

전술데이터링크 연동시스템의 개념적
소프트웨어 아키텍처 설계
(A Design of a Conceptual Software Architecture for
Inter-operational System of Tactical Data Link)

남재민·윤희병*

Abstract

To ensure interoperability among TADILs, we need inter-operational system of tactical data link that allows sharing of specific, planned information among different TADILs. In this paper, we have proposed the design of a conceptual software architecture of inter-operational system. For developing of a conceptual software architecture, we analyze the actual condition of ADSI used in the US military and identify functions and quality-attributes. Based on these factors, we design the conceptual software architecture for inter-operational system of tactical data link using Attribute-Driven Design(ADD) method. ADD is consist of three phases - choose the module to decompose, refine the module, repeat the steps for every module that needs further decomposition. To evaluate of ADD results, we apply the Software Architecture Analysis Method(SAAM) which is consist of making evaluation scenarios, choosing indirect scenarios, evaluation scenarios' interaction, and creating an overall evaluation. Through the evaluation, we verify the conceptual software architecture of inter-operational system.

(Key words: Inter-operational System, Conceptual Software Architecture, TADIL, ADD, SAAM)

* 국방대학교 관리대학원

1. 서 론

한국 공군은 자동화방공체계를 중심으로 Link-11B와 222 링크 등을 운영하고 있으며 한국 해군은 한국해군전술정보체계를 중심으로 Link-11, Link-14 등을 운영하고 있다. 그리고 한국 공군과 해군 간은 Link-11B와 Link-11 사이에 별도의 연동장치를 통해 제한된 수준의 전술데이터를 주고받고 있는 실정이다[1]. 반면에 미군은 Link-11, Link-16, Link-4A 등 다양한 전술데이터링크를 운영하고 있으며 이들 전술데이터링크 간 전술데이터의 교환을 위해서 미군은 별도의 연동시스템운영센터를 구축하여 운영하고 있다 [2]. 한국군도 미군의 연동시스템운영센터에 있는 연동시스템을 통해 전술데이터를 받고 있지만 이것은 한국 공군과 해군에 공개가 가능한 전술데이터에 국한하므로 매우 일방적이고 제한적인 데이터라 할 수 있다.

현재 한국 공군은 F-15K를 도입 중에 있으며, 해군은 KDX-III를 도입할 예정이므로 이러한 두 가지의 무기체계를 통해 한국 공군과 해군은 또 하나의 새로운 전술데이터링크를 갖추게 될 것이다. 따라서 기존의 플랫폼과 새롭게 도입하거나 도입할 예정인 플랫폼과의 효율적이고 원활한 전술데이터의 교환 및 상호 공유를 위해서는 무엇보다 먼저 이들 플랫폼들이 보유하고 있는 전술데이터링크 간 전술데이터를 교환할 수 있는 연동시스템의 개발이 필요한 실정이다.

연동시스템을 개발하기 위해서는 먼저 사용자의 요구사항에 대한 분석과 이러한 분석을 통한 연동시스템의 개념연구가 먼저 선행되어야 한다. 그러

나 지금까지 한국군에는 이러한 연동시스템에 대한 수요가 요구되지 않았고 앞으로 요구되는 사항이었기 때문에 현재까지 한국군에서는 연동시스템과 관련한 개념연구가 수행되어 있지 못한 실정이다.

따라서 본 연구에서는 한국군에 필요한 전술데이터링크의 연동시스템을 구축하기 위해서 먼저 미군에서 운용하고 있는 연동시스템인 방공체계통합기(ADSI)로부터 그 개념을 가져오고 이를 이용하여 본 논문의 주목적인 연동시스템의 개념적 소프트웨어 아키텍처를 설계한다. 설계를 위해 가장 권위 있는 CMU/SEI의 ADD(Attribute-Driven Design) [3] 설계기법을 사용하였으며 소프트웨어 아키텍처 설계에 대한 평가는 CMU/SEI의 SAMM(Software Architecture Analysis Method) 기법[4]을 이용한다.

2. 관련연구

2.1 방공체계통합기(ADSI)

ADSI(Air Defense System Integrator) [5]는 미군의 전술데이터링크 연동에 핵심적인 장비로서 전술정보를 수신, 전송 및 전시할 수 있는 자료통신 및 융합체계라고 할 수 있다. ADSI는 전술데이터링크, 레이더자료, 정보자료, 근실시간 자료를 처리할 수 있는 능력을 가지고 있다. ADSI 소프트웨어는 C언어로 작성되어 있으며, 대략 200개의 모듈과 약 200만 라인으로 구성된다. ADSI는 함정, 항공기, 이동장비 및 상업용 컴퓨터 등 다양한 하드웨어에 장착될 수 있다. ADSI 소프트웨어는 다양한 형태의 센서로부터 들어온 항목을 2,000개까지 전시가 가능하며 최대 16개의 데이터링크 내에서 4,000트랙까지 전송이 가능하

다. 그 외에 공중·지상을 포함함 총 24개의 레이더로부터 자료를 수신할 수 있는 등 다양한 능력을 가지고 있다.

2.2 소프트웨어 아키텍처 설계 및 평가기법

2.2.1 소프트웨어 아키텍처 설계기법

소프트웨어 아키텍처 설계기법은 오늘날 많은 사람들이 그 방법을 제시하고 있으며 아직도 진화하고 있는 분야라 할 수 있다. 그 중에서 본 연구에서 사용한 설계기법인 ADD[3]는 미국 카네기 멜론 대학의 소프트웨어 공학연구소(SEI : Software Engineering Institute)에서 제안한 설계기법이다.

ADD는 품질속성 요구사항과 기능적 요구사항 모두를 만족시키며 소프트웨어가 실행해야 하는 품질속성의 분할과정에 기초하여 소프트웨어 아키텍처를 정의하는 기법이다. 이 방법은 반복적인 분할과정으로 각 단계별로 품질속성 시나리오를 만족하는 전술(tactics)과 아키텍처 패턴이 선택되며, 패턴에 의해 생성된 모듈 타입에 기능이 할당된다. ADD의 입력물은 품질속성 시나리오, 제약사항, 기능적 요구사항 등이다.

2.2.2 소프트웨어 아키텍처 평가기법

소프트웨어 아키텍처 평가는 목표시스템에 적절하게 아키텍처가 설계되었는지 또는 가장 적절한 아키텍처가 무엇인지 판단하는 작업이라 할 수 있다. 이러한 평가기법 중에 CMU/SEI에서 개발한 시나리오 기반의 소프트웨어 아키텍처 평가 방법인 SAAM[4], ATAM(Architecture Trade-off Analysis Method) [6], 그리고 CBAM(Cost

Benefit Analysis Method) [7] 등이 대표적이라 할 수 있다.

SAAM은 시나리오를 기반으로 소프트웨어 아키텍처를 평가하는 최초의 기법으로 널리 알려져 있으며 변경가능성을 평가하기 위해 만들어졌으나, 그 외에도 확장성, 이식성 등을 평가하는데도 유용하게 사용되는 평가기법이다. SAAM의 평가절차는 브레인스토밍을 통한 시나리오 개발, 후보 아키텍처 설명, 그리고 시나리오에 우선순위를 부여하며, 직·간접 시나리오 분류, 간접시나리오 평가, 둘 이상의 시나리오가 동일 컴포넌트에 대해 변경을 요구할 경우를 평가하는 시나리오 상호작용 평가, 그리고 전반적인 평가 등의 총 6단계로 구성된다[4].

ATAM은 SAAM에서 유래된 시나리오 기반의 소프트웨어 아키텍처 평가기법으로 품질속성을 찾아내고 아키텍처가 특정한 품질속성을 만족하는지를 분석하며 위험요소, 민감점과 품질속성 간의 충돌을 분석해서 절충점을 찾아내는 평가기법이다 [6].

CBAM은 아키텍처와 관련된 결정을 할 때 비용, 이익, 위험 등을 적용하여 의사결정을 지원하는 기법이다[7]. <표 1>에 3가지의 소프트웨어 아키텍처 평가기법에 대한 비교가 나타나 있다.

<표 1> 소프트웨어 아키텍처 평가기법 비교

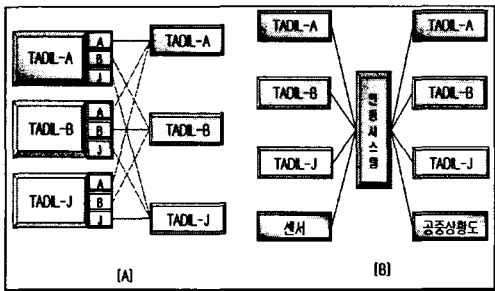
구분	평가속성	평가수단	평가대상
SAAM	변경 가능성	시나리오분류 (직/간접시나리오)	모두
ATAM	시스템 품질속성	민감포인트 협상포인트	모두
CBAM	비용, 이익	시간, 비용	모두

3. 연동시스템 요구사항 도출

3.1 연동시스템 운영개념

한국군의 연동시스템 운영을 위한 2가지의 가능한 연결개념이 <그림 1>에 도시되어 있다. <그림 1>의 [A]의 경우, 즉 ‘메시형’에는 연동을 위한 추가 장비 없이 연동이 가능하다는 장점이 있으나 플랫폼마다 전술데이터를 변환해야만 하는 별도의 추가 장치를 설치해야 한다는 단점을 가지고 있다.

그러나 <그림 1>의 [B]의 경우, 즉 ‘연동형’에는 별도의 연동시스템이 필요하다는 단점이 있으나, [A]에 비해 연결선을 줄일 수 있어 복잡도를 많이 낮출 수 있고 또한 전술데이터링크의 추가, 삭제 및 변경이 용이하고 융통성이 크다는 장점을 가지고 있다. 따라서 한국군의 전술데이터링크 연동시스템은 <그림 1>의 [B]와 같은 ‘연동형’으로 연결해야 한다.



<그림 1> 연동시스템 연결개념

3.2 연동시스템 요구사항 및 달성 전술 도출

3.2.1 기능 요구사항

연동시스템에 대한 기능 요구사항은 미군의 ADSI와 <그림 1>의 연동시스템 연결개념으로부터

더 도출이 가능하다. 연동을 위해서는 먼저 서로 다른 메시지를 교환하는 전술데이터링크 처리 기능, 메시지 포맷변환 기능, 전술상황도 작성을 위한 항적생성 기능, 처리한 자료나 항적자료를 타 체계로 제공할 수 있는 전송기능 등이 필요하다. 지금까지 제시한 기능들을 좀 더 세분화하여 나타낸 기능 요구사항들이 <표 2>에 나타나 있다. 여기서 R은 요구(Request)를 의미한다.

<표 2> 기능 요구사항

순서	내용
R1	연동시스템은 다양한 센서로부터 탐지자료를 입력받을 수 있어야 한다.
R2	연동시스템은 다양한 전술데이터링크로부터 전술데이터 메시지를 입력받을 수 있어야 한다.
R3	연동시스템은 자료를 종합하여 항적을 생성할 수 있어야 한다.
R4	연동시스템은 생성된 항적의 구분 및 이중 항적삭제는 자동으로 이루어져야 한다.
R5	연동시스템은 실시간 항적과 비실시간 자료를 일치시킬 수 있어야 한다.
R6	워크스테이션을 이용하여 자료조회 및 메시지 입·출력이 가능해야 한다.
R7	연동시스템은 메시지 포맷을 분석하고 오류를 탐지할 수 있어야 한다.
R8	연동시스템은 Link-11/11B/16의 메시지 포맷을 상호 변환할 수 있어야 한다.
R9	연동시스템은 메시지변환을 위한 자료저장 및 변환기록 유지와 항적자료 저장을 위한 데이터베이스가 필요하다.
R10	연동시스템은 변화된 메시지나 생성된 항적은 타 체계로 전송 가능해야 한다.

3.2.2 품질속성과 달성전술 도출

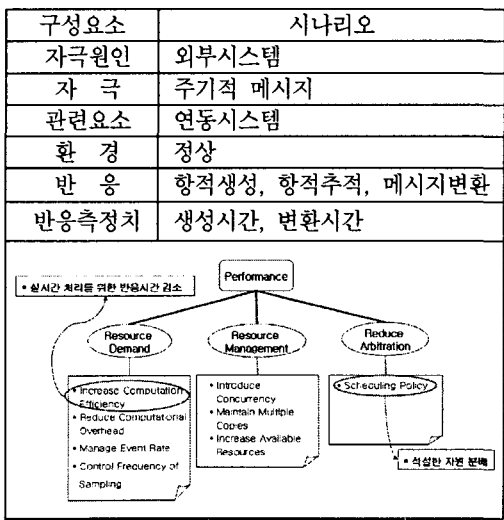
본 연구에서는 CMU/SEI에서 정의한 6가지의 품질속성[3]인 가용성, 성능, 변경가능성, 사용성, 테스트 가능성, 보안성 중에서 연동시스템 구축에 가장 중요하다고 판단된 3가지를 선정하였다. 먼

저 연동시스템에서의 실시간 처리 보장과 같은 시스템의 적시 기능수행 정도를 나타내는 성능(Performance)이고, 둘째로는 전술데이터링크의 변경이 용이하고 다른 요소들에게 미치는 영향을 최소화하면서 짧은 변경시간과 적은 비용이 필요하도록 해주는 변경가능성(Modifiability)이며 마지막으로 필요할 때 사용할 수 있도록 시스템의 24시간 지속적인 운용을 보장해 줄 수 있는 가용성(Availability) 등이다.

가. 성능

성능의 구체적 품질속성 시나리오와 이러한 시나리오를 구현할 수 있는 전술 선택과 관련한 내용이 <표 3>에 나타나 있다. 선택된 전술을 살펴보면 먼저 자원요구(Resource Demand)에서는 실시간 처리를 위해 반응시간을 줄일 수 있는 효율적인 계산(computation efficiency)이 있으며 중재 감소(Reduce Arbitration)에서는 적절한 자원분배를 위한 스케줄링 정책(Scheduling Policy) 등이 있다.

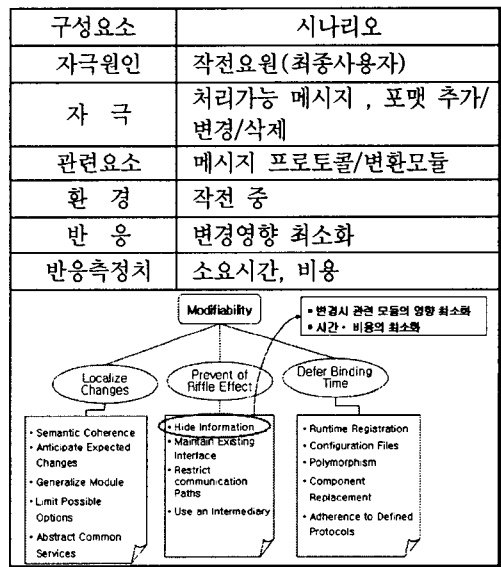
<표 3> 성능의 구체적 품질속성 시나리오와 전술 선택



나. 변경가능성

변경가능성의 구체적 품질속성 시나리오와 이러한 시나리오를 구현할 수 있는 전술 선택과 관련된 내용이 <표 4>에 나타나 있다. 여기서 선택된 전술에는 정보감추기(Hide Information)로서 이 전술은 시스템 변경시 관련 모듈의 영향을 최소화하고 이에 따라 시간 및 비용을 최소화할 수 있는 전술이다.

<표 4> 변경가능성의 구체적 품질속성 시나리오와 전술 선택



다. 가용성

가용성의 구체적 품질속성 시나리오와 이의 전술 선택 관련 사항이 <표 5>에 나타나 있다. 이러한 가용성 확보를 위해 선택된 전술에는 결합탐지를 위해 정해진 시간 내에 신호와 응답을 주고받는 핑/에코(Ping/Echo)와 결합회복을 위한 수동 중복성(Passive Redundancy) 등이 있다. 이 방법은 한 컴포넌트가 이벤트에 반응하고 이를 대

기 중인 컴포넌트에 알려 업데이트 시키는 방법으로 오류가 발생했을 때 대기 중인 컴포넌트를 통해 즉시 복구가 가능한 전술이다.

<표 5> 가용성의 구체적 품질속성 시나리오와 전술 선택

구성요소	시나리오	
자극원인	외부시스템	내부시스템
자극	타이밍 에러 메시지 입력	메시지변경 실패
관련요소	메시지변경프로세스	메시지변경프로세스
환경	정상작동	정상작동
반응	로그기록, 정상작동	로그기록, 정상작동
반응측정치	지연시간 없음	지연시간 없음


```

graph TD
    Availability[Availability] --> FaultDetection[Fault Detection]
    Availability --> RecoveryPreparation[Recovery-Preparation and Repair]
    Availability --> RecoveryReintroduction[Recovery-Reintroduction]
    Availability --> Prevention[Prevention]
    
    FaultDetection --- PingEcho["Ping/Echo  
Heartbeat  
Exception"]
    RecoveryPreparation --- Voting["Voting  
Active Redundancy  
Passive Redundancy"]
    RecoveryReintroduction --- Shadow["Shadow  
State Resynchronization  
Rollback"]
    Prevention --- RemovalFromService["Removal from Service  
Transactions  
Process Monitor"]
    
    PingEcho --- Checksum[checksum]
    Voting --- Spare["Spare"]
    Shadow --- AvailabilityInfo["가용성 확보를 위한 이중구조"]
  
```

4. 연동시스템 개념적 소프트웨어 아키텍처 설계

4.1 ADD 1단계 설계

4.1.1 요구사항 입력

3.2.2절에서 도출한 품질속성 시나리오를 고려하여 요구사항 입력을 위해 만든 연동시스템 요구사항이 <표 6>에 나타나 있다.

4.1.2 분할모듈 선택

연동시스템 요구사항을 바탕으로 분할할 모듈을 선택한다. 본 연구에서는 분할할 모듈로 연동시스템 전체를 선택한다.

<표 6> 연동시스템 요구사항

구분	요구사항
성능	<ul style="list-style-type: none"> 연동시스템은 정상작동 중 외부시스템에서 주기적으로 입력되는 메시지를 실시간으로 처리해야 한다. 항적생성, 항적추적, 메시지 변환을 수행할 수 있어야 한다.
변경가능성	<ul style="list-style-type: none"> 최종 사용자에게 의해 메시지 포맷 추가, 변경 또는 삭제 요청시 메시지 프로토콜과 메시지 변환모듈을 변경해야 한다. 이때 다른 모듈에 대한 변화 및 영향을 최소화해야 하며 소요시간과 비용도 적어야 한다.
가용성	<ul style="list-style-type: none"> 연동시스템이 정상작동 중 외부시스템에서 오류가 있는 메시지가 입력되는 경우에도 무시하고 정상 작동해야 한다. 연동시스템이 정상작동 중 내부시스템에서 메시지변경에 실패한 경우 로그에 기록하고 정상 작동해야 한다

4.2 ADD 2단계 설계

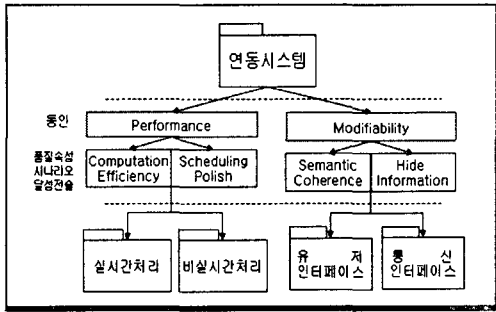
4.2.1 동인(driver) 선택

ADD 1단계에서 도출한 연동시스템 요구사항 중 특히 아키텍처 설계에 중요한 영향을 미치는 요구사항을 동인(Driver)이라 한다. 동인으로 메시지의 추가, 변경, 삭제가 용이하도록 융통성을 보장해주는 변경가능성과 항적통합과 전술상황 전이를 통해 실시간 처리를 보장해주는 성능을 선택한다.

4.2.2 아키텍처 패턴 선택

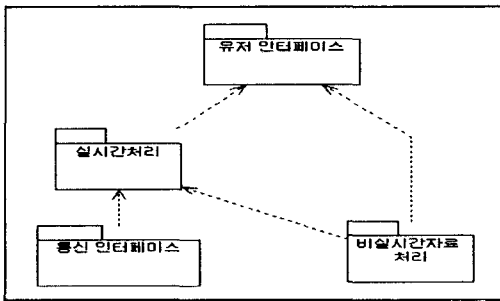
본 절은 3.2.2절에서 선택된 전술에 따라 동인을 만족할 수 있도록 시스템을 분할하고 분할된 모듈을 사용하여 아키텍처 패턴을 설계하는 과정이다. 먼저 분할을 위해 성능은 실시간과 비실시간 처리로 나누며 변경가능성은 사용자 인터페이스와 통신 인터페이스로 나눈다. 이와 같이 분할된 연동

시스템이 <그림 2>에 도시되어 있다.



<그림 2> 연동시스템 분할

<그림 2>의 분할된 모듈을 사용하여 아키텍처 패턴을 선택한다. 선택된 아키텍처 패턴은 통신 인터페이스를 통해 입력되는 자료와 비실시간 처리에서 입력되는 자료를 실시간 처리에서 종합하고 유저 인터페이스에서 조회가 가능한 구조이다. <그림 3>에 연동시스템의 아키텍처 패턴이 나타나 있다.

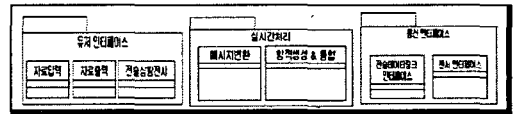


<그림 3> 연동시스템 아키텍처 패턴

4.2.3 모듈설명 및 기능할당

모듈설명 및 기능할당은 아키텍처 패턴에 나타나 있는 모듈을 필요시 분할하고 모듈에 대한 설명을 하고 기능을 할당하는 것을 말한다. 먼저 유저 인터페이스는 사용자가 시스템을 편리하게 사용할 수 있는 인터페이스를 제공하며 자료입력, 자료출력, 전술상황 전시를 담당하는 모듈로 분할한다.

둘째로 실시간 처리는 항적생성, 항적통합, 메시지 변환 등 3가지 모듈로 분할한다. 항적생성은 항적 자료의 생성과 항적의 연속성 보장을 담당하고 항적통합은 생성된 항적의 종합 및 전술상황도 작성 기능을, 메시지 변환은 전술데이터링크 메시지포맷 변환기능을 할당한다. 셋째로 비실시간 처리는 별도의 분할이 필요 없으며 비행계획서, 임무계획서, 전술상황자료, 경계선, 지형, 참조자료 등과 같이 유저 인터페이스로부터 새롭게 입력되는 자료와 기 입력 자료의 처리를 담당한다. 마지막으로 통신 인터페이스는 외부체계와의 연결을 제공하고 전술데이터링크 인터페이스와 센서 인터페이스로 분할한다. <그림 4>에 아키텍처 패턴의 분할된 모습이 나타나 있다.



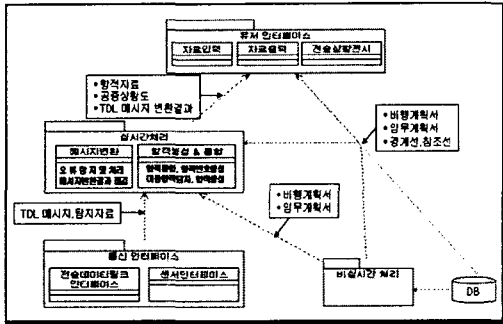
<그림 4> 연동시스템 패턴 분할

소프트웨어 아키텍처는 특정 부분만으로 표현하기 어려우므로 아키텍처에 대한 이해를 돕기 위해 뷰(View)를 사용해 표현하며 분할뷰, 병행뷰, 배치뷰 등 3종류를 사용한다.

가. 분할뷰

분할뷰는 모듈 간의 중요한 데이터 흐름을 나타낸다. 전술데이터링크 인터페이스에서는 전술데이터 메시지가 메시지변환 모듈로 이동하고 센서 인터페이스에서는 탐지자료가 항적추적 모듈로 이동한다. 비실시간 처리 모듈에서는 비행 및 임무계획서를 실시간 처리모듈로 입력하고, 유저 인터페이스로는 비행 및 임무계획서, 경계선과 참조선을 입력한다. 실시간 처리 모듈에서는 입력된 자료를 바

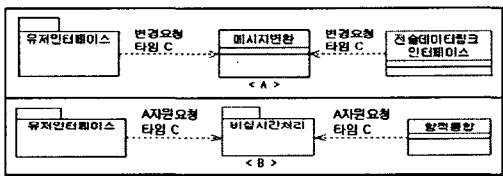
탕으로 향적을 생성하고 이중향적을 탐지하고 향적 번호생성 시 자동으로 향적정보를 통합한다. 그리고 메시지변환에서는 입력된 오류메시지를 탐지 및 처리하고 변환결과를 점검하며 이를 외부시스템 및 유저 인터페이스에 제공한다. 이와 같은 분할뷰가 <그림 5>에 나타나 있다.



<그림 5> 분할뷰

나. 병행뷰

병행뷰는 자원쟁탈 문제와 같은 교착상황을 표현한다. 교착상황은 전술데이터링크의 메시지 변환 요구와 사용자의 메시지 변환 요구가 동시에 발생한 경우에 발생한다. 또한 유저 인터페이스와 향적 통합단계에서 동시에 동일 항목의 비실시간자료를 요구할 경우, 즉 유저인터페이스와 향적통합이 동시에 임의의 비행계획서를 요구하는 경우 등이다. 이와 같은 병행뷰가 <그림 6>에 도시되어 있다.

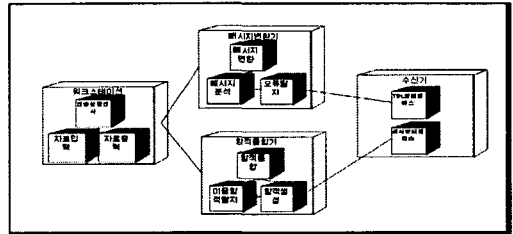


<그림 6> 병행뷰

다. 배치뷰

하드웨어와 소프트웨어의 배치를 나타내는 뷰로

서 <그림 7>에 도시되어 있다. 클라이언트 tier, 애플리케이션 tier, 데이터 tier 등 3단계로 구분되며 애플리케이션은 향적통합기와 메시지변환기로 구분된다.



<그림 7> 배치뷰

4.2.4 하위모듈 인터페이스 정의

하위모듈 인터페이스는 모듈에 따라 필요한 소비정보와 생산정보를 나타내는 것으로 <표 7>에 나타나 있다.

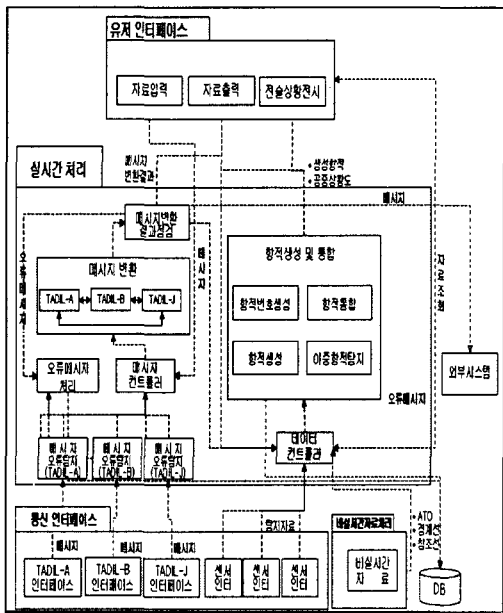
<표 7> 하위모듈 인터페이스

모 들	생산정보	소비정보
사용자 인터페이스	· 사용자 지시	· 메시지 변환결과 · 향적생성/향적정보 · 전술상황도 · ATO/ACO, 경계선
메시지변환	· 변환메시지	· 전술데이터링크 메시지 · 사용자 지시
향적추적	· 향적생성 · 위치보정	· 탐지자료 · 전술데이터
향적통합	· 향적정보 · 전술상황도	· 향적생성 · 변환메시지 · ATO/ACO
비실시간 자료	· ATO/ACO · 향적저장 · 경계선, 참고자료	· 사용자 지시
전술데이터링크 인터페이스	· 전술데이터	-
센서 인터페이스	· 탐지자료	-

4.3 연동시스템 개념적 소프트웨어

아키텍처 설계

ADD 1단계 및 2단계 작업을 통해 수행한 결과를 종합하여 연동시스템에 대한 개념적 소프트웨어 아키텍처를 설계한다. 이를 위해 4.2.3절의 모듈설명 및 기능할당, 분할뷰, 병행뷰 및 배치뷰를 종합하였으며 종합된 소프트웨어 아키텍처가 <그림 8>에 나타나 있다.



<그림 8> 연동시스템의 개념적 소프트웨어 아키텍처

<그림 8>에는 각 전송데이터링크별로 메시지 오류 탐지모듈이 있어 오류 메시지와 정상 메시지를 분류하므로 메시지 컨트롤러의 부하를 줄일 수 있다. 메시지 변환결과 점검모듈은 변환이 성공적으로 이루어진 메시지에 한해 전송되도록 하며 실패한 메시지의 경우에는 오류 메시지처리 모듈로 전송된다. 센서자료가 입력되는 경우에는 항적생성 및 통합 모듈로 입력되고 항적생성, 이중항적탐지,

항적통합의 절차를 거쳐 전송상황도가 작성된다.

4.4 연동시스템 개념적 소프트웨어

아키텍처 평가

본 절은 4.3절에서 제안한 연동시스템의 소프트웨어 아키텍처 설계를 CMU/SEI의 SAAM 기법을 이용하여 평가하며 평가결과를 제시한다.

4.3.1 시나리오 작성 및 선택

가. 시나리오 작성

평가 시나리오는 ADD 1단계에서의 연동시스템 요구사항과 ISO9126, IEEE 등에 명시된 품질속성에 기초하여 작성한다. 시나리오는 직접·간접시나리오로 구성되며, 직접시나리오는 제안된 아키텍처에 의해 만족된 시나리오를 의미하고, 간접시나리오는 아키텍처의 변경이나 분할을 통해 만족될 수 있는 시나리오를 의미한다[4]. 작성된 평가 시나리오가 <표 8>에 나타나 있으며 여기서 S는 시나리오(Scenario)를 의미한다.

나. 간접시나리오 선택

<표 8>의 S1, S2, S4, S5, S9는 제시된 아키텍처에 의해 만족되었으므로 직접시나리오로 분류하고 S6, S7은 간접시나리오로 분류한다. 이 시나리오를 만족시키기 위해서는 해당 요소의 변경이나 분할이 필요하다.

4.3.2 시나리오 평가

가. 간접시나리오 평가

간접시나리오 평가는 선택된 간접시나리오에 대한 달성방법을 제시하는 것을 말한다. 간접시나리오인 S6, S7의 평가 결과, 오류메시지처리 모듈에 오류메시지의 강제 전시 및 데이터베이스 기록 등의 기능이 추가되어야 한다.

<표 8> 평가 시나리오

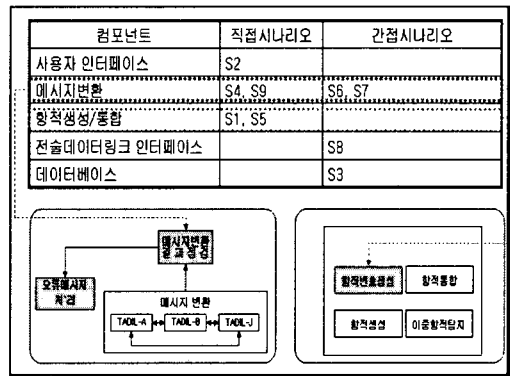
번호	시 나 리 오
S1	연동시스템은 전술데이터링크와 센서로부터 입력받은 자료를 이용해 항적을 생성한다.
S2	워크스테이션을 통한 자료조회, 메시지 입력, 전술상황 전시가 가능해야 한다.
S3	연동시스템은 데이터베이스를 이용하여 항적자료 저장, 재생, 비실시간자료, 메시지 변환 데이터 및 변환기록을 저장한다.
S4	연동시스템은 Link-11/11B/16의 메시지를 상호 간 변환해야 한다.
S5	연동시스템은 생성된 항적과 비행계획서를 일치시켜야 한다.
S6	연동시스템이 정상작동 중 외부시스템에서 오류 메시지가 입력되는 경우 메시지변경 프로세스는 이를 삭제하고 로그에 기록한 후 정상작동을 계속한다.
S7	연동시스템이 정상작동 중 내부시스템의 메시지변경이 실패한 경우 메시지변경 프로세스는 로그에 기록함과 동시에 화면에 강제 전시함으로써 운영자에게 통보해야 한다.
S8	연동시스템은 변환된 메시지나 생성된 항적을 타 체계로 전송 가능해야 한다.
S9	최종 사용자에 의해 메시지 포맷의 추가, 변경 또는 삭제 요청이 있을시 메시지 프로토콜과 메시지변환 모듈의 변경이 용이하여야 하며, 이때 다른 모듈로의 변화 및 영향을 최소화하여야 한다.

나. 시나리오 상호작용 평가

2개 이상의 시나리오가 같은 모듈에 대해 변화를 요구할 때, 해당 시나리오들은 상호작용 한다고 말한다. 이 경우 영향을 받는 모듈은 시나리오들의 상호작용으로 인한 충돌을 피하기 위해 변경되거나 분할되어야 한다. <표 8>에 제시된 시나리오 중 상호작용하는 시나리오를 분석해 보면 메시지 변환과 관련된 시나리오들은 S4, S9, S6, S7이 있고, 항적생성 및 통합과 관련된 시나리오들은 S1, S5가 있다.

S4, S9, S6, S7에 대한 상호작용의 경우, 메시지 변환결과 점검모듈에서 메시지변환 실패 시

이를 오류메시지 처리모듈로 전송함과 동시에 사용자에게도 인지시킨다. 또한 오류메시지 처리모듈에서 오류메시지 입력 시 이를 데이터베이스에 기록하여 메시지 컨트롤러로 입력되는 것을 원천적으로 차단하여 시스템 부하를 줄인다. S1, S5에 대한 상호작용의 경우에는 항적번호 생성모듈에서 항적과 비행계획서를 일치시키는 역할을 하므로 충돌문제가 해결된다. <그림 9>에 이와 같은 시나리오 상호작용 평가 결과가 나타나 있다.



<그림 9> 시나리오 상호작용 평가결과

5. 결 론

본 연구에서는 전술데이터링크 간 전술데이터를 교환하기 위한 연동시스템의 개념적 소프트웨어 아키텍처를 설계하였다. ADD 1단계 설계에서는 요구사항을 입력하고 분할모듈로서 연동시스템을 선택하였다. 2단계에서는 동인으로 성능과 변경가능성을 선택하였으며, 연동시스템을 유저 인터페이스, 실시간 처리, 비실시간 처리, 커뮤니케이션 인터페이스 등의 4개 모듈로 분할하였다. 그리고 이들 4개 모듈을 연결해 아키텍처 패턴을 제시하였다. 하위모듈의 인터페이스정의에서는 각각의 모듈 별로 생산정보와 소비정보를 정의하였다. 또한 분

할뷰를 이용하여 모듈 간의 주요 데이터흐름을 파악하였고 병행뷰를 이용하여 자원쟁탈문제와 같은 가능한 교착상황을 파악하였다. 배치뷰에서는 하드웨어와 소프트웨어의 배치를 표현하였고 이를 종합하여 연동시스템의 개념적 소프트웨어 아키텍처를 설계하였다.

그리고 제안한 연동시스템의 소프트웨어 아키텍처 설계를 평가하기 위해 CMU/SEI의 SAAM 기법을 이용하여 평가하였으며, 평가 결과 제안한 연동시스템의 개념적 소프트웨어 아키텍처가 평가 시나리오를 만족하는 것을 확인하였다.

Method for Architecture Evaluation
Carnegie Mellon University, 2000.

[7] http://www.sei.cmu.edu/ata/products_services/cbam.html

참 고 문 헌

- [1] 윤희병, "전술데이터링크 구성/기능 및 발전 방향," 공군정보통신지, 2003.
- [2] DoD, *CAI Joint Tactical Data Link Management Plan*, DoD, June 2000.
- [3] Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, 2nd Edition, Addison-Wesley, 2003.
- [4] Mugurel T. Ionita, Dieter K. Hammer, Henk Obbink, "Scenario-Based Software Architecture Evaluation Methods : An Overview," International Conference on Software Engineering 2002 (ICSE02), pp.1-12, 2002.
- [5] <http://www.apcinc.com>
- [6] Rick Kazman, Mario Barcci, Mark Klein, Paul Clements, *ATAM*: