Journal of Korea Multimedia Society Vol. 19, No. 8, August 2016(pp. 1544-1552)
http://dx.doi.org/10.9717/kmms.2016.19.8.1544

디자인 패턴을 이용한 Undo 기능 설계에 대한 연구

김태석†, 김종수††

A Study on the Undo Function Implementation using the Design Patterns

Tai Suk Kim†, Jong Soo Kim††

ABSTRACT

If the undo/redo function is not reflected in the initial design of an application, it makes it difficult to implement the undo/redo function additionally, in this paper, we examined some examples to design the sodoku game and analyzed problems of the design to implement the undo/redo functions. For an efficient design of the undo/redo functions without using swing.undo package, we propose a class design using the Command, Memento, and Observer pattern these are used as organic. The proposed method is more efficient for distributed work than other method. We implemented a sudoku game using proposed design. In the undo/redo function testing, we could see that it works well.

Key words: Undo, Design Patterns, Command Pattern, Memento Pattern, Observer Pattern

1. 서 론

워드프로세스, 이미지 저작도구와 같이 특정한 작업에 필요한 데이터를 메모리로 읽어 들여서 편집하는 것을 주요 기능으로 하는 소프트웨어는 잘못된 작업을 바로 취소할 수 있는 Undo 기능의 지원이 필수적이라 할 수 있다. 이러한 Undo 기능의 구현은 텍스트 데이터를 기반으로 하는 프로그램뿐만 아니라, 그래픽 프로그램인 경우 더욱 중요한 기능이며, 해당 기능의 구현이 소프트웨어 사용자가 비생산적인 실수를 하더라도 전체 작업 스케줄에 지장 없이 안정적으로 작업할 수 있는 여건을 마련해 준다. Undo와 Redo 기능이 필요한 어플리케이션은 이와 관련된 기능의 구현을 설계 초기에 반영해 놓지 않으면, 해당 기능을 구현하기 위해 많은 코드를 수정해야하는 번거로움이 생길 수 있다. 객체 지향적으로 설계되는 소프트웨어의 설계와 구현에 있어서도 Undo와 Redo 기능의 지원이 필요한 경우, 방대한 양의 코드를 효율적으로 구현하기 위해서는 Undo와 Redo 기능이 효율적으로 구현될 수 있는 설계를 바탕으로 필요한 나머지 기능을 구현해 나가는 것이 해당 기능과 전체 어플리케이션 구현에 효과적인 경우가 많다.

2. 관련연구

2.1 선행 연구

현재 구글의 앱 스토어나 국내의 앱마켓에서는 Fig. 1과 같은 다양한 스도쿠 게임이 있으며, 해당 스도쿠 게임에는 Undo 기능이 구현되어 있는 것도 있으며, 그렇지 않은 것도 있다[1]. 리눅스 기반의 Open 소스 스도쿠 프로그램도 있지만, 이 경우, 객체

※ Corresponding Author : Tai Suk Kim, Address: Dept. of Computer Software Engineering, Dong-eui University 176 Eomgwangno Busan_jin_gu, Busan 614-714, Korea, TEL : +82-51-890-1707, FAX : +82-51-890-1724, E-mail : tskim@deu.ac.kr
Receipt date : Jun. 13, 2016, Revision date : Jun. 30, 2016
Approval date : Jun. 30, 2016
† Dept. of Computer Software Engineering, Dong-Eui University
†† Dept. of System Management, Korea Lift College (E-mail : seatree@klc.ac.kr)

Fig. 1. Some examples of the sudoku game.

지향적으로 설계되지 않아서 Android와 같이 객체 지향언어를 사용하는 경우 오픈되어 있는 기존의 소스를 재사용하는데 편리하지 않다.

Android 기반으로 구현된 스도쿠 앱도 많지만, 소스가 오픈된 경우가 매우 드물며, 스도쿠 게임서 객체지향개념의 근간이 되는 클래스의 캡슐화를 파괴하지 않고, 분산작업이 용이한 Undo 기능의 설계를 참고하기는 힘든 경우가 많다. 위에 그림 중에서 가장 오른쪽에 빨간색 박스로 감싸진 스도쿠 게임은 소스가 오픈된 스도쿠 게임이다[2]. Fig. 2는 오픈소스 스도쿠에서 사용하고 있는 Command Pattern의 구현을 보여준다.

다이어그램에서, 수행되는 명령어를 객체로 관리하기 위해 Stack 클래스를 사용하여 Command Stack 클래스 객체를 생성하고 있음을 볼 수 있고, excute()와 undo() 추상 메서드를 가지는 Abstract Command 클래스를 상속받는 AbstractCellCommand 추상 클래스와 AbstractCellCommand를 다시 상속받는 ClearAllNotesCommand, EditCellNoteCommand, FillInNotesCommand, SetCellValueCommand 클래스가 있음을 볼 수 있다. 설계에서 다소 비효율적인 부분은 추상 메서드가 정의되어 추상클래스로 정의

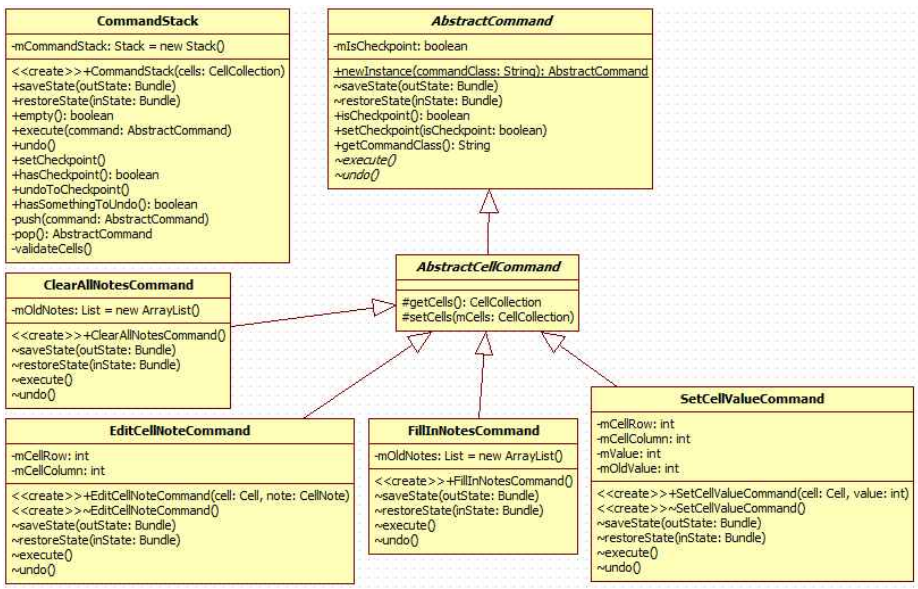

Fig. 2. Command Pattern implementation of the open source sudoku.

된 AbstractCommand를 상속 받는 AbstractCell Command는 해당 클래스에서 정의된 추상메서드가 없음에도 불구하고 추상클래스로 정의된 것을 볼 수 있다. 따라서 AbstractCommand 클래스와 Abstract CellCommand 클래스는 통합되는 것이 클래스관리와 구현에 편리하다고 할 수 있다.

Fig. 3은 오픈 스도쿠에서 게임을 구현하기 위한 주요 클래스 설계를 보여준다.

SudokuGame 클래스의 excuteCommand() 메서드는 Command 패턴으로 구현된 임의의 자식 클래스를 저장하기 위하여 부모 클래스인 Abstract Command를 아규먼트로 받고 있는 것을 볼 수 있다. 또한, undo 기능을 구현하기 위하여, 그림의 빨간 박스에서 보는 바와 같이, undo(), hasSomething ToUndo(), setUndoCheckpoint(), undoToCheckpoint (), hasUndoCheckpoint()와 같은 메서드들이 Sudoku Game 클래스에 포함되어 있음을 볼 수 있다. 다이어그램에서는 undo 기능의 구현을 위해 관련된 기능의 구현을 별도의 클래스로 분리하지 않은 것을 볼 수 있는데, 이와 같은 구현은 추후 지속적인 기능의 향상을 위하여 필요한 메서드의 개수가 지속적으로 늘어날 수 있다는 가정을 하면, SudokuGame 클래스가 전체 어플리케이션 설계에서 너무 많은 역할을 담당하게 되는데, 이는 여러 명의 개발자들이 동시에 진행하는 소프트웨어 분산작업에서 일부 클래스가 비대해짐으로써, 전체 개발일정을 지연시키는 결과를 초래할 수 있다.

2.2 GoF 디자인 패턴을 이용한 구현

Swing의 undo 패키지를 활용할 수 없는 경우에는 Undo/Redo 기능을 구현하기 위해서 GoF 디자인 패턴을 활용할 수 있다. 그래픽을 다루는 것이 주요 기능인 프로그램의 Undo/Redo 기능 구현에 있어서 디자인 패턴을 이용한 예는 http://www.developer.com과 같은 여러 사이트에 소개되어 있다[3-8]. Undo/Redo 기능 구현에 주로 사용되는 패턴으로는 Command, Memento, 그리고 Observer 패턴이 있다. 간단한 Undo/Redo 기능의 구현에는 Memento나 Observer가 필요 없는 경우도 있지만, 그래픽 처리가 주요 기능인 복잡한 어플리케이션의 Undo 기능 구현에는 이들 패턴들이 유기적으로 결합되어 해당 기능이 구현된다.

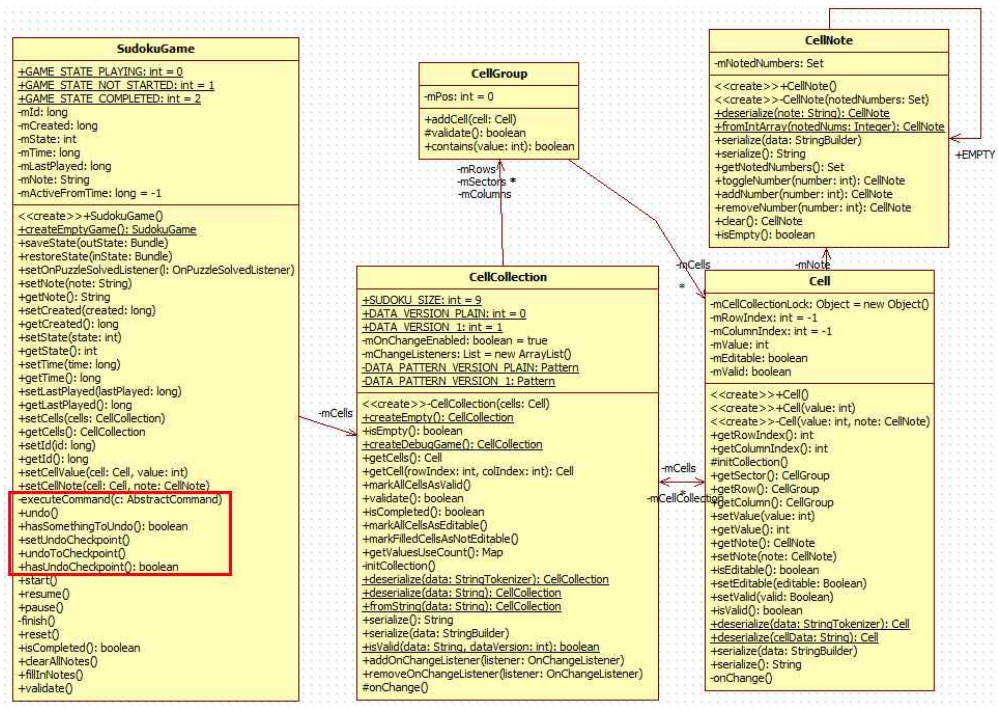

Fig. 3. related methods for implementing undo function in open sudoku.

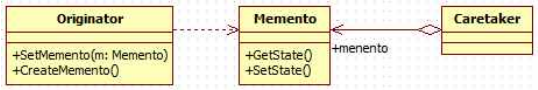

Fig. 4. Memento pattern.

일반적으로 Command 패턴은 명령을 트랜잭션 형태로 수행해야하는 경우나, 콜백 기능을 구현하고자 할 때 사용할 수 있다. Undo 기능의 구현을 위해서 사용자가 발생시키는 명령어를 LinkedList나 Stack과 같은 자료구조를 이용하여 수행될 명령어의 인스턴스를 저장하고, Undo/Redo 기능 호출시 해당 자료구조를 이용하여 관련명령어를 수행하는데 참고한다.

Fig. 4에 있는 Memento 패턴은 Undo/Redo 기능 구현에 관련된 객체들에 대하여 해당 시점의 상태를 기억하는 역할을 담당하는데, Command 패턴이 가지고 있는 명령어를 저장하는 스택을 참조하여 Undo/Redo 기능을 순차적으로 구현할 수 있도록 도와준다.

Observer 패턴은 상태변화 감시가 필요한 다양한 객체들을 제어하기 위한 패턴으로 Undo/Redo 기능 호출시 어플리케이션에서 전체적으로 다시 그려야 하는 컴포넌트들을 관리하는 역할을 담당한다.

3. 스도쿠 게임의 Undo 기능 설계

3.1 GUI 설계

GUI 설계에 많은 시간이 들어가는 게임의 구현에 있어서도 Undo 기능의 구현이 필요한 경우가 있다. 특히 기억력을 소재로 하는 게임이나, 교육용 게임에서 Undo 기능의 구현이 필요한 경우가 많다. Fig. 5에서 Undo 기능이 구현된 스도쿠의 GUI 설계를 볼 수 있다.

새로 설계된 어플리케이션은 FrameLayout을 가진다. 한 개의 Cell 클래스는 입력될 수 있는 1~9까지에 숫자를 가지며, 문제가 출제되어 숫자가 해당 Cell에 부여되거나 사용자가 정답으로 입력하는 숫자는 Cell의 중앙에 숫자의 색상과 크기를 다르게 하여 출력된다. FrameLayout의 중앙영역은 전체 9×9 Cell 객체를 배열로 관리하는 SudokuPanel, 북쪽 영역에는 퀴즈 출제, 문제 풀이 검증, 환경설정 등을 선택할 수 있는 MenuPanel, 남쪽 영역에는 선택된 셀에 숫자를 입력할 수 있는 숫자 버튼과 기존에 입

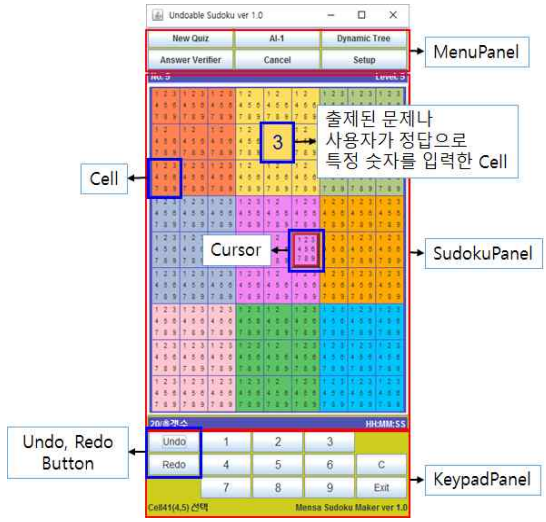

Fig. 5. GUI Design of a sudoku game.

력된 숫자를 취소하거나 되돌릴 수 있는 Undo와 Redo 버튼이 포함된 KeypadPanel이 있다.

3.2 GoF 패턴 적용한 클래스 다이어그램

스도쿠 게임에서 GoF 패턴을 이용하여 Undo 기능을 구현한 설계의 예는 Fig. 6에서 볼 수 있다.

Keypad를 이용하여 지정한 숫자를 입력하는 기능을 구현하기 위하여 GoF의 Commad 패턴을 사용하였다. Command 인터페이스를 구현한 추상 클래스인 AbstractAnswerCommand를 상속받는 Answer1Command 클래스를 사용하여 클릭된 숫자에 해당하는 정답을 가진 인스턴스가 생성된다. 게임이 시작되면서 숫자 버튼을 클릭하여 입력되는 명령을 추적하기 위해 CommandStatck 클래스의 LikedList 자료구조를 가지는 undoStack, redoStatck 객체변수가 사용된다. 명령어가 수행되는 현재의 상태를 저장하기 위하여 MementoOriginator 인터페이스를 구현한 PathCell 클래스가 사용되며 명령어가 수행될 때 마다 기존의 히스토리를 저장하는 Memento 객체를 생성한 후, PathCell 클래스의 pathCellList 객체변수에 저장한다. 9×9 Cell을 전체적으로 관리하는 SudokPanel 클래스는 Undo/Redo 명령이 수행될 경우 전체 Panel을 다시 그려야 되므로 Displayable 인터페이스를 구현하고, PathCell 클래스에 저장된 pathCellList가 어떠한 단계까지 Cell을 다시 그려야할 것인지에 대한 정보를 알려준다.

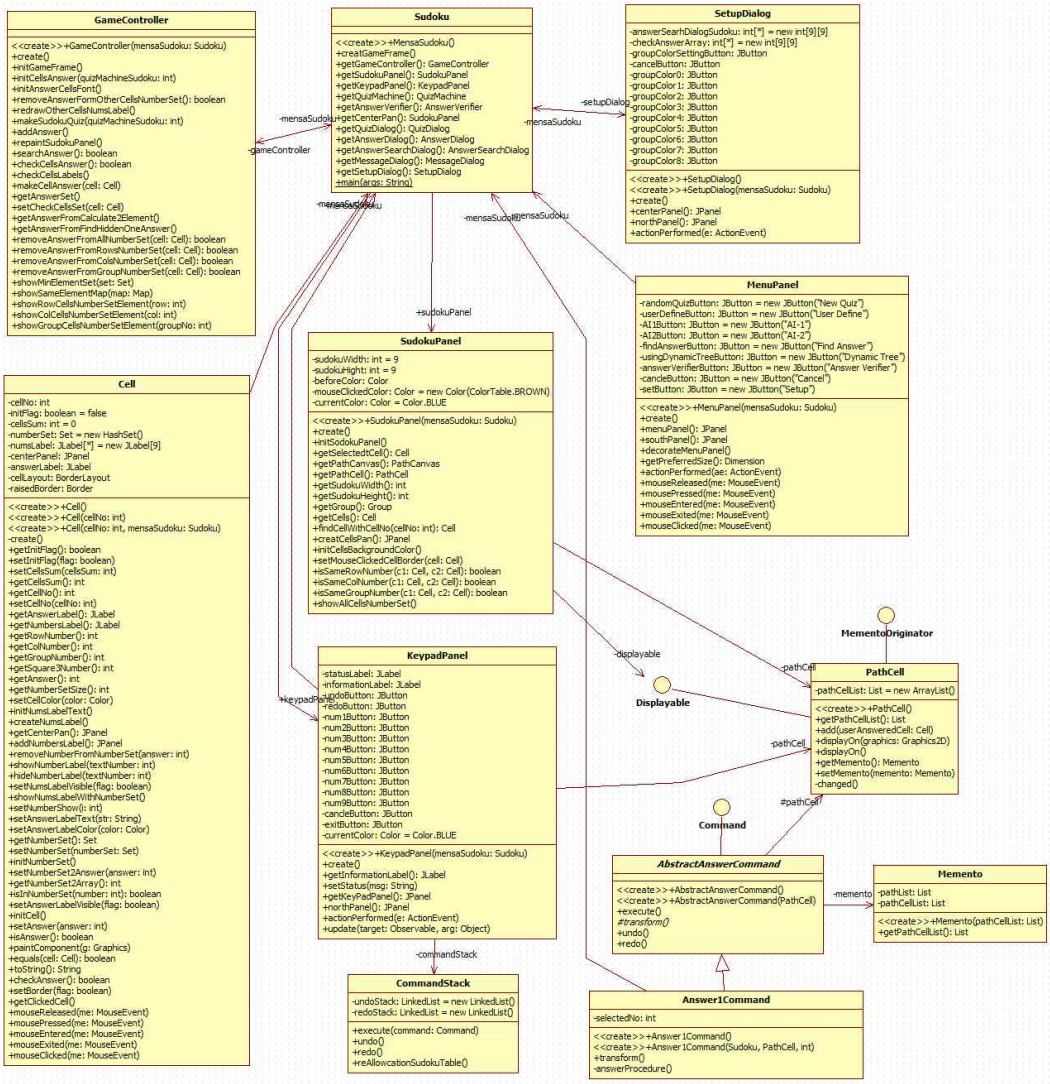

Fig. 6. Class design of the undoable sudoku game using GoF design patterns.

3.3 시퀀스 다이어그램

Undo 명령을 구현하기 위한 첫 번째 절차는 CommandStatck의 excute() 메서드를 호출하는 것으로부터 시작된다. Fig. 7의 시퀀스 다이어그램에서 클릭된 명령버튼의 인스턴스를 저장하고 관련된 처리를 수행하기 위한 메서드 수행절차를 볼 수 있다.

사용자가 숫자 버튼을 클릭함으로써 Command Statck의 excute() 명령이 호출된다. 다음으로 Command 인터페이스를 구현하고 있는 추상 클래스인 AbstractAnswerCommand를 상속받는 Answer1 Command 객체의 excute()가 호출된다. 다음으로 Answer1Command 객체는 PathCell 객체의 get Memento()를 호출한다. PathCell 객체는 숫자버튼이 클릭되는 순서와 해당 객체를 List 자료 구조에 저장하기 위한 객체인데, 현재 까지 클릭된 Cell 객체들에 대한 상태를 순서대로 저장하기 위해 생성된 List 정보를 이용하여 Memento 객체를 가져온다.

Command가 Memento를 호출하여 현재까지의 상태에 대한 정보를 객체로 전달 받는 메서드가 수행이 완료되면 transform() 메서드를 호출하여 Path Cell 객체에 숫자가 입력된 Cell의 정보를 저장하고, 선택된 Cell에 사용자가 입력한 숫자를 정답으로 처

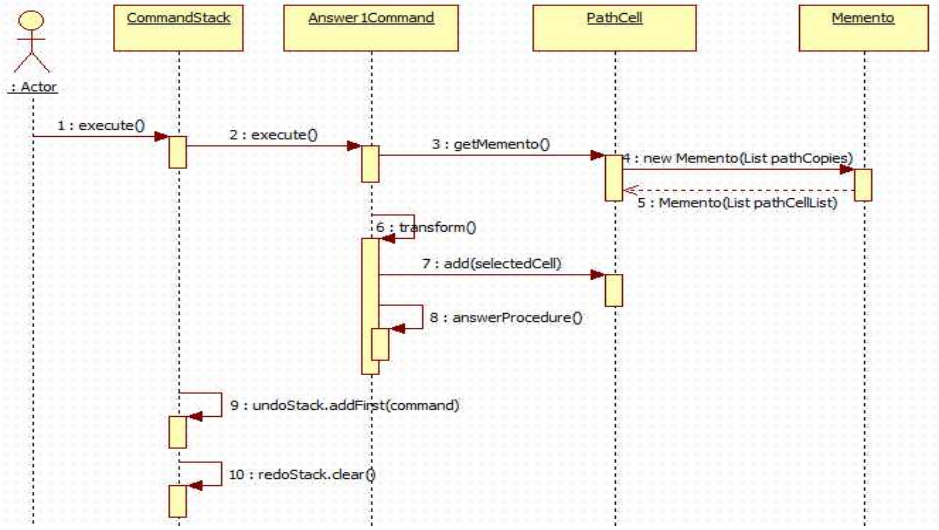

Fig. 7. Sequence diagram to save the state on clicked a number button.

리한 후, 정답이 입력되면 수행되어야할 처리를 추가적으로 수행한다. 이 메서드의 수행이 완료되면 CommandStatck에서 가지고 있는 LikedList 구조의 undoStack에 현재 클릭된 Cell 객체를 저장하고, redoStack을 초기화한다. Command와 Memento가 Undo와 관련된 데이터 처리를 끝내면 Observer 패턴의 참가자인 Displayable 인터페이스를 구현하고 있는 SudokuPanel 클래스에서 관리하고 있는 81개의 Cell 객체를 다시 그린다.

undo 기능 구현과 관련된 Sequence 다이어그램은 Fig. 8에서 볼 수 있다.

사용자가 undo 버튼을 클릭하면 CommandStack의 undo() 명령이 호출된다. CommandStack 클래스가 가지고 있는 undoStack에 값이 들어 있지 않으면 바로 리턴하며, 값이 한 개라도 있는 경우 removeFirst()를 호출하여 제일 처음에 있는 Command를 제거한 후, Answer1Command의 undo()를 호출한다. 다음으로 PathCell 객체의 getMemento() 메서드

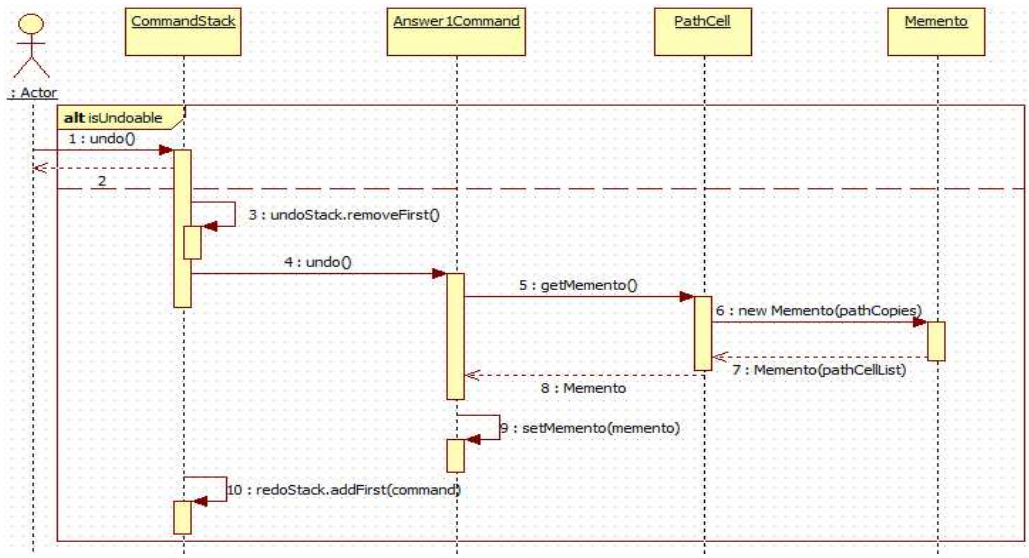

Fig. 8. Undo sequence diagram.

를 호출한다. getMemento()는 현재까지의 숫자버튼 클릭 순서를 저장하고 있는 List형의 pathCopies 객체를 이용하여 Memento 객체를 새로 생성하여 이를 돌려받는다. 다음으로 Answer1Command의 set Memento 메서드가 돌려받은 Memento 객체를 이용하여 undo와 관련된 작업을 수행한다. undo 작업이 완료되면 CommandStatck의 redoStatck에서 해당 Command를 삭제한다. CommandStatck 클래스에 있는 excute()의 sequence 다이어그램과 마찬가지로 Command와 Memento가 Undo와 관련된 데이터 처리를 끝내면 Observer 패턴을 사용하는 Sudoku Panel 객체에서 관리하고 있는 81개의 Cell을 다시 그린다.

4. 구현 및 평가

4.1 구현

소프트웨어의 Undo 기능 설계에 디자인 패턴을 활용하는 이유는 객체지향 개념의 근간이 되는 캡슐화를 위배하지 않으면서, 추후 이와 유사한 소프트웨어 구현에 해당 클래스를 재사용할 수 있다는 장점이 있기 때문이다.

자바 기반의 어플리케이션에서 Undo 기능을 구현하려면 특정 명령어가 실행되는 시점에서 메모리에 생성된 인스턴스들이 가지고 있는 정보를 저장할 필요가 있다. 전체 어플리케이션 구현에 클래스의 캡슐화를 파괴하지 않고, 특정 시점으로 인스턴스를 복원하기 위해서는 순차적으로 저장된 인스턴스 내부의 정보를 자유롭게 액세스할 수 있어야 하고, 이를 위해 GoF 패턴을 활용할 수 있다. Undo/Redo 기능이 있는 스도쿠 게임을 구현하기 위하여 GoF 디자인 패턴 중에서 Command, Memento, 그리고 Observer 패턴을 이용한 스도쿠 게임 구현을 Fig. 9에서 볼 수 있다.

SudokuPanel이 관리하는 Cell 배열에서 Cell의 좌표 번호 (6,4), (6,5), (6,6)에 차례대로 2, 7, 1을 입력한 후, Keypad 패널에 있는 Undo와 Redo 버튼을 클릭한 결과 Undo와 Redo 기능이 설계대로 구현된 것을 볼 수 있다.

4.2 평가

객체지향적인 분석과 설계에서는 주어진 요구사항을 해결하기 위하여, 현실 세계에 있는 여러 가지 사물들에 대하여 클래스라는 이름의 프로토타입을 만든다. 클래스를 설계하는 근간이 되는 개념은 다른 사물들과 구분되는 속성의 정의와 속성을 사용하는 메서드의 캡슐화이며, 설계와 소프트웨어의 구현을 보다 효율적으로 수행하기 위해 상속, 동적 할당과 같은 추가적인 개념이 포함된다.

일반적으로, 어떠한 어플리케이션을 사용하는 최종 사용자의 입장에서는 설계가 어떻든 간에 자신이 원하는 기능만 빠르고 정확하게 수행이 되지만, 지속

Fig. 9. Execution example of undo and redo function.

Table 1. Performance evaluation of a class design using design patterns

	Command 패턴	Memento 패턴	Observer 패턴	Undo 구현	Redo 구현
기타 Sudoku(앱스토어)	-	-	-	×	×
Open source sudoku	○	×	×	○	×
Undoable Sudoku	○	○	○	○	○

(○: applied, ×: not applied)

적으로 소프트웨어 개발에 종사해야하는 개발자들의 입장에서는 소프트웨어의 효율적인 설계가 추후 일어날 수 있는 작업량을 줄일 수 있어서 시간과 비용의 절감에 아주 효율적인 경우가 많으며, 이러한 이유에서 기존의 설계보다 효율적인 설계는 항상 있을 수 있다. 객체지향 패러다임의 소프트웨어 설계와 구현에서 가장 핵심이라고 할 수 있는 클래스 설계에서도 일반적으로 통용되는 클래스 설계에 다음과 같은 지침들이 있다.

① 간결한 클래스를 위하여 최소한의 공개 인터페이스 제공
② 클래스를 사용하는 다른 사용자에게 영향을 주지 않기 위한 구현 은익
③ 속성 초기화와 메모리 관리를 위한 견고한 생성자 설계
④ 다른 클래스와의 협력 관계를 고려한 느슨한 결합
⑤ 재사용을 고려한 설계
⑥ 확장성을 고려하여 설계
⑦ 코딩 표준을 위한 속성과 메서드의 명명 규칙 적용
⑧ 유지보수가 더 편리하게 만들기 위한 코드 분리

이 외에도 다양한 설계지침이 있지만, 본 논문에서는 구현된 소프트웨어의 성능평가를 위해 지침 중에서 몇 가지를 사용하여 기존에 구현된 다른 소프트웨어와의 성능을 평가하였고, Table 1에 그 결과를 정리하였다.

표에서 "-"는 소스가 공개되지 않아서 파악할 수 없음을 나타낸다. 앱 스토에서 제공되는 게임의 대부분은 소스가 공개되지 않아서, 디자인 패턴의 적용 여부를 분명하게 파악할 수 없었고, Undo나 Redo 기능이 구현되지 않은 앱이 다수 있었다[1]. 논문에서 Undo와 Rodo 기능을 비교하기 위해 분석된 오픈 소스 스도쿠는 Command 패턴이 사용되었으며, Undo 기능은 구현되었으나, Redo 기능은 구현되지 않았다[2]. 이에 반하여 새로 설계된 가칭 "Undoable Sudoku"는 클래스의 분산을 위해 Command, Memento 그리고 Observer패턴을 유기적으로 사용하여 Undo 기능을 구현하였고, 추가적으로 Redo 기능도 구현한 것을 볼 수 있다.

5. 결 론

Undo/Redo 기능이 필요한 어플리케이션의 경우, 그와 관련된 설계를 미리 반영해 놓지 않으면, 해당 기능의 구현이 어려운 경우가 많다. 본 논문에서는 Undo/Redo 기능 구현이 필요한 어플리케이션의 설계와 구현을 위하여 해당 기능을 구현하는 몇몇 사례를 조사하였다. 안드로이드 기반으로 구현된 오픈 소스 스도쿠 게임을 대상으로 클래스 다이어그램을 분석하여 스도쿠 게임에서 Undo 기능을 구현하기 위한 설계에서 몇몇 문제점을 고찰하였다. 해당 스도쿠는 Undo 기능의 구현을 위해 GoF의 Command 패턴을 사용하였지만, 필요 없는 클래스를 추가함으로써 비효율적인 설계가 있는 부분을 살펴보았고, 또한 Undo 기능 구현에 있어서 전체 게임의 진행을 담당하는 SudokuGame 클래스가 해당 기능의 구현도 같이 담당하고 있어서, 해당 클래스가 너무 비대해지는 문제점을 볼 수 있었다.

안드로이드 기반의 어플리케이션 제작에서와 같이 swing의 undo 패키지를 사용할 수 없는 경우에 있어서, Undo/Redo 기능을 구현하기 위하여 GoF 패턴 중에서 Command, Memento 그리고 Observer 패턴을 유기적으로 사용하여 효율적으로 클래스를 분산시키는 방법을 보였으며, 해당 설계에 바탕을 둔 기능을 구현하여 Undo와 Redo 기능이 정상적으로 작동하는 것을 보였다.

논문에서 적용된 Undo/Redo 기능의 구현 예는 바

둑이나 장기와 같은 게임 외에도 그래픽 에디터 기능이 필요한 어플리케이션의 설계에 도움을 줄 수 있다.

REFERENCE

[1] Find a Sudoku Quiz at Google Play, https://play.google.com/store/search?q=sudoku=apps (accessed Jun., 9, 2016).

[2] Open Sudoku, https://github.com/ogarcia/opensudoku (accessed Jun., 2, 2016).

[3] J.S. Kim and T.S. Kim, "Design of Network-based Game using the GoF Design Patterns," *Journal of Korea Multimedia Society*, Vol. 9, No. 6, pp. 742-749, 2006.

[4] D.S. Park, H.J. Shin, and H.K. Kim, "A Study on the Component based Service Oriented Architecture through Facade Design Pattern," *Proceedings of the Korea Multimedia Society Conference*, pp. 583-589, 2003.

[5] Undoable Drawing Panel, http://www.java2s.com/Code/Java/Swing-JFC/UndoableDrawingPanel2.htm (accessed Mar., 8, 2016).

[6] Working With Design Patterns: Memento, http://www.developer.com/design/article.php/3720566/Working-With-Design-Patterns-Memento.htm (accessed Mar., 8, 2016).

[7] Memento Pattern, http://www.oodesign.com/memento-pattern.html (accessed Apr., 8, 2016).

[8] E. Gamma, R. Johnson, R. Johnson, and H. Vissides, *Design Patterns(Elements of Reusable Object-Oriented Software)*, Addison-Wesley Publishing Company, Boston, 1995.

김 태 석

1981년 경북대학교 전자공학과 공학사 졸업
1989년 일본 KEIO대학 이공학부 계산기과학전공 공학석사
1993년 일본 KEIO대학 이공학부 계산기과학전공(공학박사)
1993년 일본 국제전신전화연구소(KDD) 기술고문
1993년 일본 KEIO대학 이공학부 객원연구원
1994년~현재 동의대학교 컴퓨터소프트웨어공학과 교수
관심분야 : 정보시스템, 기계번역, 인터넷비즈니스

김 종 수

1992년 부경대학교 냉동공학 전공 공학사 졸업
2003년 부산외국어대학교 컴퓨터 공학전공 공학석사
2006년 동의대학교 컴퓨터소프트웨어공학전공 공학박사
2012년 동의대학교 산업기술 개발연구소 P.D 연구원
2013년~현재 한국승강기대학 승강기공학부 교수
관심분야 : 소프트웨어 설계, 게임