

PAC 패턴

Presentation
Abstraction
Control Pattern

아꿈사 <http://cafe.naver.com/architect1>
TTF <http://www.npteam.net>

상호 작용 시스템의 2가지 패턴



M odel		P resentation
V iew		A bstraction
C ontroller		C ontrol

PAC 패턴의 3가지 컴포넌트

Presentation Component

- 표현

Abstraction Component

- 데이터 추상

Control Component

- 사용자 입력

계층구조를 이룬 에이전트들이 서로 협력을 이루어 상호작용 소프트웨어 시스템의 구조를 형성한다.

PAC Pattern - Context

■ 정황(Context)

- 에이전트들이 협력하는 상호작용 애플리케이션을 개발한다.

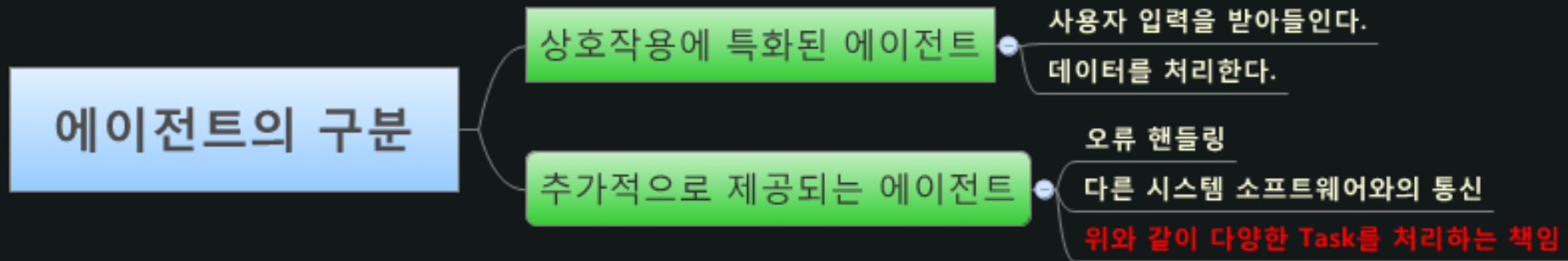


이미지 출처 : <http://imgmovie.naver.com/mdi/mi/0129/A2954-18.jpg>

PAC Pattern - Problem

■ 문제(Problem)

- 상호작용 시스템은 협력하는 에이전트들의 집합

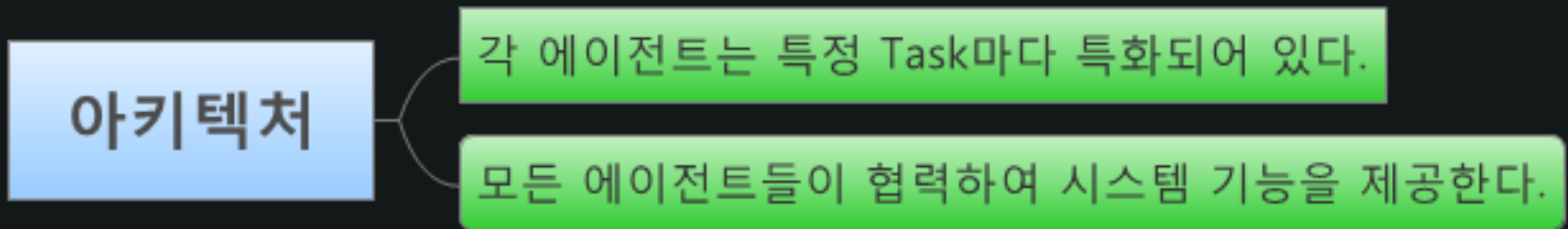


각 에이전트들은 여왕벌과 일벌처럼
분업화와 협력을 통해 일을 처리한다.

PAC Pattern - Problem

■ 문제(Problem)

- 협력하는 에이전트들로 구성된 아키텍처



각각의 에이전트마다 역할을 분리하고
각각의 에이전트간 협력을 통해서 시스템
기능을 제공한다.

PAC Pattern - Problem

■ 문제(Problem)

- 수평적, 수직적 분해를 해결해야 한다.

아키텍처

수평적 분해(horizontal decomposition)

수직적 분해(vertical decomposition)



수평적 문제와
수직적 문제를
분리하여 에이전트간
협력을 통해
문제를 단순화 한다.

PAC Pattern – Problem(Force)

■ 문제에 내포된 영향력(Force) - 01

- 에이전트는 자체 상태와 데이터를 저장한다.
- 에이전트마다 하는 일이 다르다.
- 전체 Task를 제공하기 위해 상호 협력한다.
- 이를 위해 에이전트들은
교환, 데이터, 메시지, 이벤트를 위한
단일 메커니즘이 필요하다.

PAC Pattern – Problem(Force)

■ 문제에 내포된 영향력(Force) - 02

- 상호작용 에이전트들은 각각 자체 사용자 인터페이스를 제공한다.
- 스프레드시트에 데이터를 입력하는 작업
→ 키보드 인터페이스
- 그래픽 객체를 조작하는 작업
→ 포인팅 디바이스(위치 지정 도구)

PAC Pattern – Problem(Force)

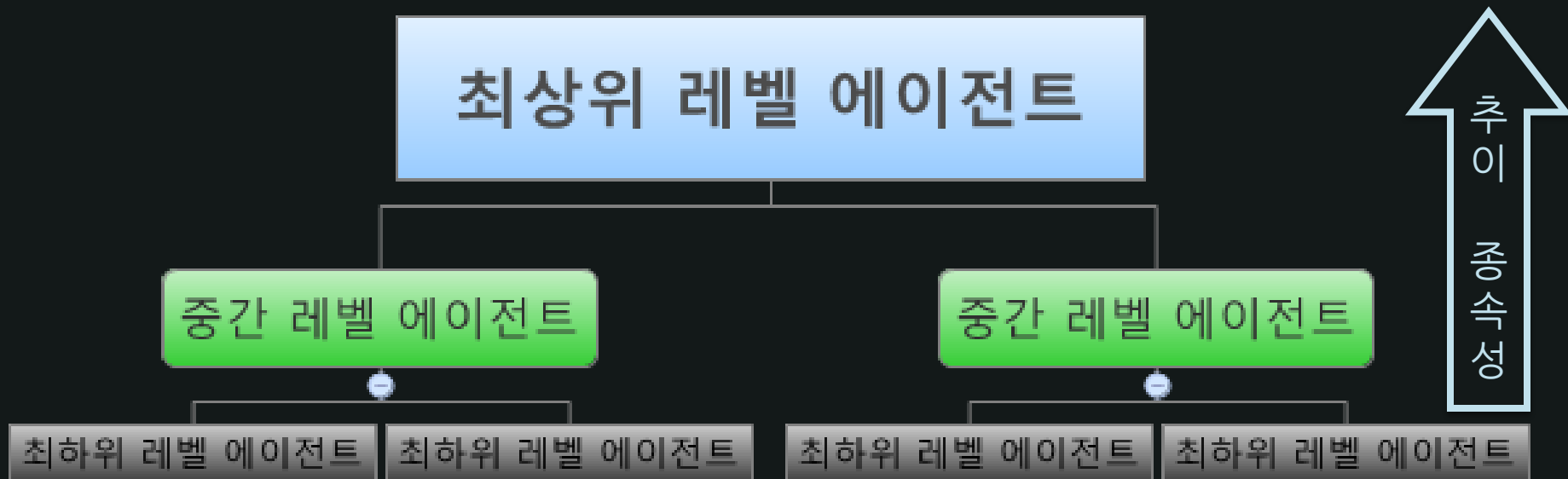
■ 문제에 내포된 영향력(Force) - 03

- 시스템에서 프레젠테이션 측면은 특히 **변경**이 잦은 편이다.
- 각 에이전트의 변경 및 추가가 이루어지더라도 전체 시스템에 영향을 미쳐서는 안 된다.

PAC Pattern – Solution

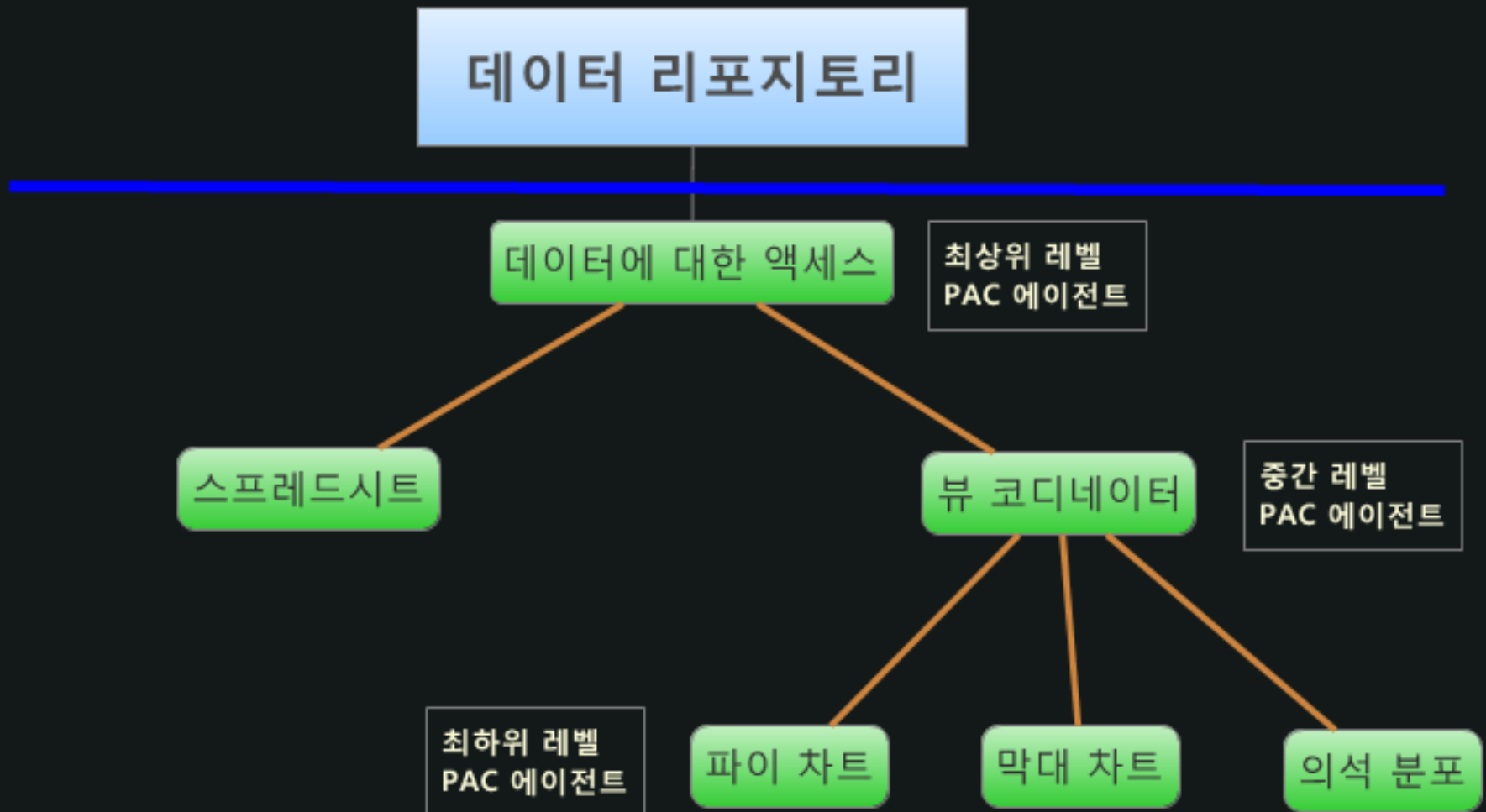
■ 해법(Solution)

- PAC 에이전트들은 트리 형태 계층구조로 구성하여 상호작용 애플리케이션을 구축한다.



PAC Pattern – Solution

■ 해법(Solution)



PAC Pattern – Solution

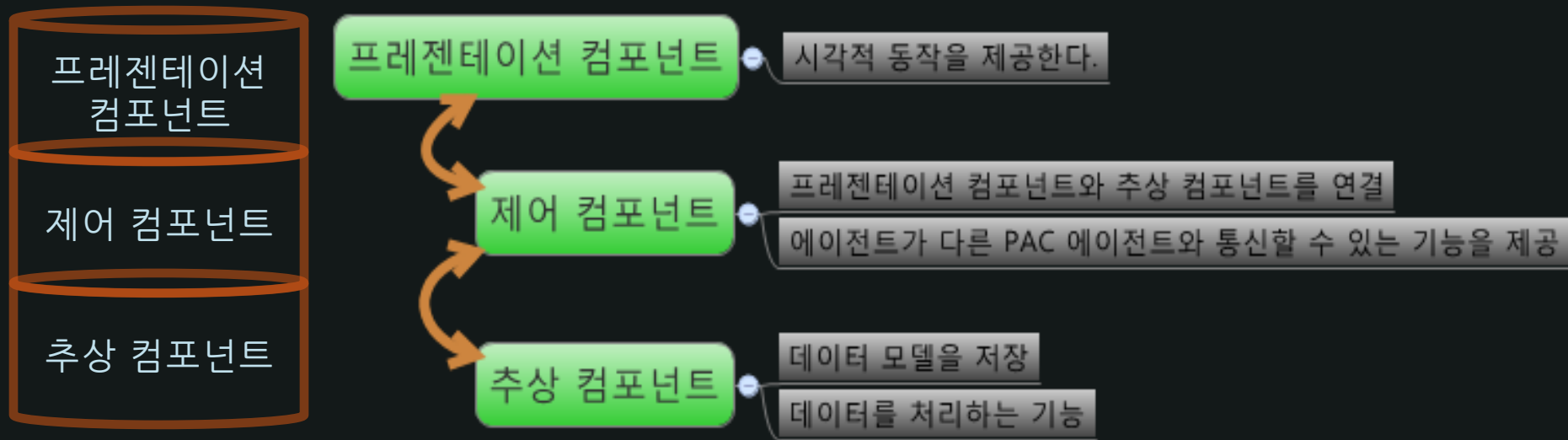
■ 해법(Solution)

- 최상위 레벨 PAC 에이전트
→ 시스템의 핵심 기능을 제공한다.
- 중간 레벨 PAC 에이전트
→ 하위 레벨 에이전트를 조합하는 역할
→ 하위 레벨 에이전트들 사이의 관계 연결
- 최하위 레벨 PAC 에이전트
→ 사용자가 다루는 의미적인 개념을 표현한다.

PAC Pattern – Solution

■ 해법(Solution)

- 모든 에이전트는 애플리케이션 기능의 특정 측면을 담당한다.



PAC Pattern – Structure

■ 최상위 레벨 PAC 에이전트 - 01

- 추상 컴포넌트에서 하는 일
 - 소프트웨어 전역 데이터 제공 및 관리
- 프레젠테이션 컴포넌트에서 하는 일
 - 맡고 있는 책임이 거의 없다.
- 제어 컴포넌트에서 하는 일
 - 하위 레벨 에이전트들이 최상위 레벨의 서비스를 사용할 수 있도록 해준다.
 - 계층 구조간 연결 정보를 관리한다.
 - 사용자와 시스템의 상호작용 정보 저장

PAC Pattern – Structure

■ 최하위 레벨 PAC 에이전트 - 02

- 프레젠테이션 컴포넌트에서 하는 일
 - 뷰와 함수에 대한 액세스를 제공한다.
- 추상 컴포넌트에서 하는 일
 - 다른 PAC와 의존성이 없는 데이터 관리
- 제어 컴포넌트에서 하는 일
 - 인터페이스와 데이터 사이의 어댑터 역할
 - 이벤트와 데이터 교환을 위해
 - 최상위 레벨 에이전트와 통신한다.
 - 수신 데이터를 자체 추상 컴포넌트에 전달

PAC Pattern – Structure

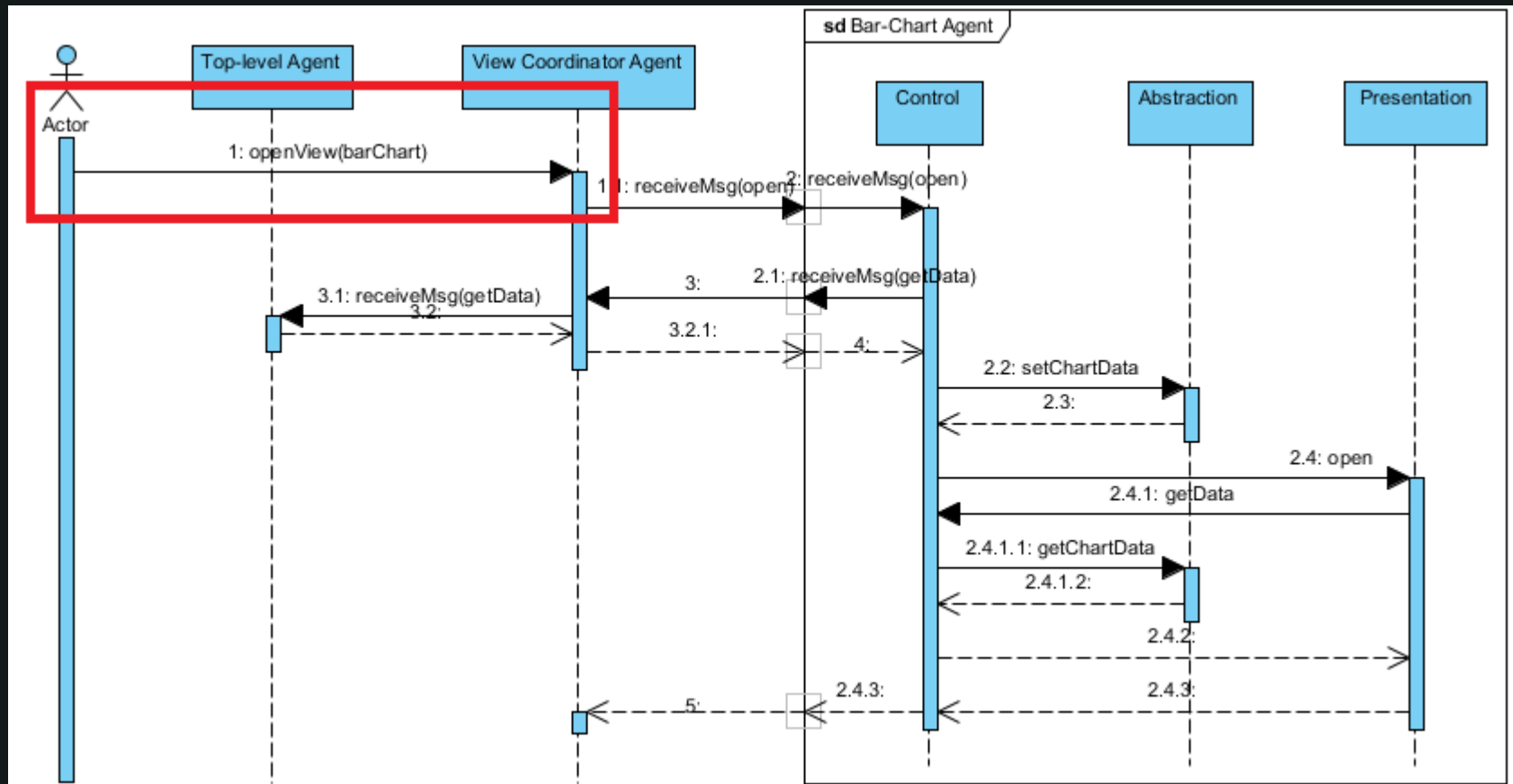
■ 중간 레벨 PAC 에이전트 - 03

- 복합과 조정 두 가지 다른 역할을 수행한다.
 - 독립적인 객체를 하나의 복합 객체로 묶는다.
 - 하위 에이전트들 간의 일관성을 유지한다.
- 추상 컴포넌트
 - 데이터 저장
- 프레젠테이션 컴포넌트
 - 자체 사용자 인터페이스 구현
- 제어 컴포넌트
 - 최상위, 최하위 레벨 PAC와 동일한 책임

PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

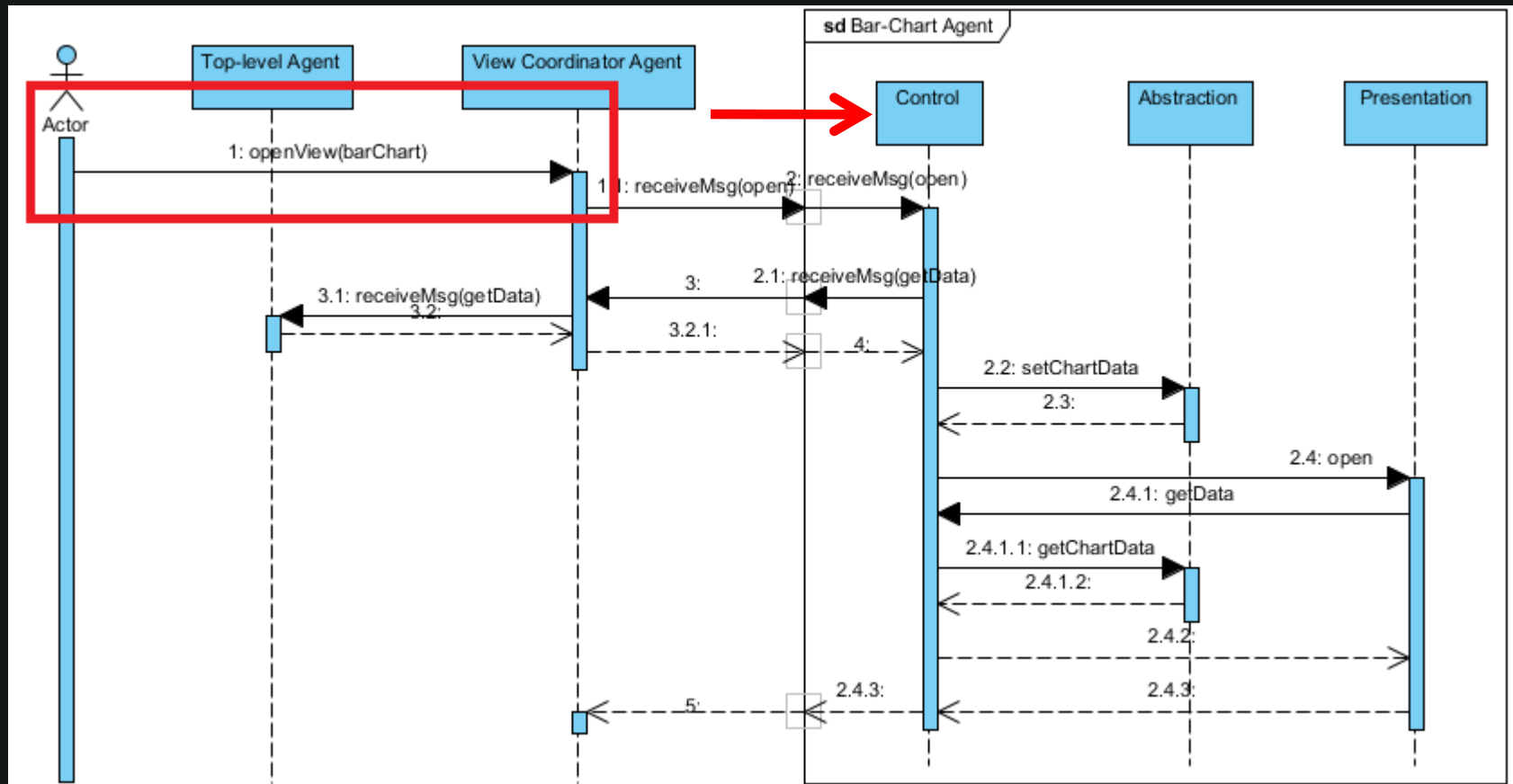
- 사용자가 뷰 코디네이터 에이전트의 프리젠테이션 컴포넌트에 새 막대 차트를 열도록 요청한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

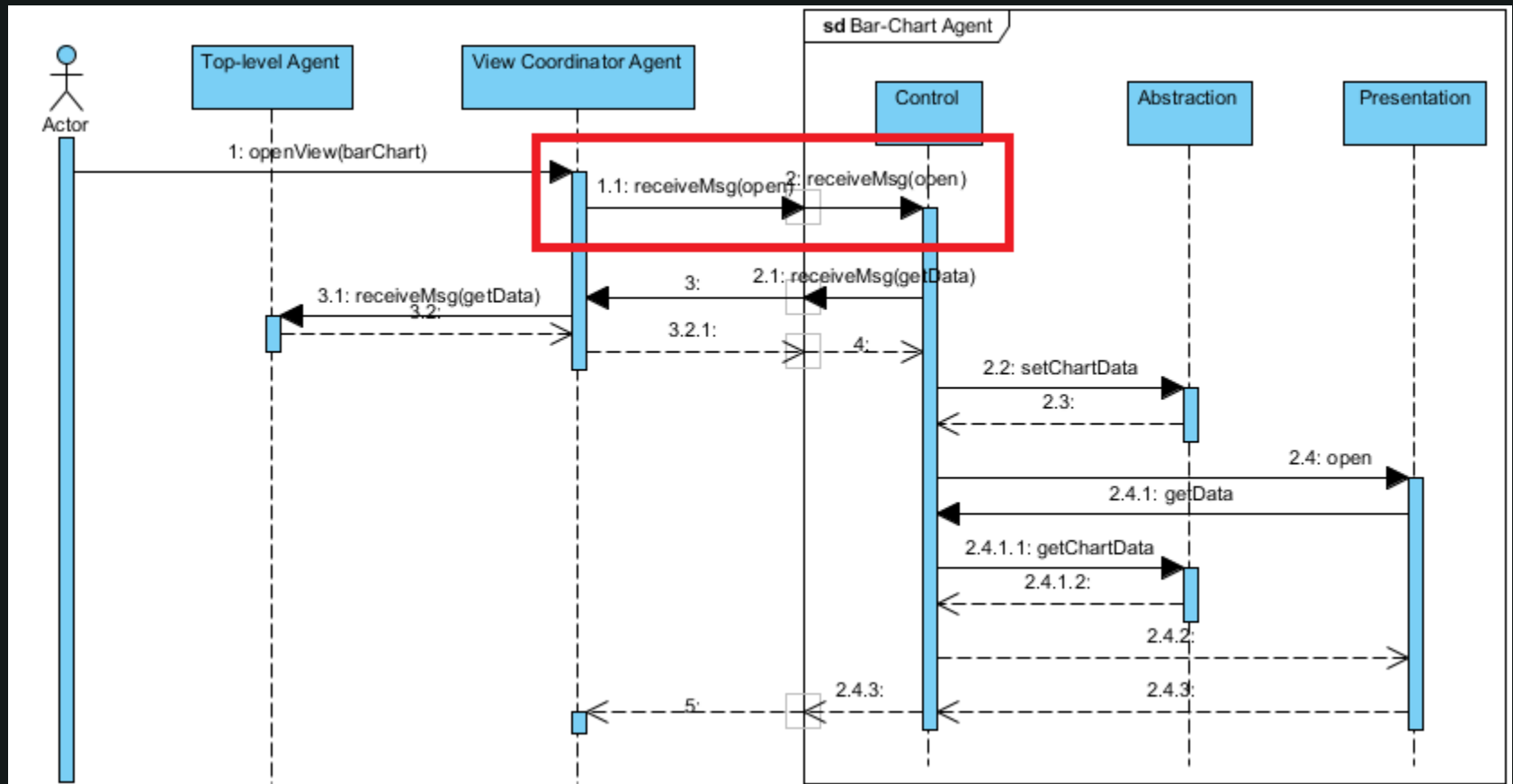
- 뷰 코디네이터 에이전트의 제어 컴포넌트가 막대 차트 에이전트를 인스턴스로 생성한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

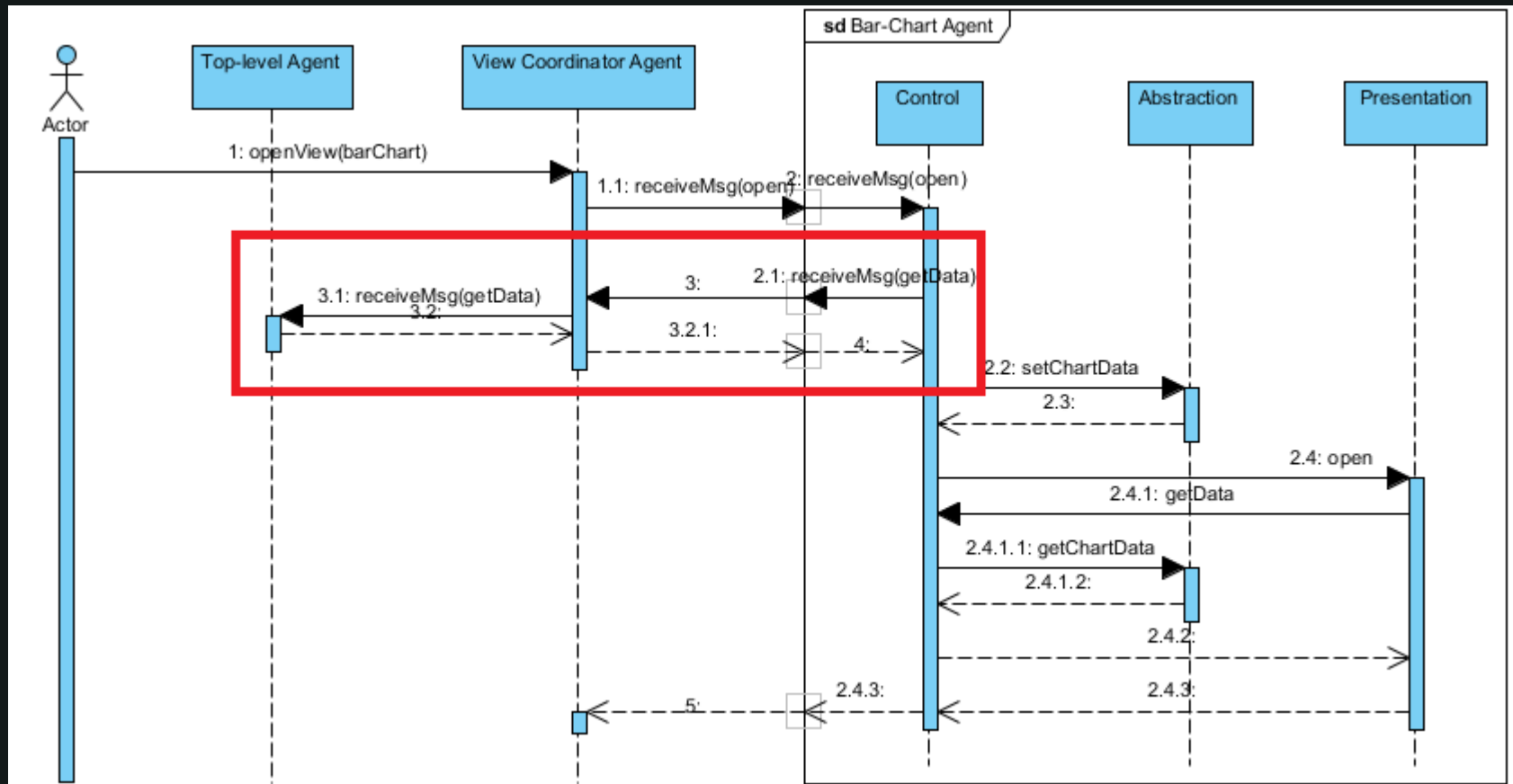
- 뷰 코디네이터 에이전트는 open 이벤트를 새 막대 차트 에이전트의 제어 컴포넌트에 보낸다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

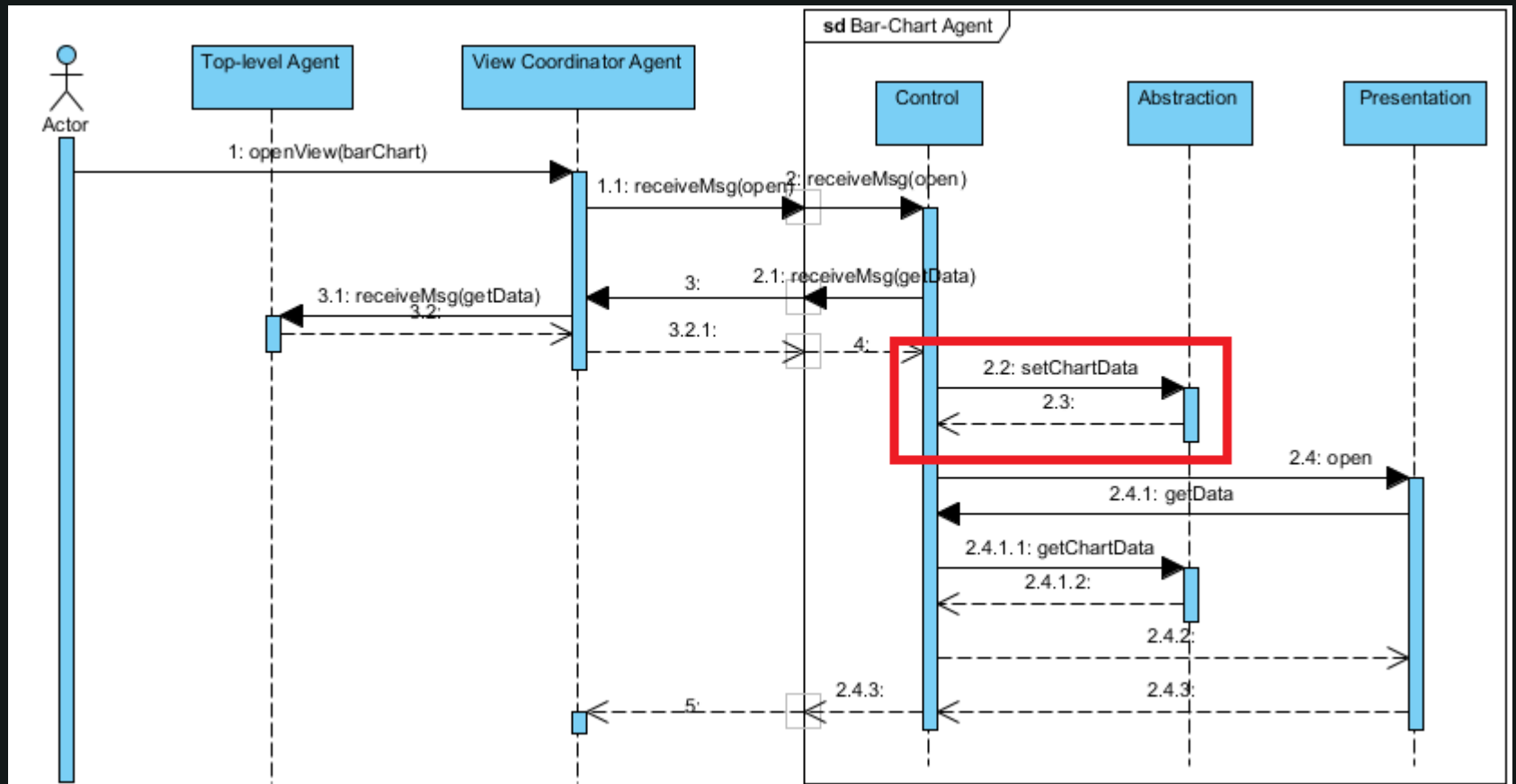
- 막대 차트 에이전트의 제어 컴포넌트는 뷰 코디네이터 에이전트를 통해 데이터를 가져온다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

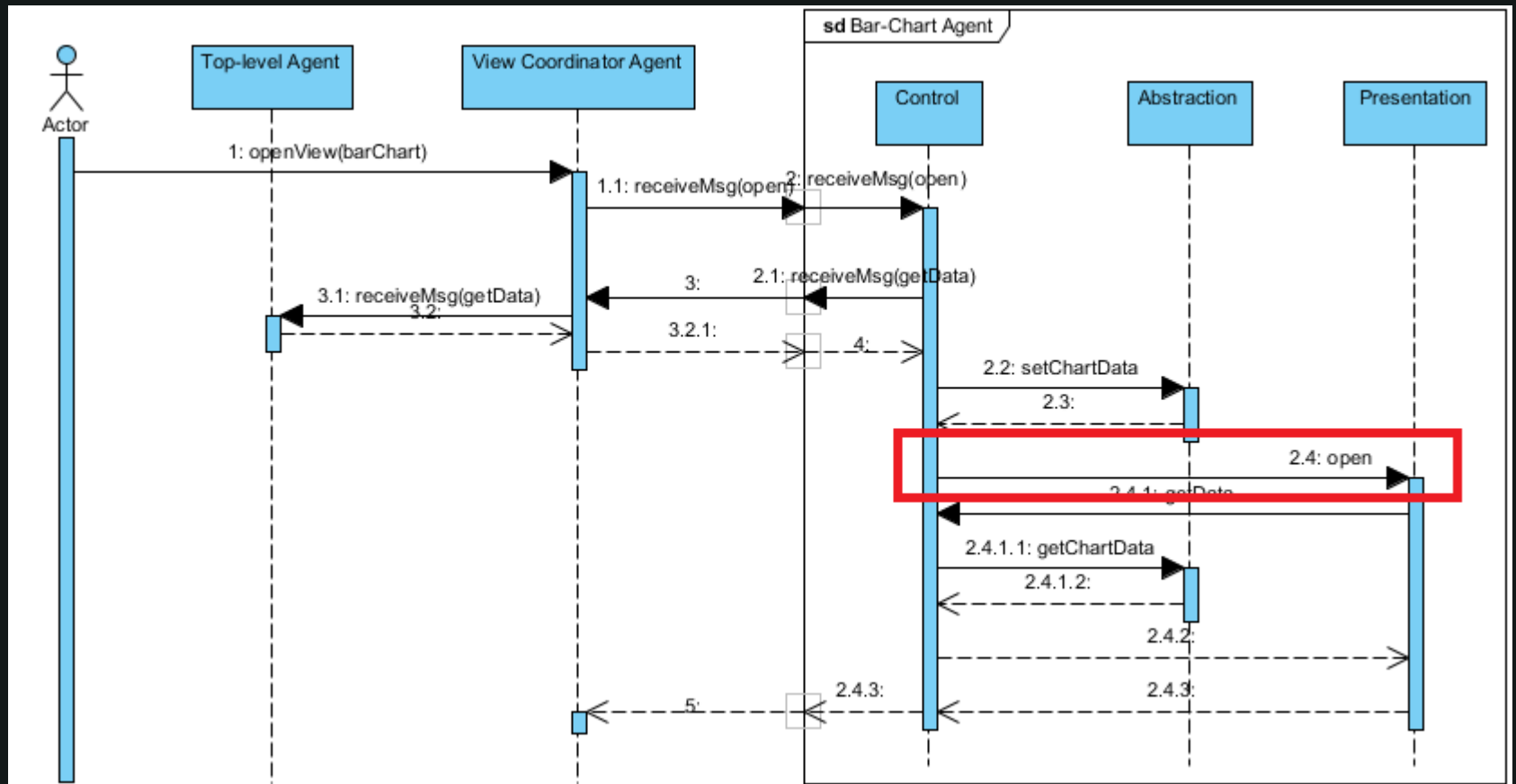
- 막대 차트 에이전트에 반환된 데이터는 자체 추상 컴포넌트에 저장된다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

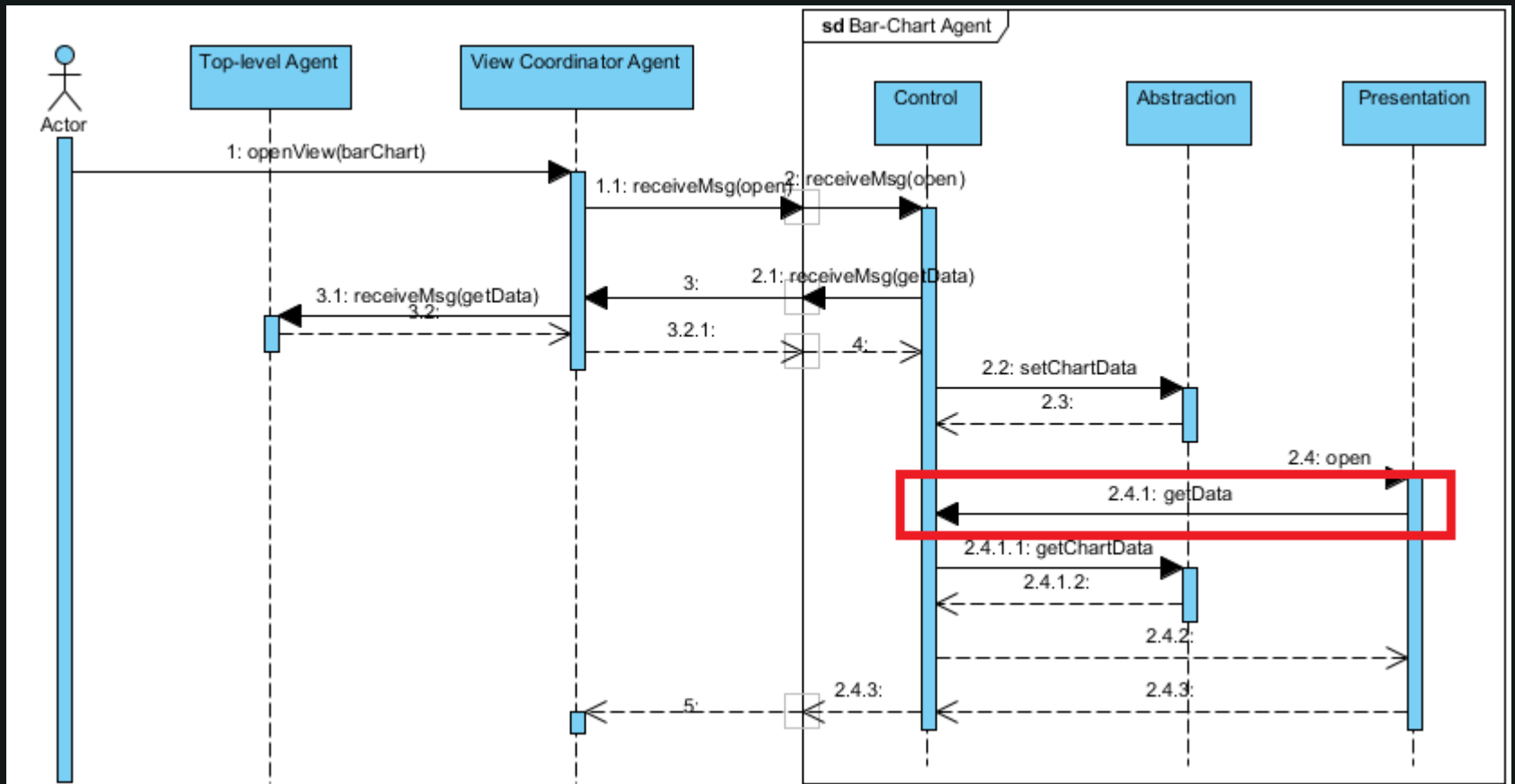
- 제어 컴포넌트는 막대 차트를 디스플레이 하기 위해 프레젠테이션 컴포넌트를 호출한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

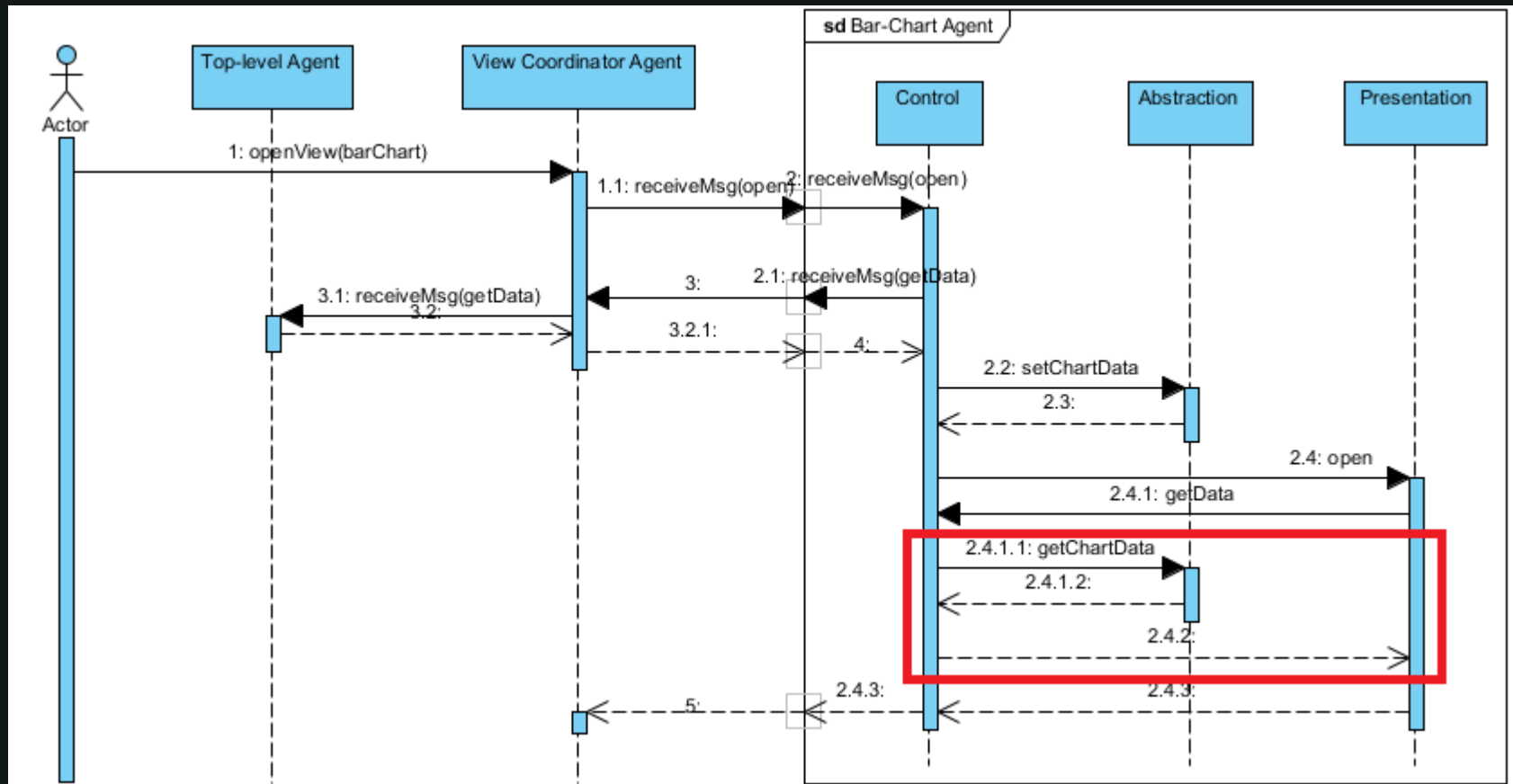
- 프레젠테이션 컴포넌트는 화면에 새 창을 생성하고, 제어 컴포넌트에 데이터를 요청한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

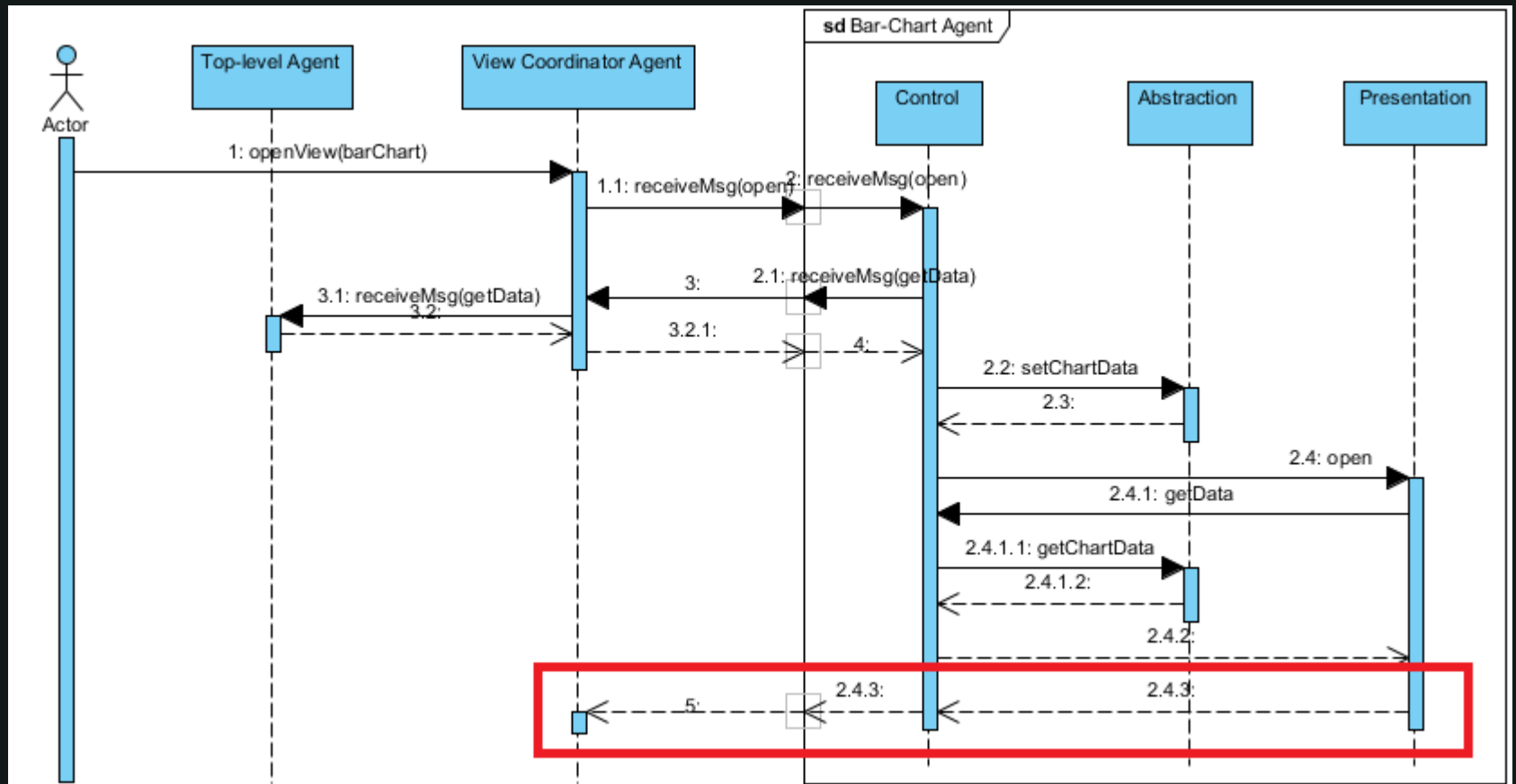
- 제어 컴포넌트는 데이터를 추상 컴포넌트로부터 가져와 프레젠테이션 컴포넌트에 보낸다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오1)

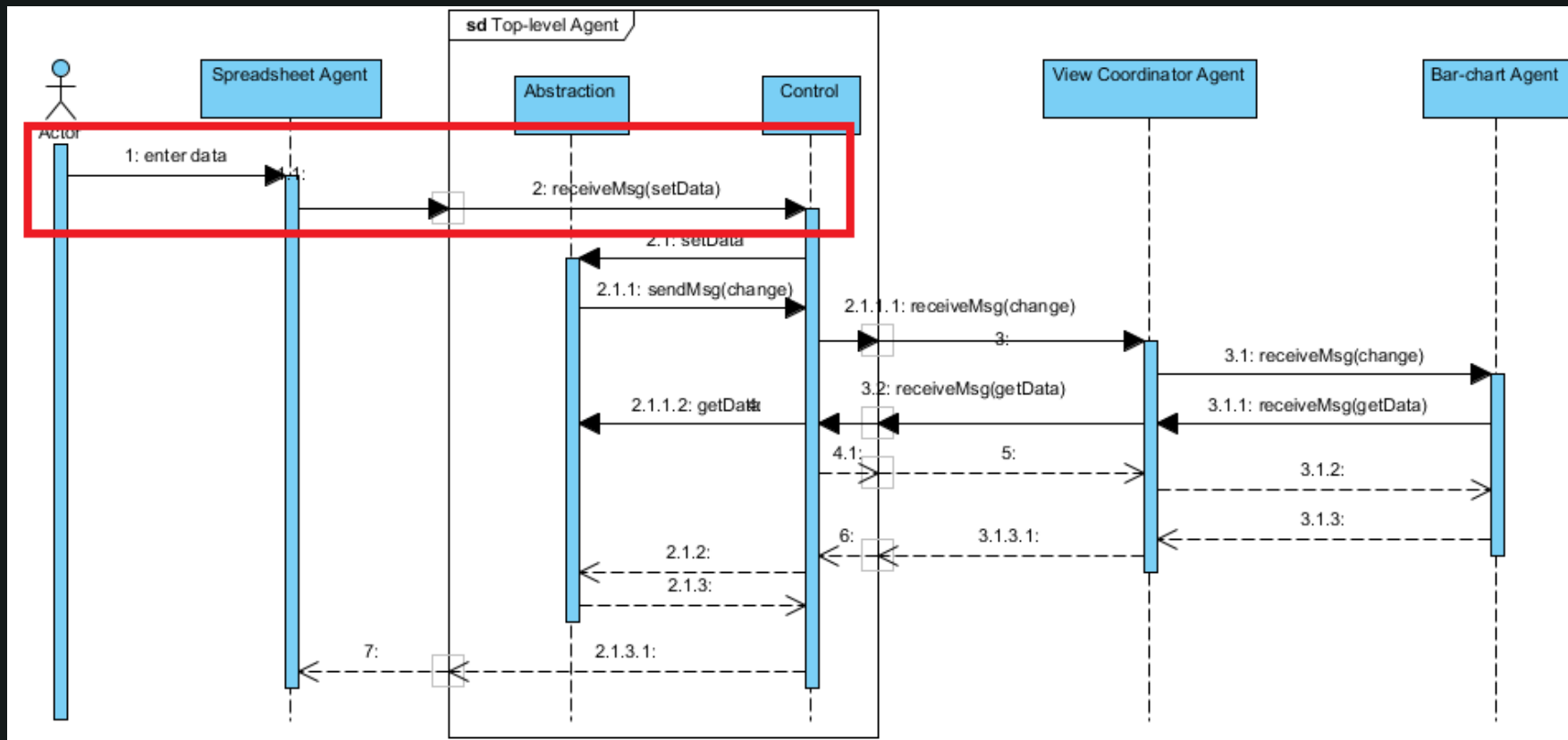
- 최종적으로 프레젠테이션 컴포넌트는 받은 데이터를 새 창에 표시한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오2)

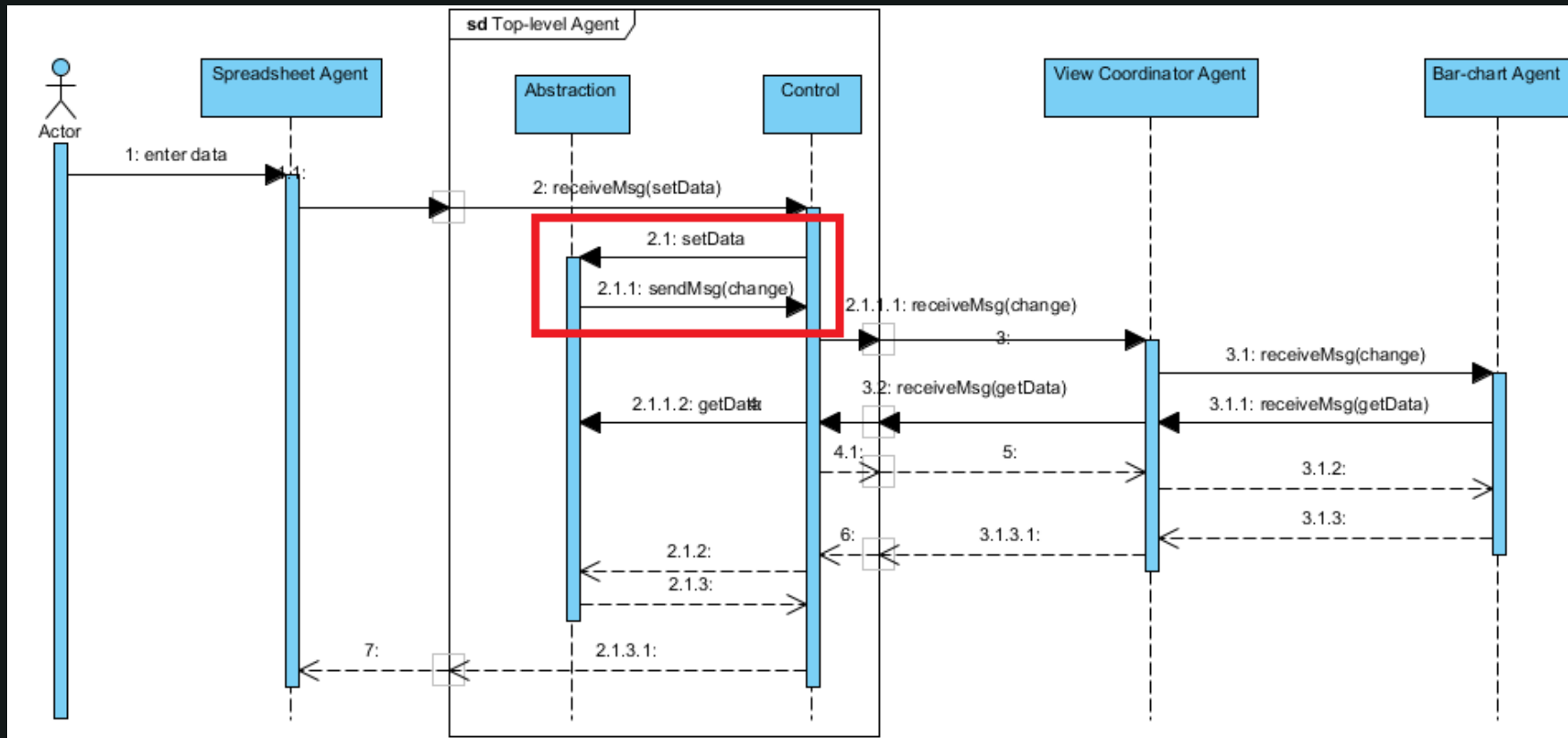
- 사용자가 스프레드시트에 새 데이터를 입력한다.
- 제어 컴포넌트는 데이터를 최상위 레벨 PAC로 전달한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오2)

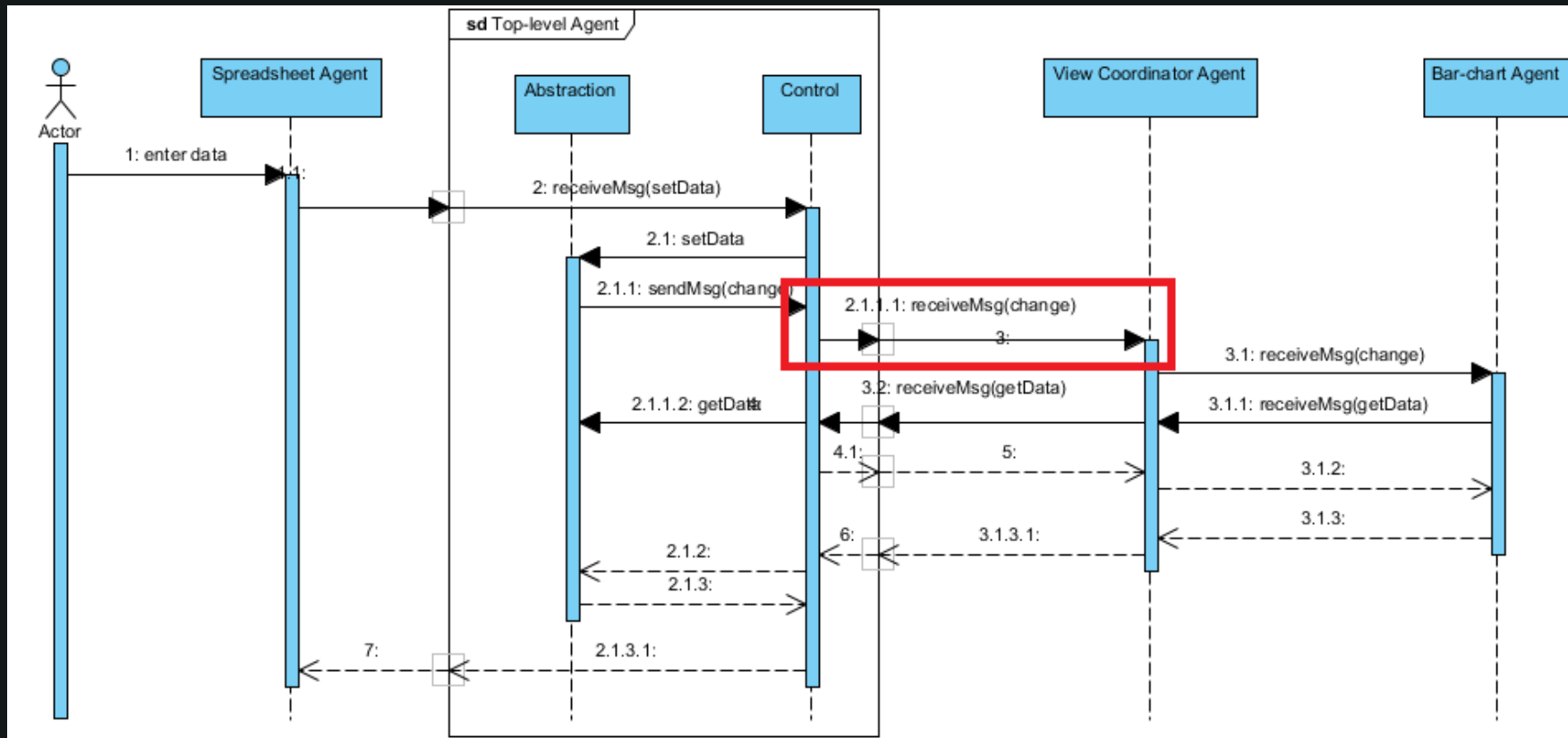
- 제어 컴포넌트는 데이터를 추상 컴포넌트에 알려준다.
- 추상 컴포넌트는 에이전트들을 업데이트 하도록 요청한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오2)

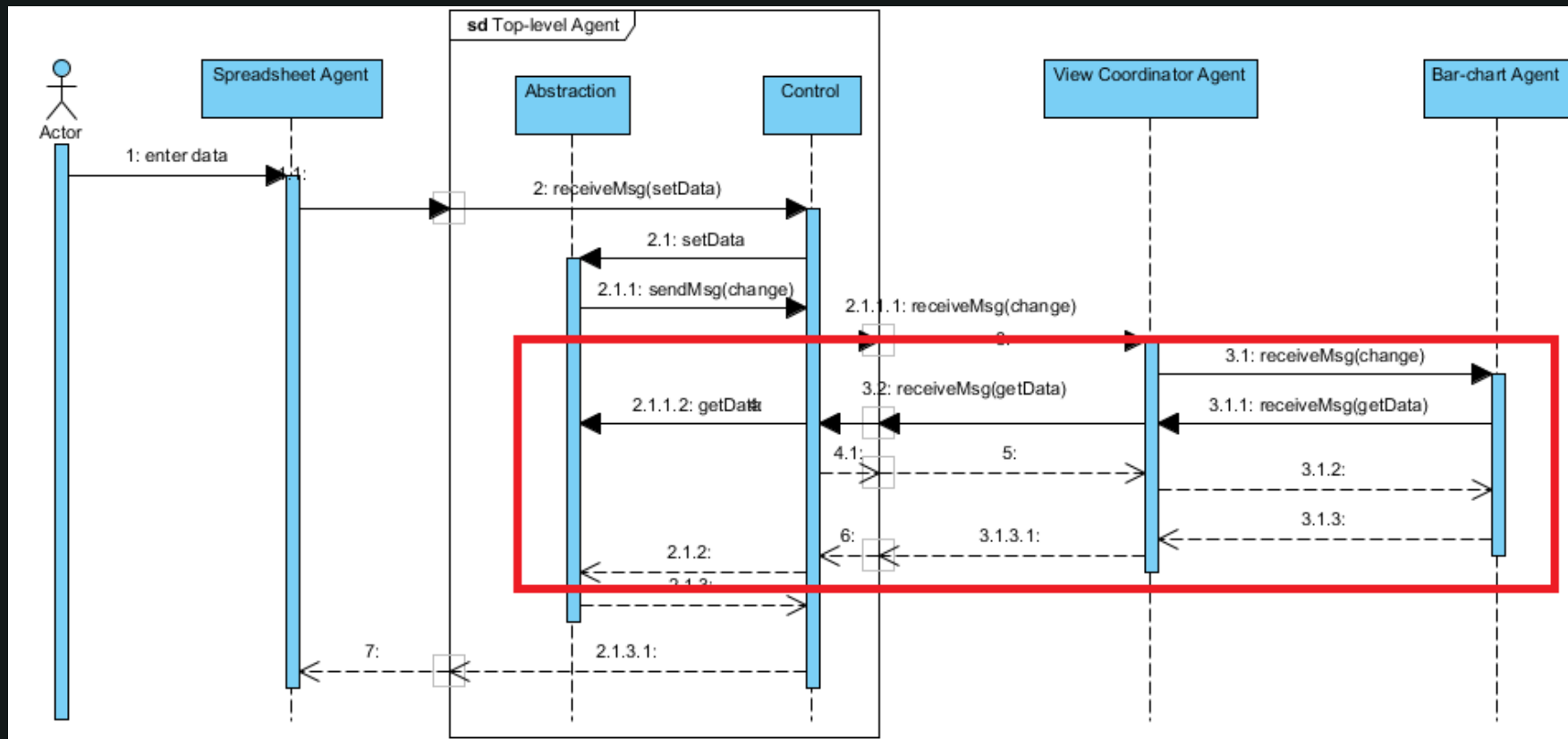
- 제어 컴포넌트는 뷰 코디네이터 에이전트에 통지를 보낸다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오2)

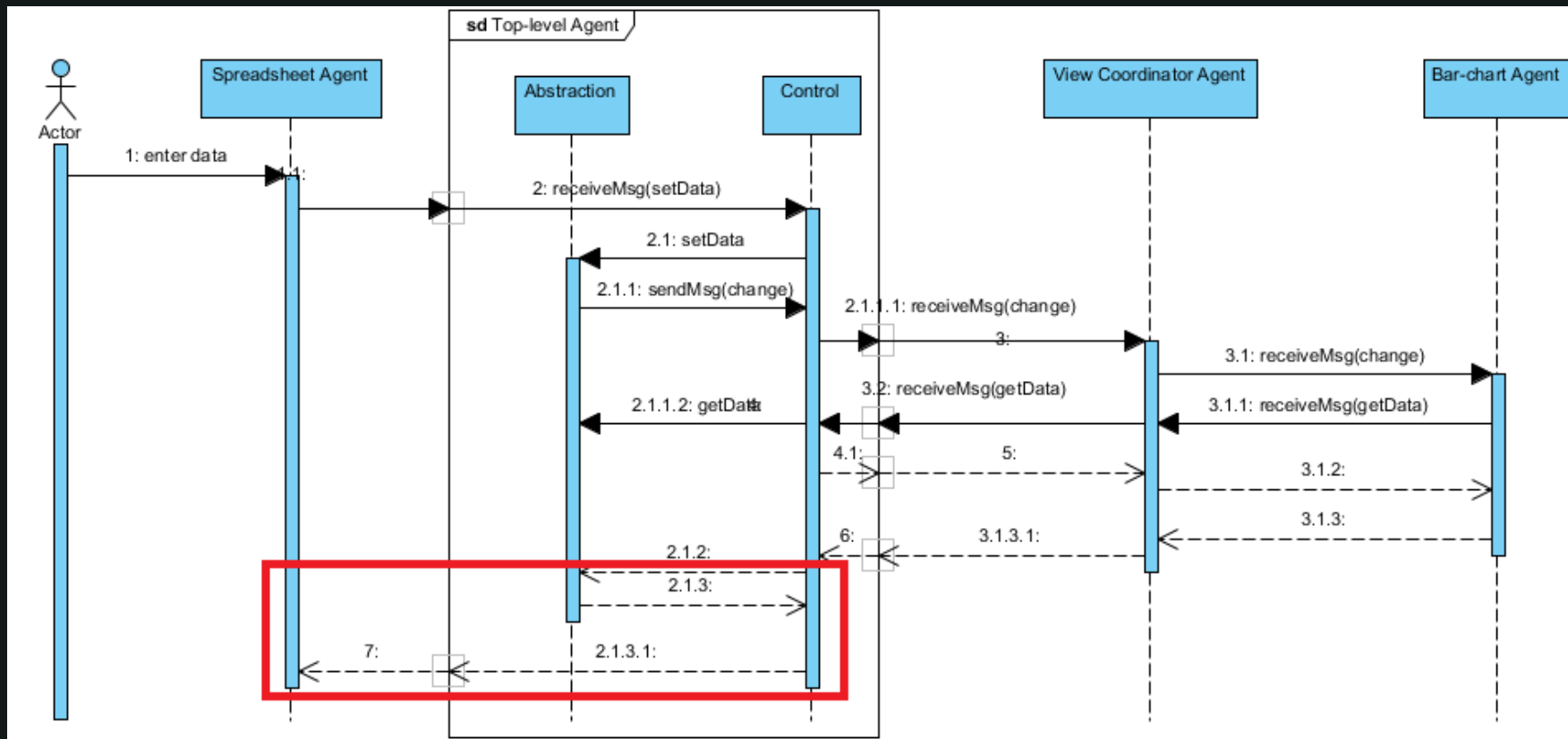
- 뷰 코디네이터 에이전트의 제어 컴포넌트는 조정 책임을 맡은 모든 뷰 PAC에게 변경 통지를 전달한다.



PAC Pattern – Dynamic

■ PAC 동작 묘사(시나리오2)

- 자체 데이터를 업데이트하고 이미지를 갱신한다.



PAC Pattern – Implementation

■ 구현(Implementation)

- **1단계** 애플리케이션의 모델을 정의한다.
 - 시스템은 어떤 서비스를 제공해야 하는가?
 - 어떤 컴포넌트가 이 서비스들을 수행할 수 있는가?
 - 컴포넌트들 간의 관계는 어떠한가?
 - 컴포넌트는 어떤 데이터를 처리하는가?
 - 사용자는 어떻게 소프트웨어와 상호작용 하는가?

PAC Pattern – Implementation

■ 구현(Implementation)

- **2단계** PAC 계층을 조직화하기 위해 일반 전략을 정의한다.

→ 첫 번째 : 최하위 공통 조상 규칙

→ 두 번째 : 계층구조(hierarchy)의 심도(depth)

- **3단계** 최상위 레벨 PAC 에이전트를 정의한다.

→ 시스템의 핵심 기능에 해당하는 분석 모델의 부분들을 판별해낸다.

PAC Pattern – Implementation

■ 구현(Implementation)

- **4단계** 최하위 레벨 PAC 에이전트를 정의한다.

→ 시스템에서 가장 작은 독립적 단위의 분석 모델을 컴포넌트로 판별해낸다.

- **5단계** 시스템 서비스들을 제공하는 최하위 레벨 PAC 에이전트들을 정의한다.

→ 주요 목적과 직접 관련되지 않은 부가적인 서비스들을 포함한다.

PAC Pattern – Implementation

■ 구현(Implementation)

- **6단계** 하위 레벨 PAC 에이전트들을 조합하기 위한 중간 레벨 PAC 에이전트들을 정의한다.

→ 사람-컴퓨터 상호작용을 제공하며, 하위 레벨 에이전트를 조작하는 역할을 하는 단계

- **7단계** 하위 레벨 PAC 에이전트들을 상호조정하기 위해 중간 레벨 PAC 에이전트들을 정의한다.

→ 동일한 의미적 개념에 대해 여러 뷰를 제공한다.

PAC Pattern – Implementation

■ 구현(Implementation)

- **8단계** 사람-컴퓨터 상호작용으로부터 핵심 기능을 분리해낸다.

→ 모든 PAC 에이전트마다 프레젠테이션 컴포넌트와 추상 컴포넌트를 도입한다.

- **9단계** 외부 인터페이스를 제공한다.

→ 다른 에이전트들과 협력하기 위해 모든 PAC 에이전트들은 이벤트와 데이터를 보내고 받는다.

PAC Pattern – Implementation

■ 구현(Implementation)

- **10단계** 계층구조(hierarchy)를 함께 연결한다.

- 각각의 PAC 에이전트를 구현해야 최종 PAC 계층구조를 구축할 수 있다.

- 협력 관계에 있는 하위 레벨 PAC 에이전트들을 연결함으로써 모든 PAC 에이전트를 함께 연결시킨다.

PAC Pattern – Variant

■ 변형(Variant)

- 활성 객체로서의 PAC 에이전트

→ 멀티스레딩의 이점을 활용하기 위해,
활성 객체(Active Object)로 구현될 수 있다.
활성 객체 : 자체 제어 스레드 안에 살아있는 객체

- 프로세스로의 PAC 에이전트

→ 서로 다른 프로세스나 원격 머신 간의 PAC 통신을
지원하기 위해 PAC 에이전트간 IPC(프로세스간 통신)을
구현한다.

PAC Pattern – Consequence

■ 결과(Consequence)

- 역할이 명확히 분리된다.
- 가변성과 확장성이 지원된다.
- 멀티태스킹이 지원된다.
- 시스템 복잡성이 증가한다.

PAC Pattern – Consequence

■ 결과(Consequence)

- 제어 컴포넌트가 복잡해진다.
- 효율성이 떨어진다.
- 패턴 적용 자체가 무모한 함정일 수 있다.
- 적용가능성이 떨어진다.

감사합니다.