P11. *Architecture Analysis*

2014

Sungwon Kang

11. 아키텍처의 분석

11.1 아키텍처 분석과 평가의 정의

11.2 아키텍처의 분석

11.3 아키텍처 분석 방법의 사례: [ATAM 99]

11.1 아키텍처 분석과 평가의 정의

- **Analysis (분석)**
  - “separation of a whole into its component parts”
  - “identification or separation of ingredients of a substance”
  - “examination of a complex, its elements, and their relations”

- **Evaluation (평가)**
  - “determining or fixing the value of something”
  - “determining the significance, worth, or condition of something usually by careful appraisal and study”

[Source: Webster Dictionary]

Analysis and Evaluation

분석

- Investigating by analyzing
- Need viewpoint
- Provide raw data for judgment

평가

- Integrate analysis results and make judgment
- Mapping to the ultimate value measures

☞ Since analysis is a process of investigating something by closely looking into it, need a process of integration to utilize the analysis results
☞ Evaluation can be further decomposed into a hierarchy of evaluations
☞ Architecture analysis and evaluation assumes the existence of architecture or architectural strategies

Architecture Analysis/Evaluation Methods

1. SAAM: Software Architecture Analysis Method (1994)
2. ATAM: Architecture Tradeoff Analysis Method (1998, 2000)
3. ARID: Active Review for Intermediate Designs (2000)
4. CBAM: Cost-Benefit Analysis Method (2002)
5. BITAM: Business IT Alignment Method (2005)

☞ Each has aspects of both analysis and evaluation

☞ **Can classify them depending on whether the dominant results are analysis results or architecture evaluation results**

Analysis Methods and Evaluation Methods

Analysis Methods	Evaluation Methods
SAAM	
	CBAM
ATAM	
	BITAM
ARID	

- No standard definitions of analysis/evaluation exist.
  => Often ‘analysis’ and ‘evaluation’ are used interchangeably.

11.2 아키텍처의 분석

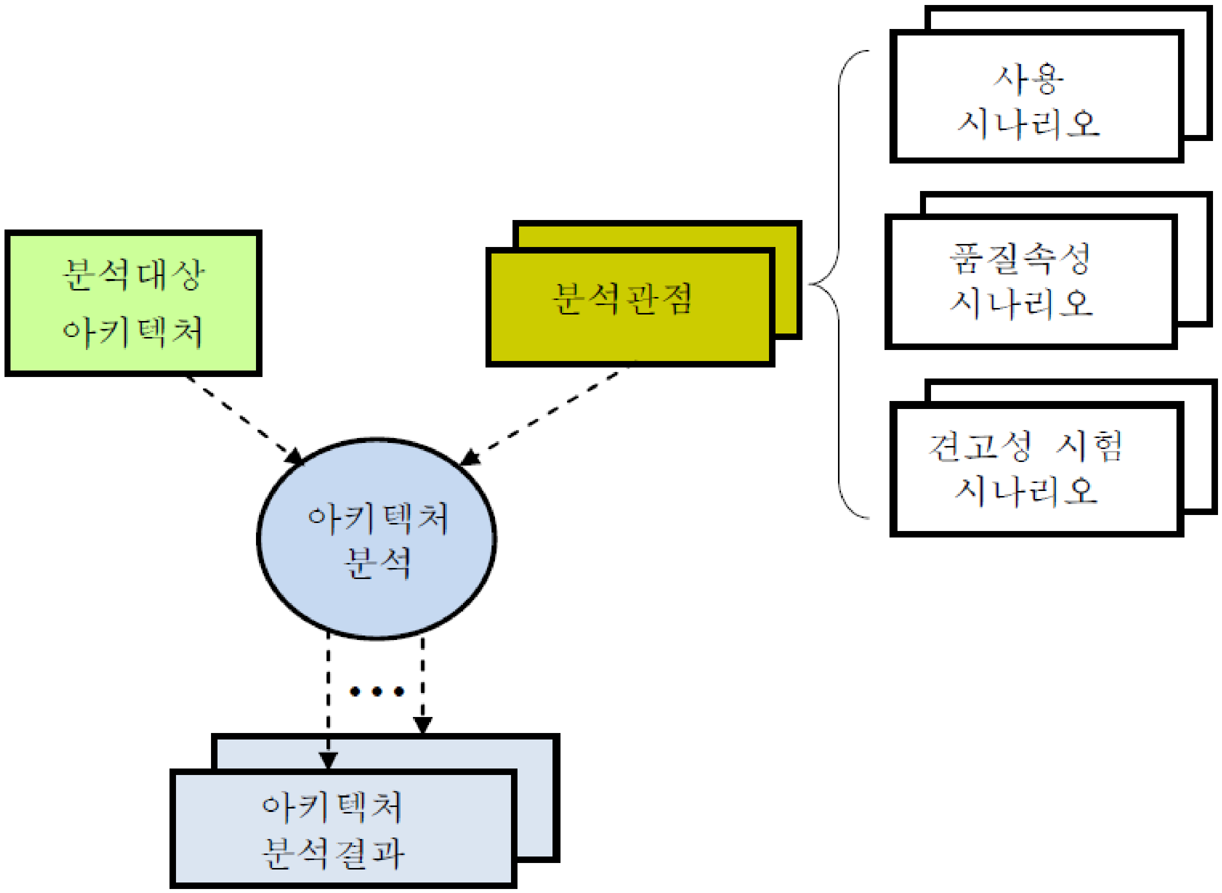


그림 11-1. 아키텍처 분석활동

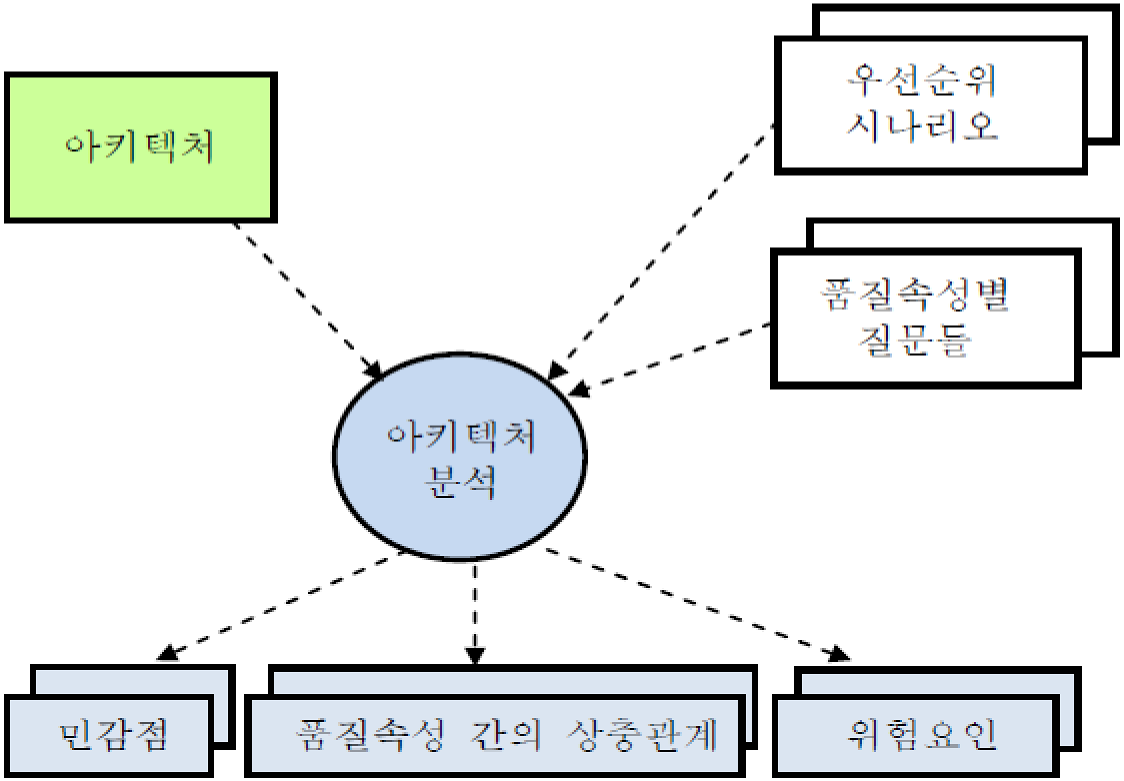


그림 11-2. ATAM 분석의 개요

11.3 아키텍처 분석의 사례

Analysis Case Study: [Kazman 98][Kazman 00]

Battlefield Control System

- BCS (Battlefield Control System)
  - For army battalions to control the movement, strategy, and operations of troops in real time in the battlefield.
    - US government: provides requirements
    - Contractor: builds the system

Battlefield Control System

Business goals…	Relevant Quality Attributes…
"The system must be accessible by customers 24/7."	*Availability* <– **We will focus on this.**
"User data must never be compromised."	*Security*
"Must have a faster response time than our competitors."	*Performance* <– **We will focus on this.**
"Deliver version 1 by 1Q2002 and version 2 by 4Q2002."	*Modifiability (maybe others…)*

Architecture

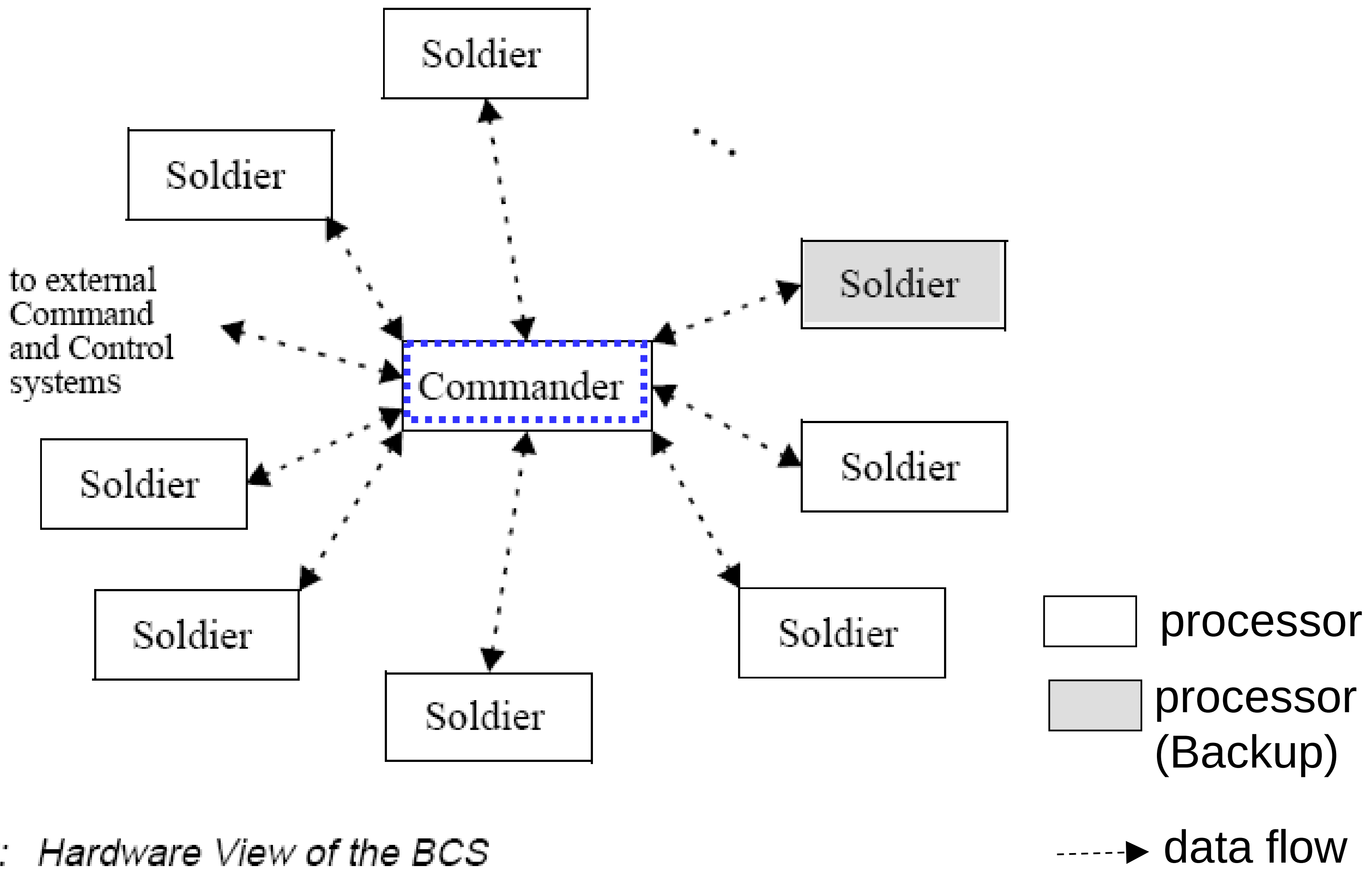
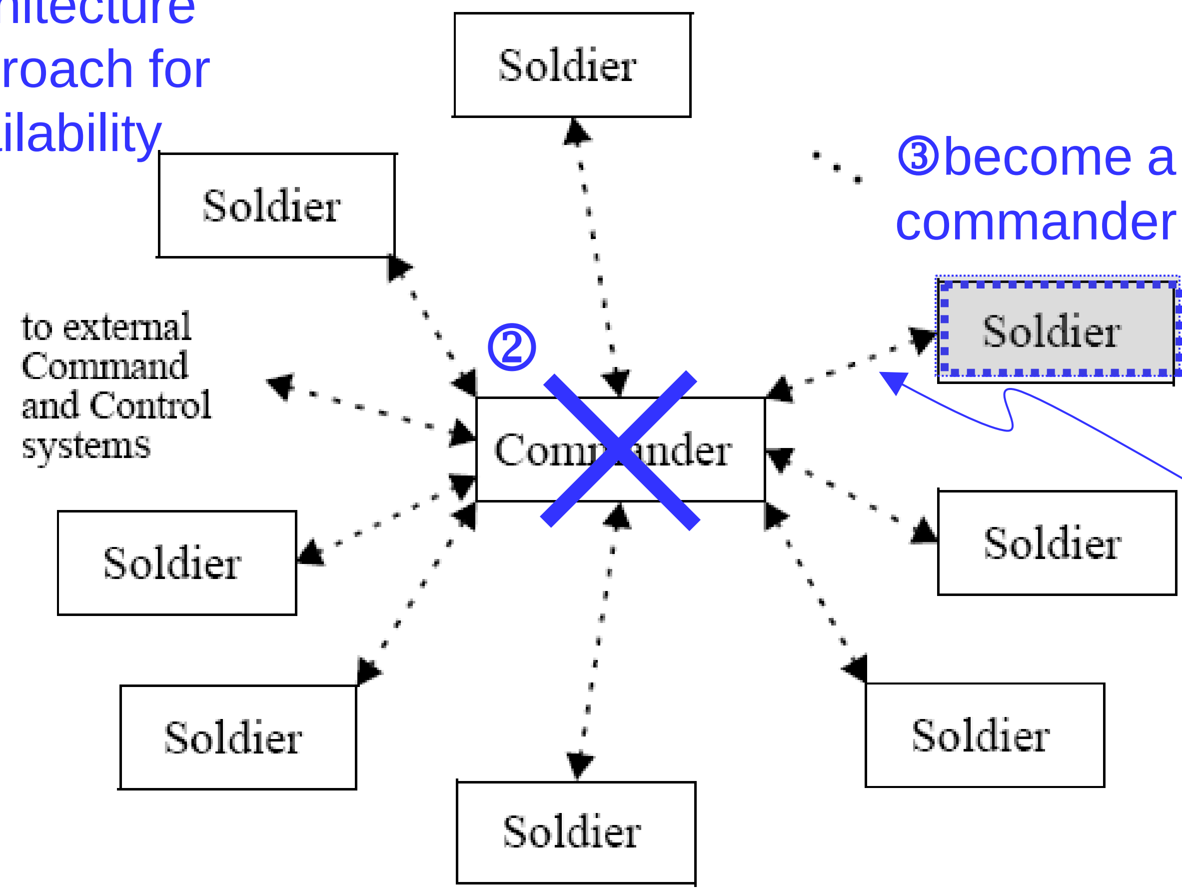


Figure 12: Hardware View of the BCS

Architectural approaches

Architecture approach for Availability



③become a new commander

☛ Availability

①Acknowledged communication (state messages) takes place between the Commander and the backup Soldier node

Figure 12: Hardware View of the BCS

Architectural approaches

Architecture approach for Performance

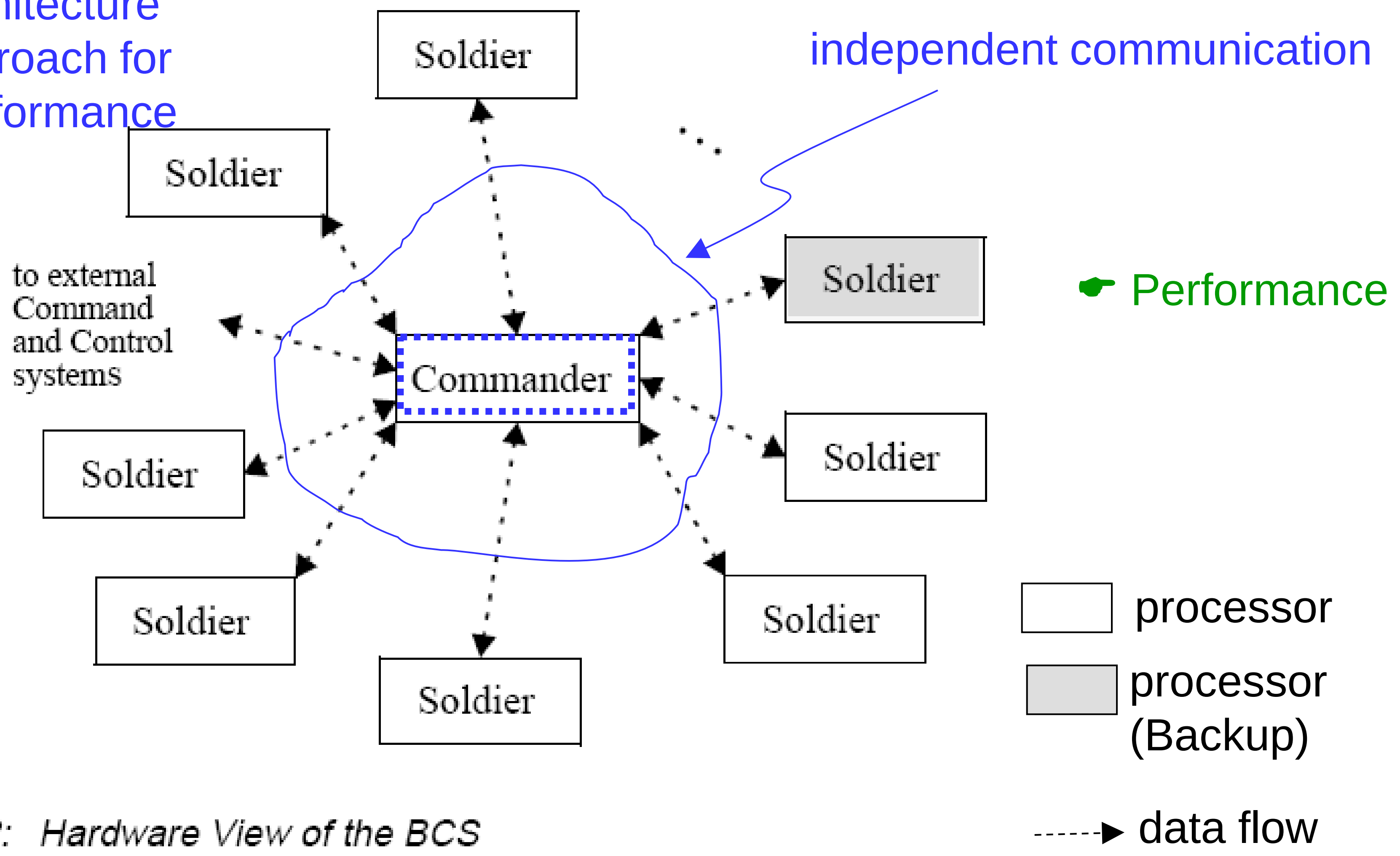


Figure 12: Hardware View of the BCS

Architectural approaches

- We elicited information on the architectural approaches with respect to modifiability, availability, and performance scenarios.

- Main approach: clients and servers
  - Additional approaches:
    1. (Availability) A backup commander approach ☜ System quality
    2. (Performance) An independent communicating components approach ☜ System quality
    3. (Modifiability) Standard subsystem organizational patterns ☜ Development quality

$Q_S = f(Q_M, Q_A, Q_P)$

"Quality of the system is a function of the quality of modifiability, availability and performance."

Analyze architectural approaches

- For example, for the backup commander approach (for availability) we generated **a set of questions**:
  - How is the failure of a component detected?
  - How is the failure of a communication channel detected?
  - What is the latency for a spare component to be turned into a working replacement?
  - By what means is a failed component marked for replacement?
  - How are the system's working components notified that they should use the services of the spare?

- **Definitions**
  - The system is considered to be working if there is a working Commander and any number of Soldier nodes.
  - When the Commander fails, the system has failed.
  - The repair time for the system is the time to turn a Soldier node into the Commander and thus restore the system to operation.

Availability Analysis

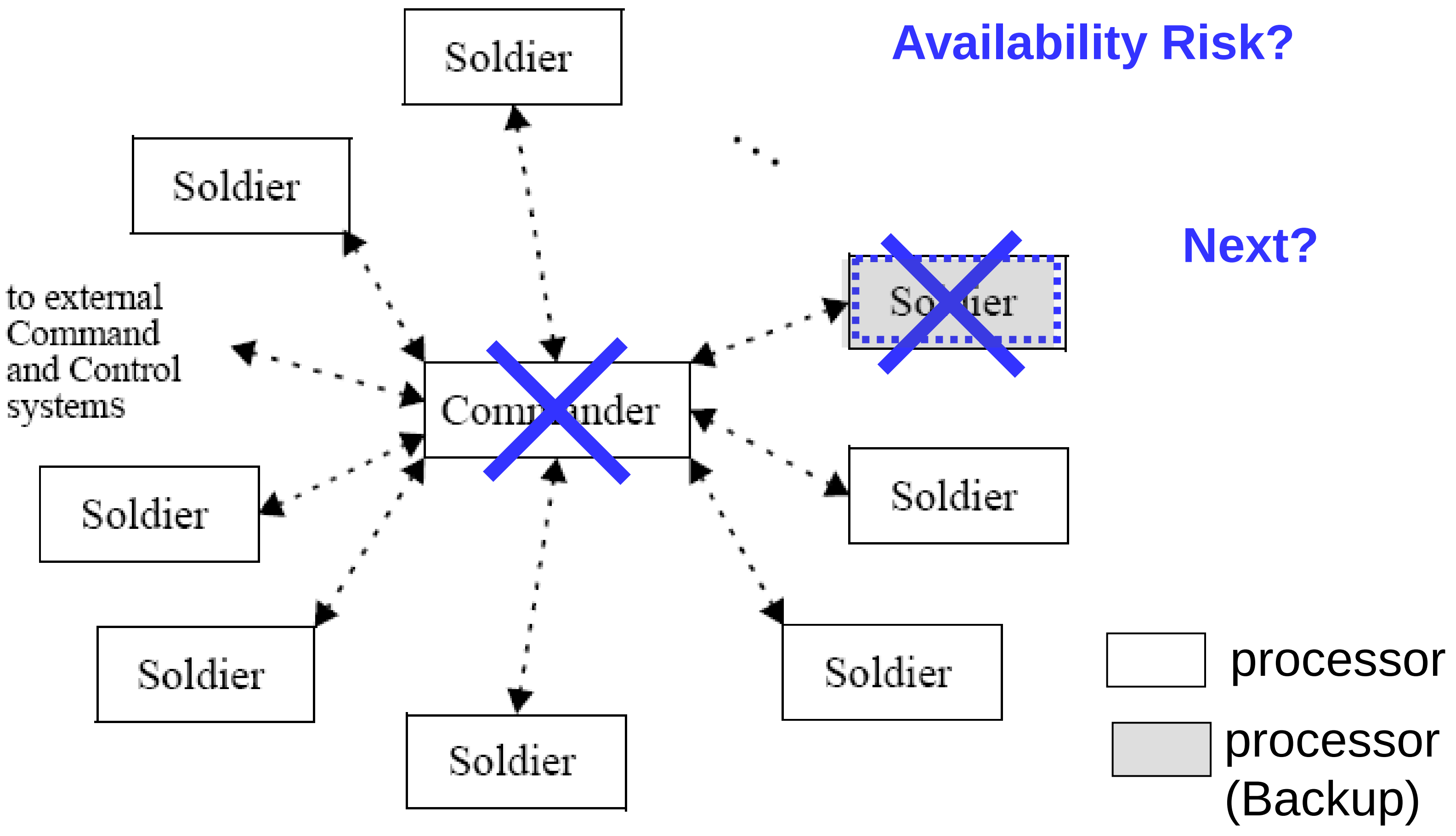


Figure 12: Hardware View of the BCS

Availability Analysis – Derive Alternatives

Considerations for Architecture Improvement

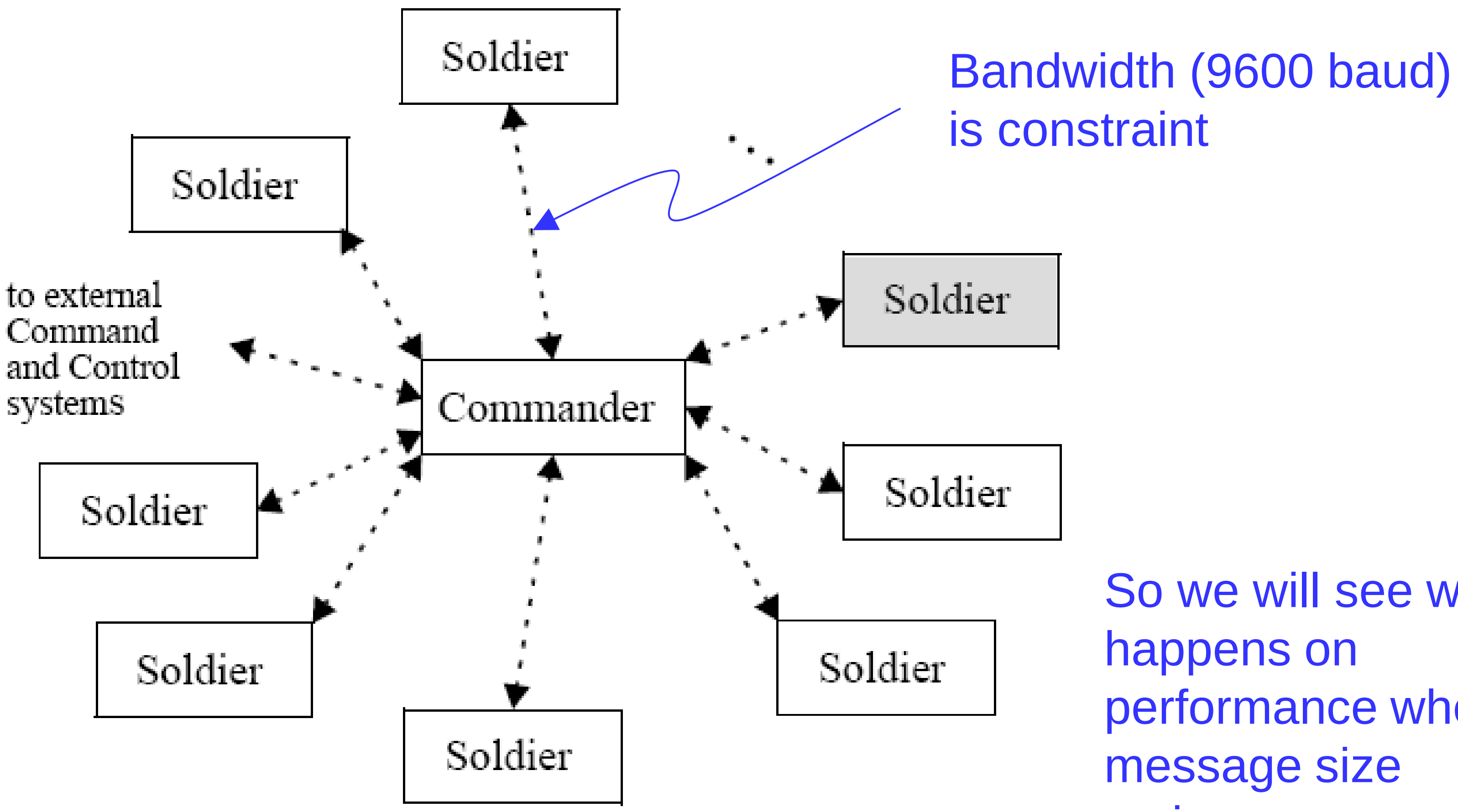
1. A backup could be an "acknowledging backup", which is kept completely synchronized with the Commander
2. A backup might be only a "passive backup" and not ask for re-sends
3. A backup, when it becomes the new Commander, or when it becomes an "acknowledging backup," could request any missed information from the upper level Command and Control systems and/or the other Soldier nodes.

Or generally where *n* is *the number of acknowledging backups* and *m* is *the number of passive backups*, the availability can be described as:

$$Q_A = g(n,m)$$

☞ As availability increases, the communication overhead increases, (Tradeoff)

Performance Analysis



So we will see what happens on performance when message size varies.

Figure 12: Hardware View of the BCS

Performance Analysis

- Three situations to consider to build performance model:

A) Regular, periodic data updates to the Commander, various message sizes and frequencies.

B) Turning a Soldier node into a backup:

Switchover requires that the backup acquires information about

- all missions
- updates to the environmental database
- issued orders
- current Soldier locations and status
- detailed inventories from the Soldiers.

C) Doubling number of weapons or the number of missions.

Performance Analysis

- Need to consider
  - Various message sizes
  - 9600 baud modem rate
  - the fact that there are a maximum of 25 Soldiers per Commander

- Downloading mission plans:
  - 280 Kbits / 9.6 Kbits/second = 29.17 seconds
- Updates to environmental database:
  - 66 Kbits / 9.6 Kbits/second = 6.88 seconds
- Acquiring issued orders:
  - 24 Soldiers * (18 Kbits/9.6 Kbits/second) = 45.0 seconds
- Acquiring Soldier locations and status:
  - 24 Soldiers * (12 Kbits/9.6 Kbits/second) = 30.0 seconds
- Acquiring inventories:
  - 24 Soldiers * (42 Kbits/9.6 Kbits/second) = 105.0 seconds

=> Total = 216.05 seconds for Soldier to become a backup

Performance Analysis – Derive Alternatives

- Thus, to keep each backup informed of the state of the Commander requires 99.67 bits/second, or approximately 1% of the system's overall communication bandwidth.

  => Use new modem hardware with increased communication speeds
  (NOT accepted as a solution)

- The system's performance model can be described as:

  $Q_P = h(n, m, CO)$ ☛ many different solutions possible

  => The system is sensitive to
  - The number of acknowledging backups (*n*)
  - The number of passive backups (*m*)
  - Communication Overhead (*CO*).

Sensitivities and Tradeoffs

- **Performance vs. availability**
  - "High performance requires less communication overhead between the commander and the backups, which decreases availability"
  - "High availability needs high communication load between the commander and the backups, which decreases performance"

$f(x,y,z) = c_1 * x + c_2 * y^{0.5} + c_3 * z^2$
In this system consisting of x, y and z, which one is a sensitivity point?

- **Sensitivity point**
  - Definition: "A property of one or more components (and/or component relationships) that is critical for achieving a particular quality attribute response" [Kazman 00]

    Example Latency of the communications channel as determined by n and m affects performance and availability

Architectural Risks

- Risk
  - The enemy may detect heavy traffic between commander and backup and make the commander the target of attack

  => Risk Mitigation
  - Add multiple Soldier backups and keeping them more or less synchronized with the Commander.

- Non-risk example
  - "*Assuming* message-arrival rates $\leq$ once per second *and a* processing time < 30 ms, *the architecture should meet the 1-second soft deadline requirement.*"

  $\Rightarrow$ *Why non-risk ?*

  If hard deadline, risk

  If soft deadline, non-risk

Lab 4. 아키텍처 분석

- **D**에서 설계한 아키텍처에 대하여 아키텍처 문서의 **F**를 작성

- 분석관점의 예
  - 서비스의 제공여부 확인
  - 품질속성 수준의 확인
  - 민감점, 상충점, 위험 등

Questions?