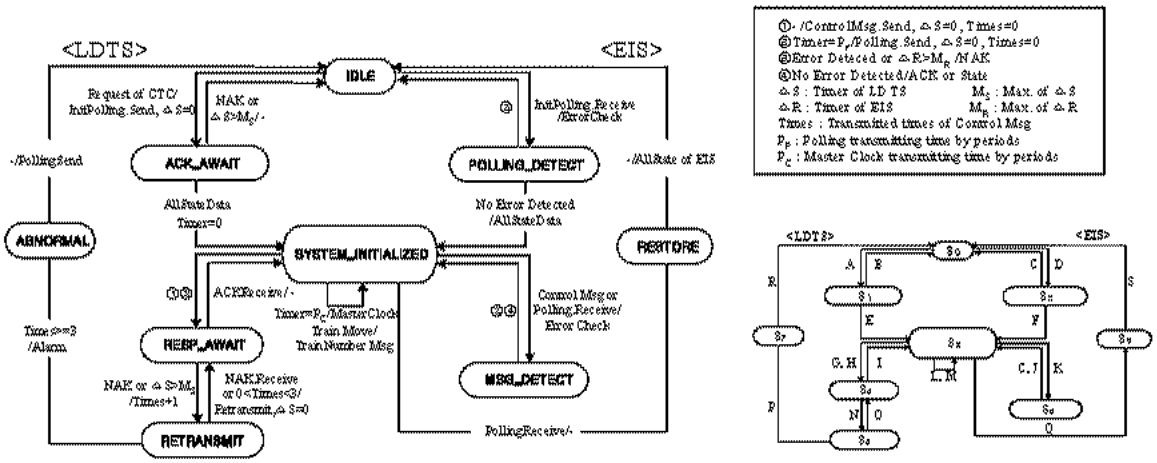


그림 1. 철도 신호 제어 프로토콜 type 1의 LTS 명세



(a) (b)

그림 2. 철도 신호 제어 프로토콜 type 1의 I/O FSM 명세(a)와 약어표현(b)

2.2 LTS와 I/O FSM의 비교 분석 및 I/O FSM의 변형

모형검사 방법을 이용하여 LTS로 명세화된 프로토콜을 검정하는 방법에 I/O FSM을 그대로 사용할 경우, I/O FSM의 많은 스테이트에서 livelock이 발생한다. 예를 들어, 그림 2의 I/O FSM 약어모델(b)를 검정기에 그대로 입력할 경우, S_0, S_1, S_2, S_3, S_4, S_5, S_6의 대부분의 스테이트에서 livelock이 발생한다. 이는 I/O FSM 명세 방법이 LTS를 기반으로 하는 검정방법과 통신의 상호동작개체 및 행위명세 기준에서 차이가 나기 때문이다. 따라서 I/O FSM을 모형검사 방법에 적절하게 변형해줘야 할 필요성이 생긴다.

LTS는 하나의 객체를 기준으로 그 객체의 스테이트와 행위의 천이의 관계를 나타낸 모델로서, 초기화 된 이후의 스테이트와 행위를 명세한다. LTS 명세모델의 쉬운 예로, 그림 3과 같이 10 coin을 넣고 coffee 버튼을 누르면 coffee가 나오고 20 coin을 넣고 milk 버튼을 누르면 milk가 나오는 자판기 LTS는 자판기 시스템의 전원이 on된 이후의 동작만을 명세한

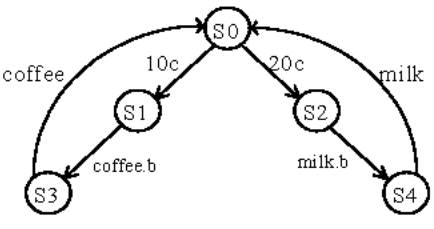


그림 3. 자판기 LTS

다.

반면, I/O FSM은 상호작용하는 두 개체간의 통신 동작을 명세화할 뿐만 아니라 시스템이 초기화되기 전의 상태인 idle 상태부터 모든 동작을 고려한다. 이 때 S_3(그림 2(b))에서 S_1과 S_2로 향하는 행위가 발생하지 않기 때문에 항상 이곳에서 livelock이 발생한다. 따라서 LTS 기반의 모형검사 알고리즘을 I/O FSM에 그대로 적용하기 위해서는 그림 2의 I/O FSM 모델에서 시스템 초기화 이전의 상태인 S_0, S_1, S_2를 제외하고 검정해야 한다. S_0, S_1, S_2를 제외한 모델은 그림 4와 같다.

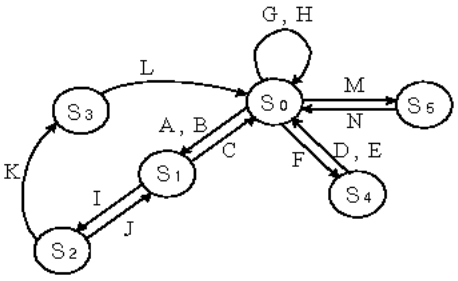


그림 4. 리모델링된 I/O FSM

3. 지역 오토마타

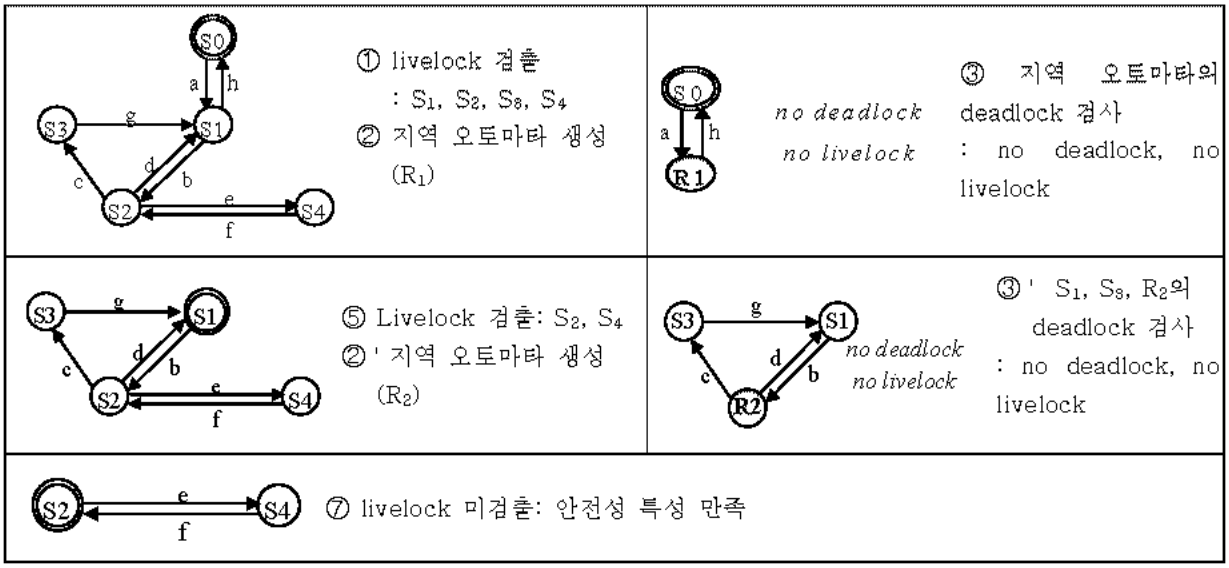
I/O FSM을 LTS와 동일한 조건을 갖는 모델로 만들었음에도 그림 4를 모형검사 방법에 의한 프로토콜 검정기법에 그대로 적용할 경우, S_1과 S_2에서 livelock이 발생한다. 이러한 livelock을 제거하기 위해 지역 오토마타 알고리즘을 적용한다. 지역 오토마타란, 전체 오토마타의 검정을 수행하여 livelock이 걸린 상태만으로 새로운 오토마타를 만들고 새롭게 생성된 부분적인 오토마타를 검사하는 방식의 검정 방법이다.

수행 알고리즘은 다음과 같다.

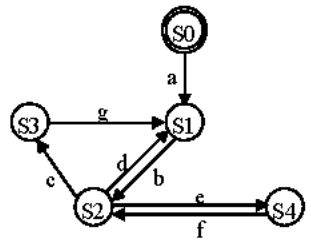
① I/O FSM에 모형검사 알고리즘인 Solve알고리즘을 수행하여 livelock이 걸린 상태 검사
② livelock이 걸린 상태만을 포함하는 지역 오토마타를 생성
③ 지역 오토마타의 deadlock 검사
④ deadlock이 발생하면 지역 오토마타 내의 모든 상태는 livelock
⑤ deadlock이 없으면 지역 오토마타의 livelock을 검사
⑥ 지역 오토마타가 livelock이 아닐 때까지 2~5 반복수행
⑦ 마지막 지역 오토마타에서도 deadlock이 검출되지 않으면 I/O FSM은 안전성 특성을 만족

위의 지역 오토마타 알고리즘을 이용하여 livelock이 발생하지 않은 I/O FSM과 livelock이 발생한 I/O FSM을 검정한 예를 보인다.

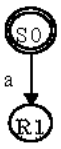
➢ livelock이 발생하지 않은 예



➢ livelock이 발생한 예



① livelock 검출
: S_1, S_2, S_3, S_4
② 지역 오토마타 생성 (R_1)



deadlock
no livelock

③ 지역 오토마타의 deadlock 검사
: deadlock, no livelock

해당 예시는 단계 ④에서 Deadlock이 발생했으므로 지역 오토마타는 livelock이고 안전성 특성을 만족하지 못함을 판단한다.

4. 지역 오토마타 알고리즘을 적용한 철도 신호제어 프로토콜 검정

본 소절에서는 Modal mu-calclulus 논리식에 Solve 알고리즘[4]을 적용하여 LTS 모델 검정 결과를 출력하는 기존의 모형검사 방법에 제안된 지역 오토마타 알고리즘을 추가 적용하여 I/O FSM을 검정하는 과정을 기술한다. I/O FSM으로 명세화된 그림 4 모델의 안전성 특성[2](deadlock 및 livelock 이 없음) 검정을 위한 modal mu-calculus의 식은 $\nu Z.\ (\mu Y.A \vee(\langle - \rangle tt \wedge [-]Y)) \wedge [-]Z$, $A=\{S_0\}$이다.

그림 5는 위 식에서 최대고정점과 최소고정점을 사용하여 생성된 max block과 min block을 나타내고, 그림 6은 max block과 min block의 변수 Xi들의 천이 관계를 나타낸 edge-labeled directed graph G이다.

$B_1 = \min\{$	$B_2 = \max\{$
$X_1 = X_2 \vee X_3$	$X_7 = X_1 \wedge X_8$
$X_2 = A$	$X_8 = [-]X_7\}$
$X_3 = X_4 \wedge X_5$	
$X_4 = [-]X_1$	
$X_5 = \langle - \rangle X_6$	
$X_6 = tt\}$	

그림 5. Max block, min block

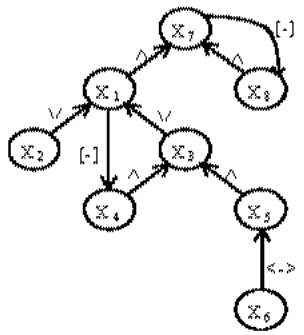


그림 6. Edge-labeled directed graph G

그림 7은 solve 알고리즘의 초기화 규칙에 의해 초기화된 bit-vector, counter 및 배열을 나타낸다. 그림 8은 배열 M[i]가 공집합(empty)이 될 때까지 갱신 알고리즘을 적용하여 bit-vector와 counter를 갱신한 결과로, 검정 대상인 deadlock, livelock을 판단할 수 있다.

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_0	0	1	0	0	0	1	1	1
S_1	0	0	0	0	0	1	1	1
S_2	0	0	0	0	0	1	1	1
S_3	0	0	0	0	0	1	1	1
S_4	0	0	0	0	0	1	1	1
S_5	0	0	0	0	0	1	1	1

C	X_3	X_4
S_0	2	4
S_1	2	2
S_2	2	2
S_3	2	1
S_4	2	1
S_5	2	1

M[1]=<<S_0, X_7>, <S_0, X_8>, <S_1, X_8>, <S_2, X_8>, <S_3, X_8>, <S_4, X_8>, <S_5, X_8>
M[2]=< >

그림 7. 초기화된 bit-vector, counter 및 배열

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_0	1	1	0	0	1	1	0	0
S_1	0	0	0	0	1	1	0	0
S_2	0	0	0	0	1	1	0	0
S_3	1	0	1	1	1	1	0	0
S_4	1	0	1	1	1	1	0	0
S_5	1	0	1	1	1	1	0	0

C	X_3	X_4
S_0	1	1
S_1	1	1
S_2	1	1
S_3	0	0
S_4	0	0
S_5	0	0

M[1]=< >
M[2]=< >

그림 8. Bit-vector와 counter를 갱신한 결과

Deadlock은 bit-vector의 요소에 의해 판단되는데, 그림 8의 결과에서 bit-vector의 X_5의 요소는 모두 1로 갱신되어 deadlock이 발생되지 않았음을 알 수 있다. 또한 livelock은 관련된 상태의 counter 요소에 의해 판단되는데 위의 결과에서 counter의 요소 S_0, S_1, S_2의 요소가 1이 됨으로서 livelock이 검출된다. 그러므로 livelock이 검출된 S_1, S_2로 구성(초기상태인 S_0를 제외)되는 지역 오토마타를 생성하여, 위의 Solve 알고리즘을 다시 적용한다.

지역 오토마타를 포함한 I/O FSM은 그림 9와 같다. Solve 알고리즘을 다시 적용하여 bit-vector, counter 및 배열을 초기화 한 것은 그림 10이고, 이의 갱신 결과는 그림 11과 같다. 결과의 모든 bit-vector의 X_5 원소가 1이고, counter는 0로 갱신됨을 통해 그림 9의 I/O FSM에서는 deadlock과 livelock이 검출되지 않음을 알 수 있다.

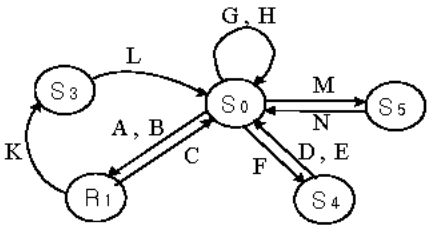


그림 9.지역 오토마타를 포함한 I/O FSM

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_0	0	1	0	0	0	1	1	1
r_1	0	0	0	0	0	1	1	1
S_3	0	0	0	0	0	1	1	1
S_4	0	0	0	0	0	1	1	1
S_5	0	0	0	0	0	1	1	1

C	X_3	X_4
S_0	2	3
r_1	2	2
S_3	2	1
S_4	2	1
S_5	2	1

M[1]=<<S_0, X_2>, <S_0, X_6>, <r_1, X_6>, <S_3, X_6>, <S_4, X_6>, <S_5, X_6>>
M[2]=< >

그림 10. 지역오토마타를 포함한 I/O FSM 초기화

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_0	1	1	1	1	1	1	1	1
r_1	1	0	1	1	1	1	1	1
S_3	1	0	1	1	1	1	1	1
S_4	1	0	1	1	1	1	1	1
S_5	1	0	1	1	1	1	1	1

C	X_1	X_2
S_0	0	0
S_1	0	0
S_3	0	0
S_4	0	0
S_5	0	0

M[1]=< >
M[2]=< >

그림 11. 지역오토마타를 포함한 I/O FSM 검정 결과

마지막으로 지역 오토마타만을 검사하여 내부에 deadlock 또는 livelock이 있는지를 판단한다. 지역오토마타 R_1 내부의 행위는 그림 12과 같고, 이를 검정하기위한 초기화 및 결과는 그림 13, 14와 같다.

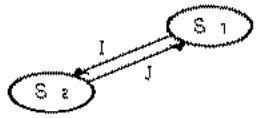


그림 12. R_1의 I/O FSM

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_1	0	1	0	0	0	1	1	1
S_2	0	0	0	0	0	1	1	1

C	X_2	X_4
S_1	2	1
S_2	2	1

M[1]=<<S_1, X_2>, <S_1, X_6>, <S_2, X_6>>
M[2]=< >

그림 13. 지역오토마타의 검정 초기화

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
S_1	1	1	1	1	1	1	1	1
S_2	1	0	1	1	1	1	1	1

C	X_2	X_4
S_1	0	0
S_2	0	0

M[1]=< >
M[2]=< >

그림 14. 지역오토마타의 검정 결과

그림 14의 결과에서 bit-vector X_5의 원소가 1이고, counter의 모든 요소가 0로 갱신됨을 통해 지역 오토마타 내부의 행위는 안전성 특성을 만족함을 알 수 있다.

지역 오토마타가 livelock이 아닐 때까지 검정 알고리즘을 수행하여, 마지막 지역 오토마타에서도 deadlock이 검출되지 않았으므로 그림 4의 I/O FSM 모델은 안전성 특성 논리식인 vZ. (μY.A ∨(<-> tt ∧ [-]Y)) ∧ [-]Z, A={S_0}를 만족하는 완전한 프로토콜 모델임을 판단한다.

5. 결론 및 향후 계획

본 논문에서는 철도 신호 제어 프로토콜 type 1을 대상으로 I/O FSM으로 모델링된 프로토콜을 기반으로 시스템 특성 언어인 Modal Mu-Calculus를 이용하여 행위에 의한 순환적 정의를 통해 모형검사기법을 적용하고, 또 제안된 지역 오토마타 알고리즘을 적용하여 프로토콜 검정을 효율적으로 수행함을 보였다. 본 논문에서 사용된 알고리즘은 LTS와 I/O FSM 어느 명세 방법에도 적용이 됨으로써, 시스템 모델을 매우 정확하고 명료하게 검사하도록 하여 비형식적 방법에서 야기될 수 있는 오류

와 모호함을 제거하였다.

해당 프로토콜의 정확성을 검정한 결과에 따르면, 역정보전송장치(LDTS)와 전자연동장치(Electronic Interlocking System) 간의 상태정보 전송방식 프로토콜인 type 1 프로토콜은 안전성을 만족하는 규격으로 그 내용이 검정되었다.

향후 추진 사항은 I/O FSM을 이용하여 명세화된 철도 프로토콜을 효율적으로 검정하기위해 기 개발된 프로토콜 검정기에 제안된 지역 오토마타 알고리즘을 삽입하여 실제적이고 자동적인 검정이 가능하도록 하는데 있다.

참고 문헌

1. D. Schwabe, "Formal Techniques for the Specification and Verification of Protocol, Ph.D Thesis", Univ. of California Los Angeles, Apr., 1981.
2. 서미선, 김성운, 황종규, 이재호, "LTS로 명세화된 철도 신호제어용 프로토콜 검정 및 적합성시험", 한국철도학회 추계학술대회, p.581-586, 2003.10.
3. 서미선, 황진호, 황종규, 이재호, 김성운, "신뢰성 확보를 위한 철도 신호제어용 프로토콜 검정기 개발", 한국철도학회 춘계학술발표대회, p.389, 2004.6.
4. R. Cleaveland, B. Steffen, "A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus", Formal Methods in System Design 2(2) : p.121-147, 1993.

후 기

본 연구는 철도청 철도기술연구개발사업으로 지원된 "신뢰성 확보를 위한 프로토콜 검정기 및 적합성시험 생성기 기능 확장 개발" 과제의 연구결과의 일부입니다.