

P04. *Component and Connector*

20134

Sungwon Kang

4. 컴포넌트와 커넥터(The Grand Dichotomy)

4.1 문제 해결에 도움을 주는 시각으로서 컴포넌트와 커넥터 이분법

4.2 표현력을 높이기 위한 아키텍처 요소들

4.1 문제 해결에 도움을 주는 시각으로서 컴포넌트와 커넥터 이분법



그림 4-1. 분할정복 문제 해결 방법의 개념



그림 4-2. 개념수준의 아키텍처 설계

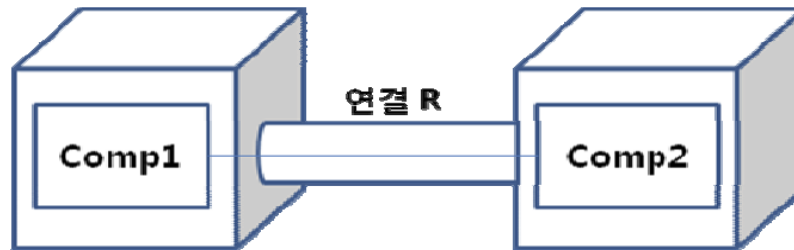
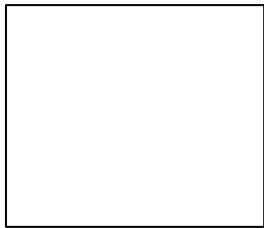


그림 4-3. 아키텍처 설계의 구현

- 컴포넌트
- 컴포넌트간의
관계 (= 커넥터)

4.1 문제 해결에 도움을 주는 시각으로서 컴포넌트와 커넥터 이분법



component



connector

- In this paradigm, component and connector should alternate in a diagram
- 서로 다른 뷰들은 각각 그 뷰에 고유한 컴포넌트와 커넥터로 구성됨.
- How would you express a "complex connector" where a complex connector itself has an internal structure?

Taxonomy of Connectors (in Execution View)

“Connector embodies interaction among Components”

Service	Type
Communication & Coordination	Procedure call, Event, Stream
Conversion	Data Access, Adaptor
Facilitation	Linkage, Distributor, Arbitrator

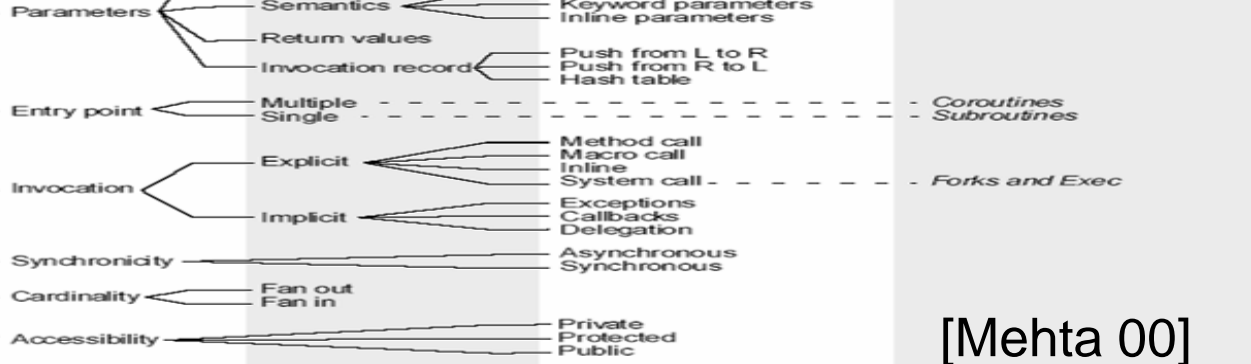
[Mehta 00] Mehta, N.R., Medvidovic, N., Phadke, S., “Towards a taxonomy of software connectors”, Proceedings of the 2000 International Conference on Software Engineering, 2000.

☛ 나아가 다양한 **Complexity**의 **Connector**가 존재 한다!

서비스 구현

Communication
Coordination

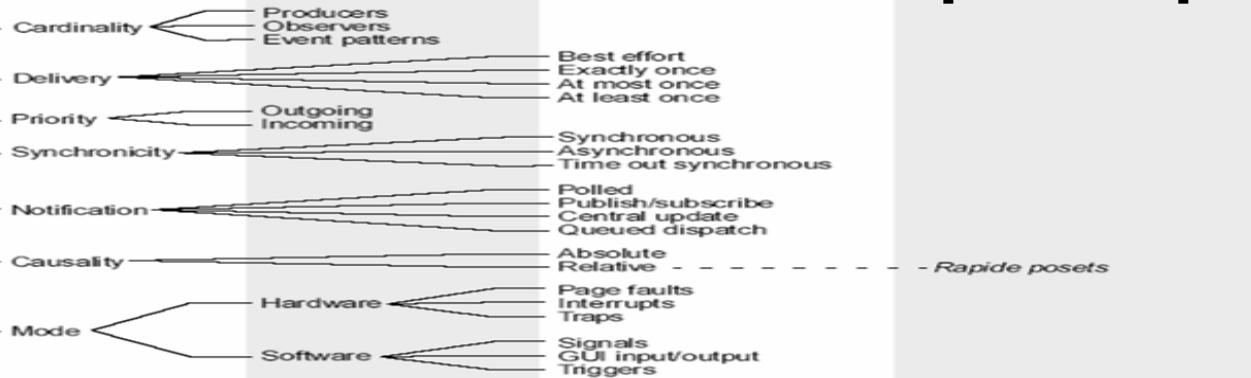
Procedure call



[Mehta 00]

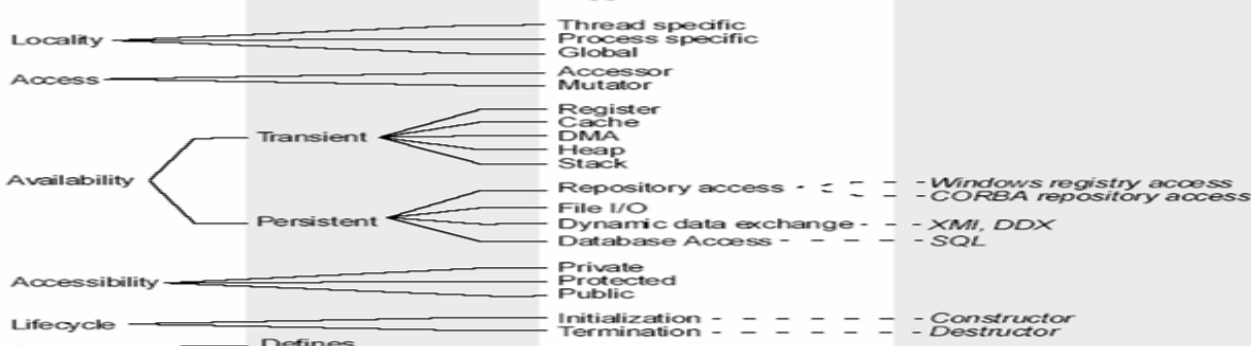
Communication
Coordination

Event



Communication
Conversion

Data Access



Facilitation

Linkage

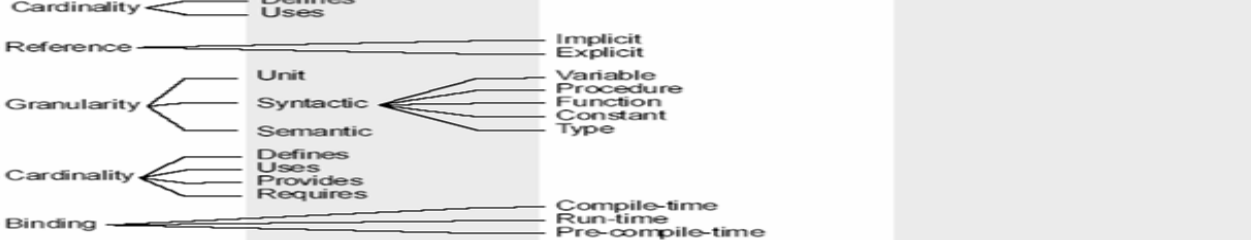
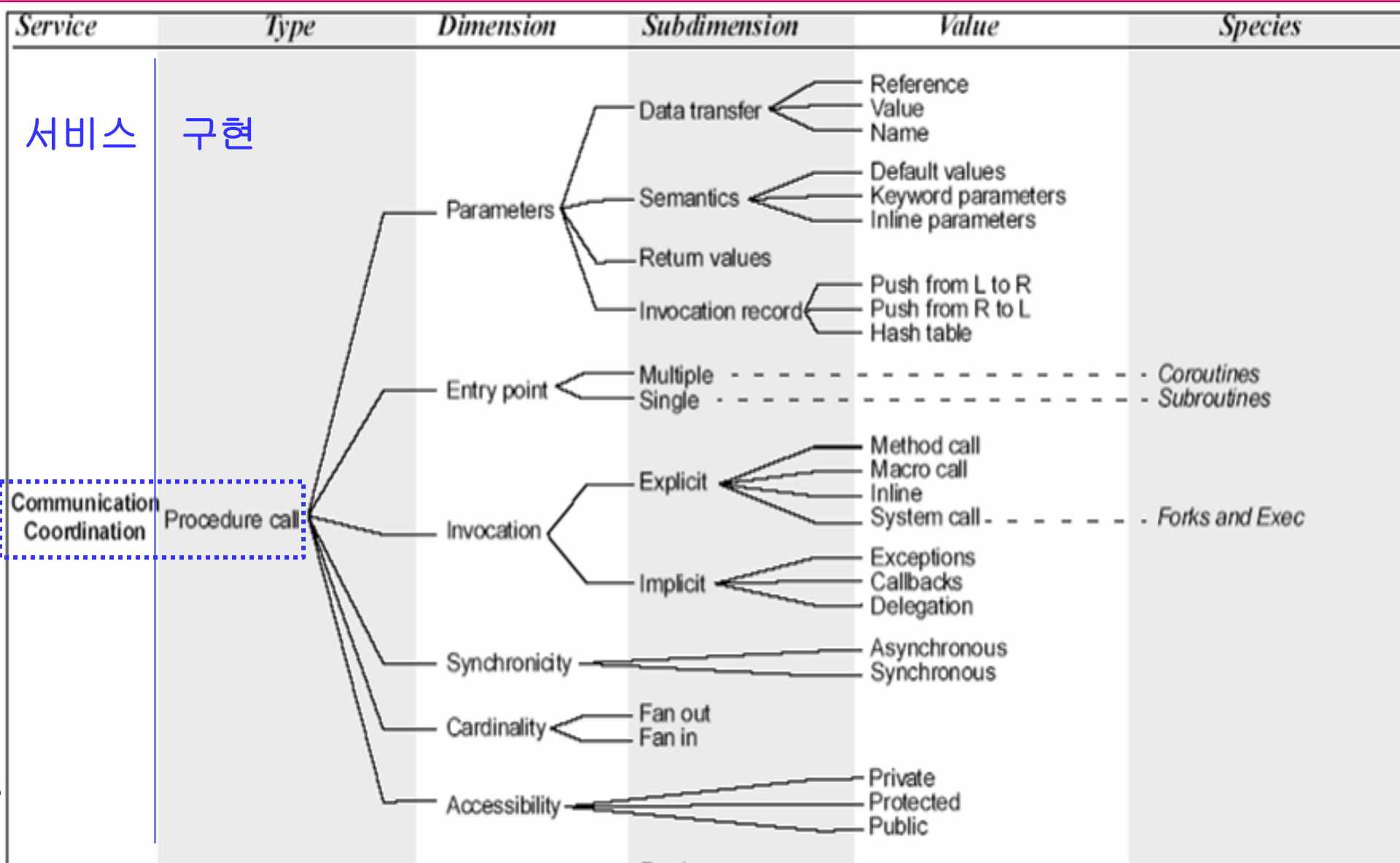


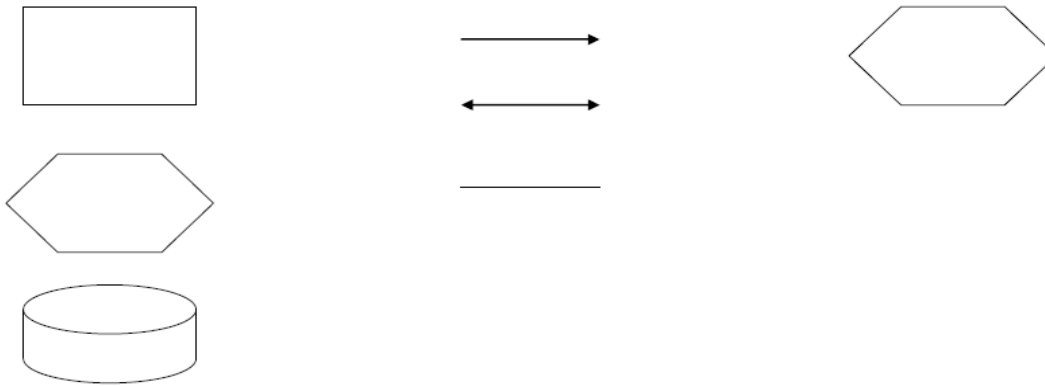
Figure 3. Connector Taxonomy. A small number of connector species is included for illustration purposes.

[Mehta 00]



4.2 표현력을 높이기 위한 아키텍처 요소들

To be flexible and effective, we introduce the following notation.



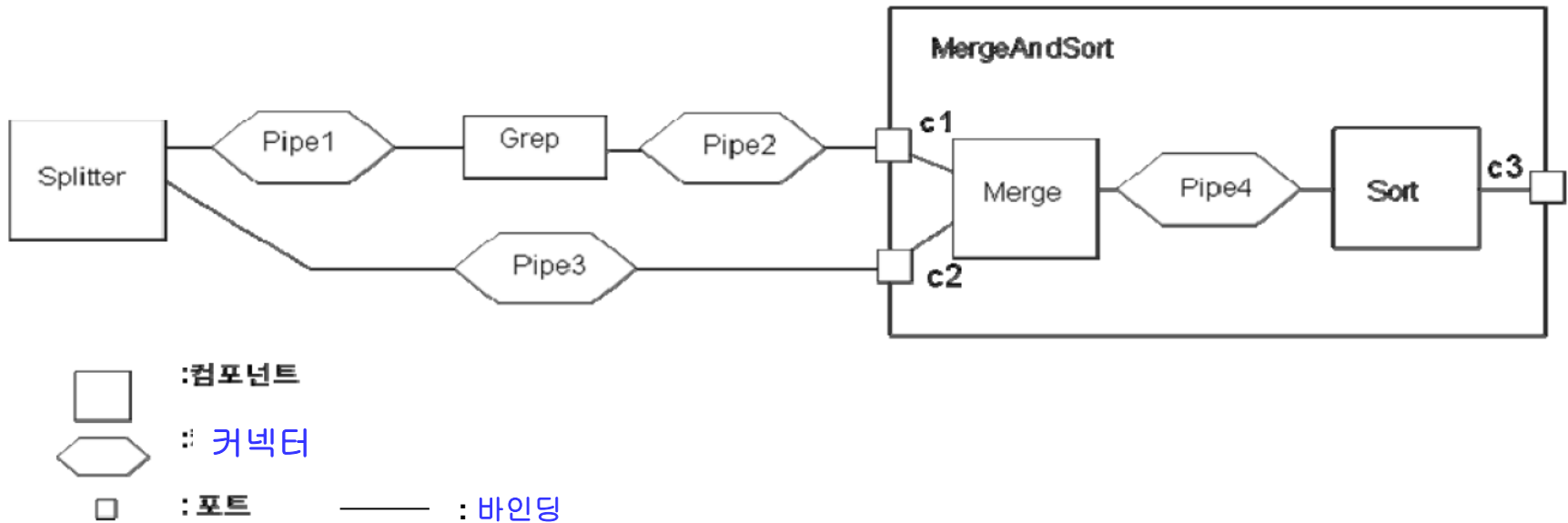
(a) 컴포넌트 기호 (b) 커넥터 기호 (c) 커넥션 컴포넌트

그림 4-4. 아키텍처 요소를 나타내는 기호들



커넥션컴포넌트와 커넥터

커넥터 외에 연결을 위한 요소가 더 필요한가?

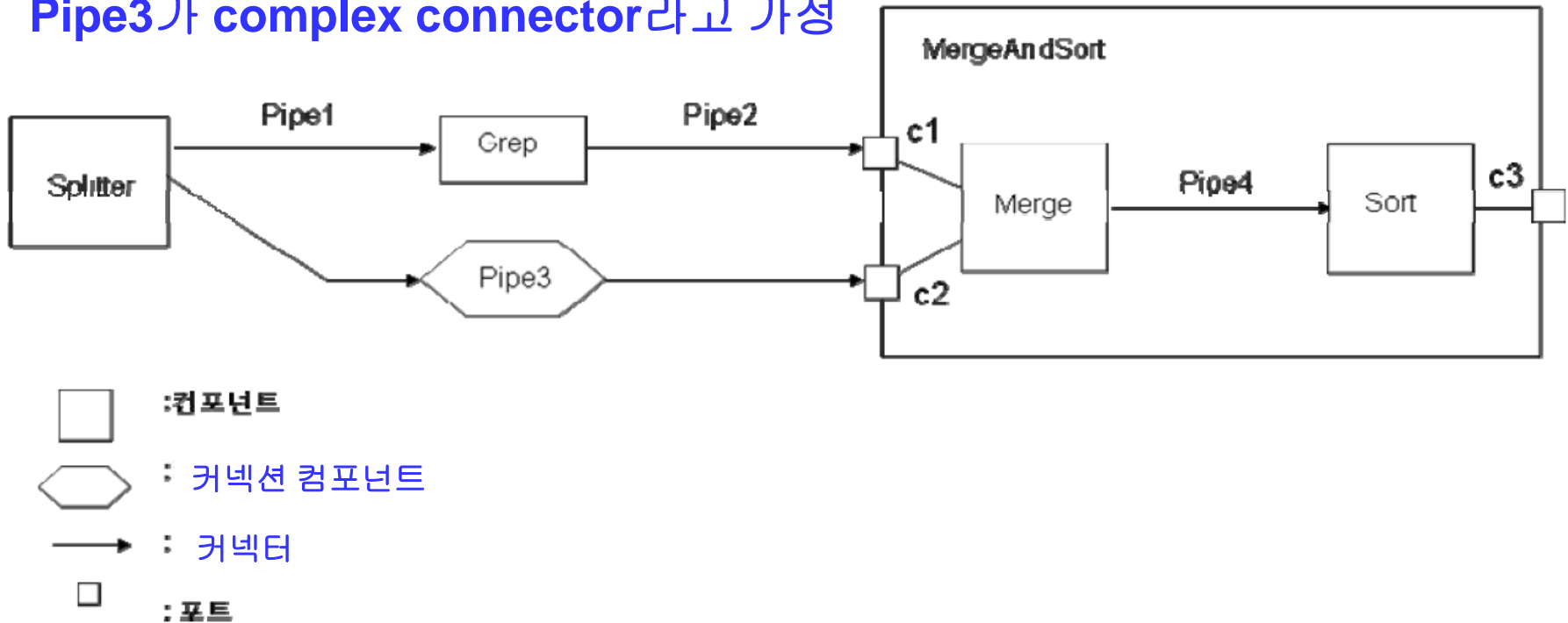


Acme 스타일의 아키텍처 표현

Acme의 용어는 컴포넌트, 커넥터의
2분 법과 일치하지 않아 혼란을 준다.

커넥터 외에 연결을 위한 요소가 더 필요한가?

Pipe3가 complex connector라고 가정



우리의 표기를 사용한 아키텍처 표현 예

- ▶ 컴포넌트와 커넥션 컴포넌트와의 연결은 필요하다.
- ▶ 그러나 컴포넌트들은 커넥터가 연결한다.
- ▶ 커넥션 컴포넌트는 커넥터로 추상될 수 있다.

Examples

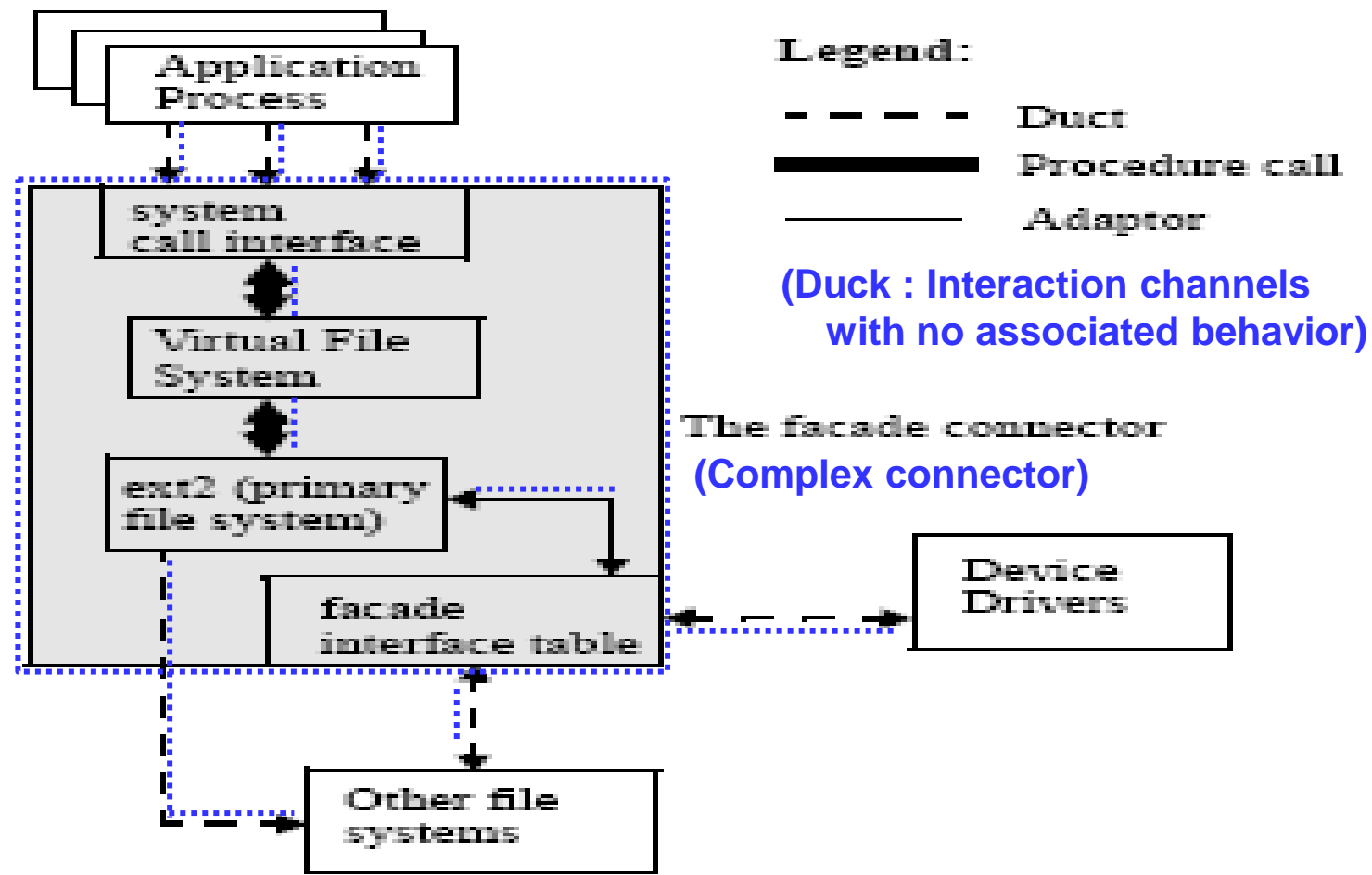
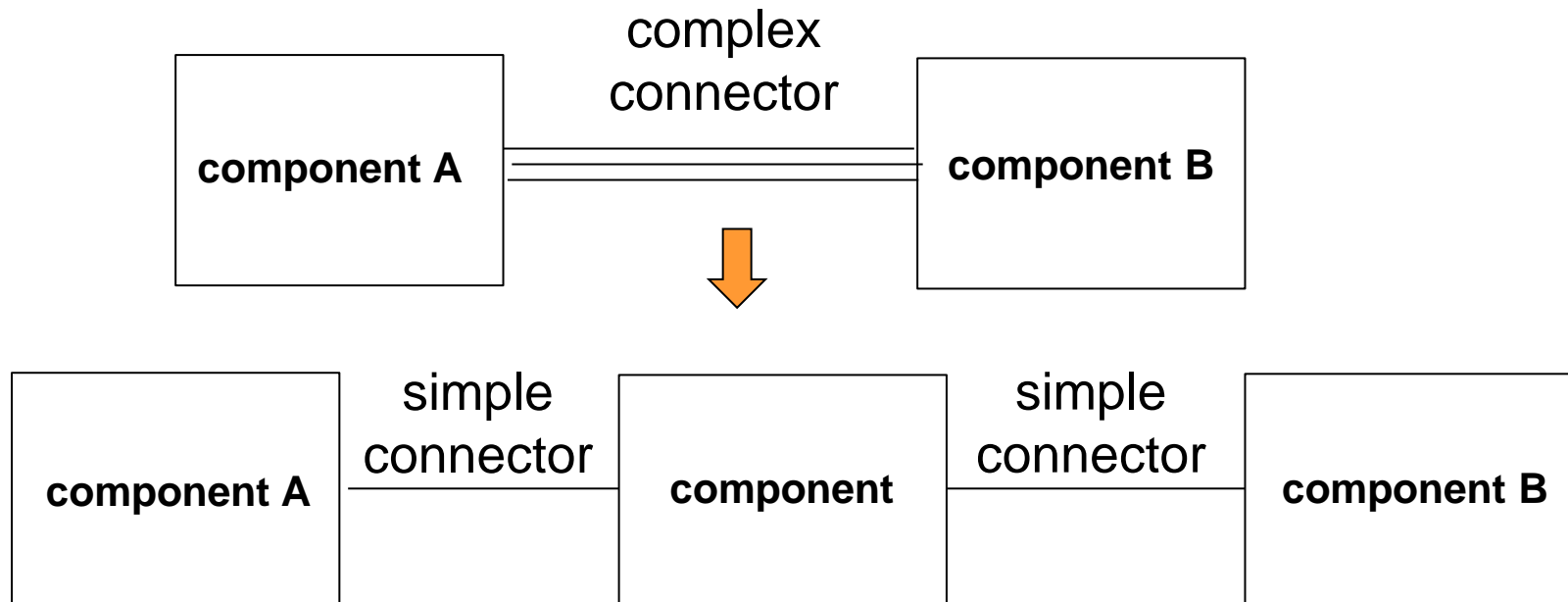


Figure 4. The Linux file facade connector.

Table 1: The file facade connector

Connector Type	Dimension	Value
Arbitrator	Authorization	Access Control Lists
Arbitrator	Isolation	Read/Write
Adaptor	Invocation conversion	Translation

[Clements 11] p135 David Garlan



- A complex connector can be replaced by a component.
- Then is complex connector still needed?
- Garlan's answer: Yes
- In [Clements 02], Garlan almost said that the bottom one cannot replace the top one.

Our view is that the bottom one naturally represents the **top** one and, moreover, it is desirable to **have** a distinct notation for the bottom one (i.e. connection component).

커넥터 외에 연결을 위한 요소가 더 필요한가?

- If we have to use **complex connector** instead of **connection component**, then we would have to now ask what connects components and the complex connector.
- However, with our approach (without special complex connector), we do not need any additional "connection" element (i.e. binding) since **connection component** is a component.
- With our notation, complex connector and simple connector can coexist (and be represented without complex connector as a first class element of architecture).

Questions?