

시멘틱 웹서비스를 이용한 워게임 시뮬레이터 제작

김병철¹ · 이강선^{1†}

Wargame Simulator using Semantic Web Service

Byoung-Chul Kim · Kang-Sun Lee

ABSTRACT

The next-generation war game simulators need a technique that reuses resources disperse on the web, and reorganizes federates on the fly based on the various events in real time. So far, HLA-based federates limit their interoperability to military networks, and in syntax-level. Web services techniques are widely used in enterprise applications and provide many proven practices to extend interoperability between WAN resources in semantic level. Two problems are met in order to utilize web services into war-game simulator : 1) How to achieve semantic-level interoperability between federates disperse on WAN, 2) How to interoperate web-based federates and RTI-based federates. In this paper, we provide solutions to the problems and highlight advantages using web-based federates with an example of ASuW(Anti-Surface Warfare).

Key words : Defense modeling and simulation, Semantic web services, Dynamic web services composition

요 약

차세대 워게임 시뮬레이터는 웹상에 분산되어 있는 다양한 자원들을 재사용하고 실시간에 발생하는 각종 이벤트에 따라 동적으로 모델을 재조합하는 기술을 필요로 한다. 기존의 HLA 기반 페더레이트는 군 전용망 내에 있는 다른 페더레이트와의 상호운영성을 보장하기 위해 정해진 문법 수준의(syntax-level) 규칙들을 준수하도록 제한한다. 웹서비스는 비즈니스 영역에서 상호운영성을 보장하기 위해 제시된 기술로 이미 많은 사례를 통해 의미수준(semantic-level)에서 WAN 상의 자원들을 연동시키는 데 사용되어 왔다. 이러한 웹서비스 기술을 워게임 시뮬레이션에 응용하기 위해서는, 1) WAN상에 분산된 페더레이트들을 의미수준에서 상호 운영 할 수 있는 기술 및 2) RTI 기반 페더레이트와 Web 서비스 기반 페더레이트를 상호 운영하기 위한 기술이 제공되어야 한다. 본 논문에서는 상기한 문제들의 해결책을 제공하고, 수상전 예제를 통해 웹기반 페더레이트 사용으로 인한 장점을 보이도록 한다.

주요어 : 국방 모델링 및 시뮬레이션, 시멘틱 웹서비스, 동적 웹서비스 조합

1. 서 론

미래 군대 구조 및 첨단 무기체계가 네트워크 중심(Network-Centric)의 전장 환경으로 진행됨에 따라¹⁾, 다음과 같은 사항을 고려한 워게임 시뮬레이터 구축이 필요

하게 되었다.

- 기존 워게임 시뮬레이션은 보안상 같은 HLA 네트워크에 연결된 페더레이트간의 상호운영성만을 보장하는 매우 제한적인 형태이다. 그러나 다양하고 복잡한 미래 무기체계를 포함한 워게임 시뮬레이터를 효율적으로 구축하기 위해서는 군사 도메인뿐만 아니라 WAN(Wide Area Network)상의 여러 다른 엔터프라이즈 영역에서 개발된 페더레이트를 연결하여 시뮬레이션에 포함시키는 과정이 필요하다.

- HLA에서는 Interface specification, Object Management Template를 준수한 federate간의 문법 수준의 상호

* 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.(UD080042AD)

2008년 11월 11일 접수, 2008년 11월 30일 채택

¹⁾ 명지대학교 컴퓨터공학과

주 저 자 : 김병철

교신저자 : 이강선

E-mail: ksl@mju.ac.kr

운영성을(syntax-level interoperability) 보장하고 있으나^[2], 개발된 페더레이트의 재사용을 극대화하기 위해서는 상호운영성을 시멘틱 수준(semantic-level interoperability)으로 확대할 필요가 있다.

웹서비스(WS : Web Service)는 상호운영성을 보장하기 위해 엔터프라이즈 비즈니스 영역에서 사용하는 기술로, WSDL(Web Service Description Language), UDDI(Universal Description Discovery and Integration)등의 표준화 기술을 통해 구현되며^[5], 이미 많은 상용화된 서비스를 도출하였다. 따라서 웹서비스 기반의 페더레이트를 구축할 경우, WAN상에 널리 분산되어 있는 여러 자원들을 재사용하여 위게임 시뮬레이션 구축에 효과적으로 사용할 수 있는 장점이 있다.

이러한 인식을 기반으로, 웹 서비스 기반 페더레이트를 구축하기 위한 다수의 연구가 있어왔으며^[8], 표준화를 위해 2006년부터 IEEE 1516 HLA standard에 웹서비스 기반 페더레이트 구축을 위한 API(WSDL API)를 정의를 시도하였으며^[6], 현재 HLA Evolved는 IEEE 1516 HLA standard에 Web Services(WSDL) API를 추가한바있다^[12].

그러나 웹서비스 기반의 페더레이트를 기존 위게임 시뮬레이터에 연동할 경우 다음의 문제들에 대한 연구가 필요하다.

- 시멘틱 웹서비스 기술 : 웹서비스에서 WSDL과 UDDI를 이용한 키워드 기반 단순 검색의 한계를 극복하기 위해, 다양한 시멘틱 기반 웹서비스 검색 연구가 진행되었으며, 시멘틱 웹서비스^[5]를 기술하는 대표 언어 OWL-S^[10]를 포함한 여러 언어들이 개발되었다. OWL-S는 웹서비스를 호출하기 위한 WSDL과 시멘틱을 위한 온톨로지를 필요로 한다. 즉, 시멘틱을 제공하기 위해서는 WSDL, 서비스 명세(specification)에 대한 온톨로지, 서비스 설명에 대한 온톨로지가 필요하다. 따라서 시멘틱 웹서비스들의 수가 증가할수록 WSDL, OWL-S, 시멘틱에 관련된 문서들의 수도 증가하며, 이들에 대한 효율적인 관리가 요구된다. 위게임 시뮬레이션은 다양한 무기체계, 전장 환경, 명령 및 지휘 체계에 대한 방대한 시멘틱 명세를 필요로 하므로, 제시한 문제점을 해결할 수 있는 효과적인 온톨로지 구축 방안이 마련되어야 한다.

- 상태정보 유지 기술 : 시뮬레이션 모델들은 시간의 흐름에 따라 상태가 변화한다. 그러나 웹서비스가 갖는 무상태(stateless) 특성으로 인해 웹서비스를 시뮬레이션에 직접 적용할 때 많은 어려움이 발생된다. HLA에서는 로컬 페더레이트들의 상태를 유지하기 위해서 LRC(Local RTI Component)를 사용하고 있으나, 웹 서비스 페더레이트는 웹 서비스의 특성상 상태를 유지 할 수 없기 때문에 이를 극복하기 위한 방법이 요구된다.

본 논문에서는 WSDL2.0과 OWL을 사용하여 웹서비스 기반의 페더레이트에 대한 시멘틱을 기술하며, 웹서비스의 상태정보 유지를 위해 WSPRC(Web Service Provider RTI Component)^[12]를 사용하여 상기한 문제점을 해결하고자 한다.

본 논문은 다음과 같이 구성된다. 2장에서 웹서비스 기반 페더레이트 구축을 위한 온톨로지 및 HLA 기반 페더레이트와의 상호운영 방법론을 소개하고, 3장에서는 수상전(ASuW : Anti-Surface Warfare) 예제를 통해 제안된 방법론의 효율성을 보인다. 4장에서 향후 연구 과제를 도출하고 결론을 맺도록 한다.

본 논문은 다음과 같이 구성된다. 2장에서 웹서비스 기반 페더레이트 구축을 위한 온톨로지 및 HLA 기반 페더레이트와의 상호운영 방법론을 소개하고, 3장에서는 수상전(ASuW : Anti-Surface Warfare) 예제를 통해 제안된 방법론의 효율성을 보인다. 4장에서 향후 연구 과제를 도출하고 결론을 맺도록 한다.

2. 웹서비스 페더레이트 구축

2.1 시멘틱 웹서비스

2007년에 발표된 WSDL 2.0은 WSDL 자체를 RDF/XML(시멘틱을 표현하는 언어)로 맵핑할 수 있도록 하였으며, documentation 부분을 확장하여 웹서비스에 대한 다양한 명세가 가능하도록 하였다^[11]. 이것은 다른 온톨로지 언어(DAML, DAML+OIL, OWL 등)에서 WSDL 내용을 인식할 수 있음을 의미한다.

본 논문에서는 OWL 언어를 이용해 WSDL2.0의 documentation 엘리먼트를 확장하여 시멘틱을 표현하는 방법을 제안한다. OWL(Web Ontology Language)는 대표적인 시멘틱 표기언어 중 하나로 RDF/XML의 표기법을 그대로 사용하면서, 추가적으로 OWL 속성을 사용하여 온톨로지를 표현한다^[4,9]. 그림 1은 OWL의 생성 과정을 도

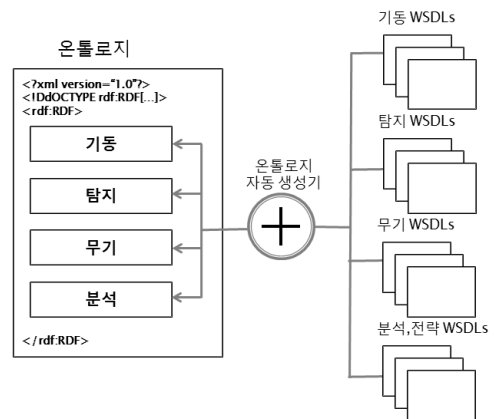


그림 1. 온톨로지 생성 도식화

식화 한 것으로 온톨로지는 각 웹서비스에 대한 WSDL 명세에서 documentation이 포함하고 있는 시멘틱 내용으로 구축됨을 보여준다. 즉, 위게임 시뮬레이터가 기본적으로 수행하는 기동, 탐지, 교전, 분석 단계별로 웹서비스를 조직화 시키고, 페더레이트 개발자는 각 단계의 웹서비스 후보군 중에서 원하는 서비스를 선택하도록 한다. 이때, 선택된 웹서비스의 온톨로지(해당 웹서비스를 명세한 WSDL의 documentation 내용)는 페더레이트의 온톨로지에 복사되어 자동 구축되도록 한다. 본 논문에서 제안하는 시멘틱 웹서비스 온톨로지 구축 방법은 다음과 같은 장점을 갖는다.

그림 1과 같이, 유사한 서비스의 WSDL들을 그룹화함으로써, 기존의 시멘틱 표현 방법(WSDL, OWL-S, OWL 필요)과 비교하여 문서의 간결성과 유지보수성이 개선될 수 있다.

WSDL 내 시멘틱 표현 방법은 RDF/XML으로 변환이 가능하여 WSDL 내 전체의 내용을 온톨로지화 할 수 있다. WSDL의 내용 전부를 온톨로지화 하는 것은 웹서비스의 기술적인 부분 또한 시멘틱 수준(semantic-level)으로 검색이 가능하다는 것이다. 추후에 웹서비스를 조합할 때, IOPE(Input Output Precondition Effect)에 기반의 미 검색을 지원할 수 있다.

온톨로지 구축 과정에 대한 상세한 설명 및 온톨로지를 활용한 모델의 동적 재구성 내용은 3장의 수상전 예제를 통해 살펴본다.

2.2 상태 정보 유지 기법

웹서비스를 시뮬레이션에 적용하기 위해 상태정보를 유지하기 위한 연구들이 진행되고 있으며 대표적으로 다음의 두 가지 방법이 제안되었다^[12].

- 그림 2와 같이 서버에 WSPRC를 두고 원격 페더레이트들의 상태를 유지하기 위한 세션들을 활용하는 방법이 있다. 세션 핸들링을 위해 어플리케이션을 하나 또는 여러 페더레이션에 결합하도록 하고, 여러 웹서비스 페더레이트들은 같은 WSPRC에 연결한다.
- 페더레이션들 사이에 “페더레이션 브릿지”의 역할을 하는 하나의 페더레이트를 사용한다^[8]. 그림 3과 같이 기존의 페더레이션과 웹서비스 페더레이트들을 연결해주는 브릿지를 만들고 웹 서비스 페더레이트의 상태는 브릿지 역할을 하는 페더레이트가 보관하게 된다.

본 논문에서는 전자의 방법을 선택하여, 여러 웹서비스 페더레이트들이 하나의 WSPRC에 연결되도록 하였으며, 웹 서비스 페더레이트들의 동적 조합과 상황에 따른 자동

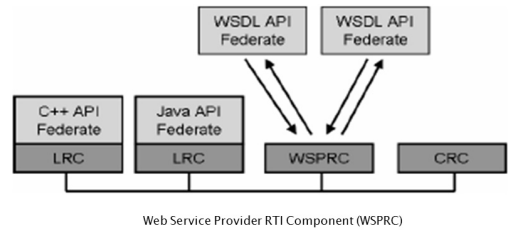


그림 2. 웹서비스 API를 포함한 HLA 구조도

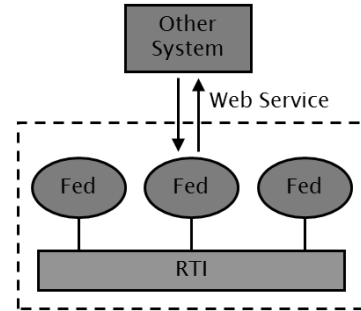


그림 3. 하나의 페더레이트를 브릿지로 사용

판단 및 동적 수행을 위해 온톨로지를 이용하여 구현 하였다.

3. 수상전 위게임 시뮬레이터 제작

그림 4는 구축된 수상전 위게임 시뮬레이터를 도식화 한 것으로 크게 3개의 페더레이트로 구성된다.

- 아군 페더레이트 : 아군함의 기동 및 탐지를 수행하며, 탐지된 상황을 구축된 온톨로지에 적용하여 현재 상황에서 가장 적합한 무기를 선택한 후, 교전 및 분석을 수행한다.
- 적군 페더레이트 : 적함의 기동, 탐지, 교전, 분석을 수행하며, 아군 페더레이트와 같은 논리 아래 수행된다.
- Command/Control 페더레이트 : 아군 및 적군의 페더레이트를 모니터링하고 현재 선택된 무기·엔진·탐지 및 식별기 웹서비스를 보여준다.

이상 3개의 페더레이트는 분산 환경에서 작동한다. 군함 페더레이트는 웹서비스와 RTI와의 연동을 위한 WSPRC를 포함한다.

3.1 군함 온톨로지 제작 과정

실제 군함은 군함 프레임에 레이더, 무기, 적 식별기, 엔진 등을 장착하고 있다. 본 연구에서는 가상의 군함을

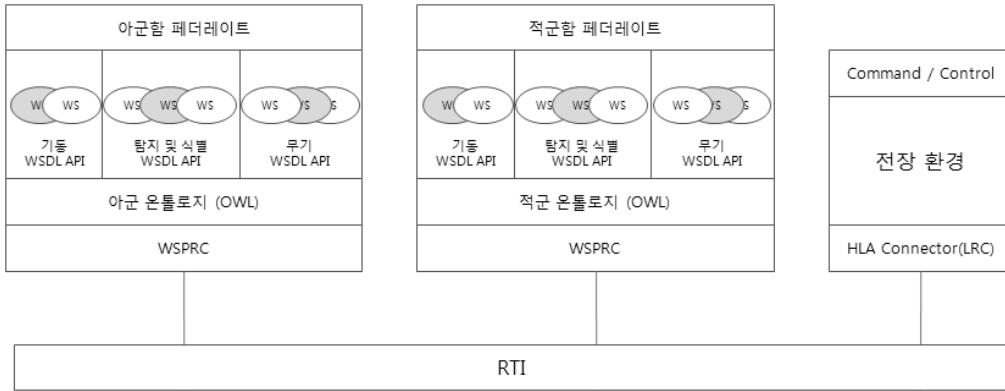


그림 4. 위게임 시뮬레이터 구조도

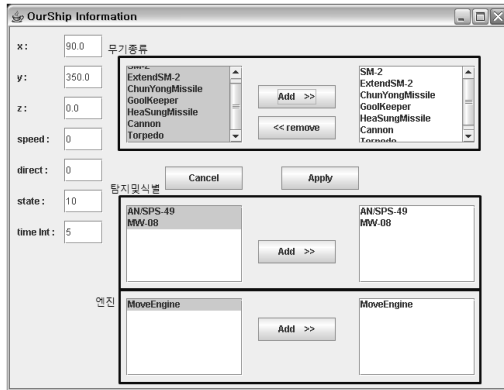


그림 5. 아함 페더레이트 구성 과정

제작하기 위해 실제 군함을 구축하는 방식으로 군함 온톨로지 틀에 그림 5와 같이 무기, 기동 엔진, 탐지 및 식별기를 선택하여 군함 온톨로지를 자동 생성하도록 한다. 그림 6은 군함 설계 시, 사용자의 무기 선택에 따라 동적으로 자동 생성된 온톨로지(OWL)의 예이다. ship:weaponsOfShip rdf:ID="AegisShip"은 AegisShip이 가지고 있는 무기들의 종류를 나타내고, ship:KindOfWeapons rdf:ID="SM-2"는 무기 모델명 SM-2에 대한 속성을 나타낸다.

그림 7은 WSDL2.0의 documentation 엘리먼트 내에 OWL로 작성된 미사일 웹서비스의 시맨틱 정보를 나타내며 1) 천용은 미사일의 한 종류이며, 2) 최소 사거리는 500km이고, 3) 최대 사거리는 1,000km, 4) 미사일 속도는 1,260km/h, 5) 목표물 적중에 대한 정확도는 95%임을 의미한다.

3.2 시뮬레이션 수행

본 연구에서 교전 절차는 1) 기동, 2) 탐지 및 적 식별,

```

<owl:ObjectProperty rdf:ID="hasWeapons">
<rdfs:range rdf:resource="#WeaponsOfAegis" />
</owl:ObjectProperty>

<ship:WeaponsOfAegis rdf:ID="AegisShip">
<ship:weaponOfShip rdf:resource="#SM2" />
<ship:weaponOfShip rdf:resource="#ExtendSM2" />
<ship:weaponOfShip rdf:resource="#ChunYongMissile" />
<ship:weaponOfShip rdf:resource="#GoalKeeper" />
<ship:weaponOfShip rdf:resource="#HeaSungMissile" />
<ship:weaponOfShip rdf:resource="#Cannon" />
<ship:weaponOfShip rdf:resource="#TOrpedo" />
</ship:WeaponsOfAegis>

<ship:KindOfWeapon rdf:ID="SM2">
<ship:missileType rdf:datatype="http://www.w3.org/2001/XMLSchema#String">
Missile</ship:missileType>
<ship:minRange rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
74</ship:minRange>
<ship:maxRange rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
167</ship:maxRange>
<ship:speedOfWeapon rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
9600</ship:speedOfWeapon>
<ship:successfulRate rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
85</ship:successfulRate>
<ship:countOfWeapon rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
30</ship:countOfWeapon>
</ship:KindOfWeapon>
    
```

그림 6. 자동생성된 아함 온톨로지의 무기부분

3) 적에 따른 무기 선택, 4) 교전, 5) 손상정도 파악 순으로 반복 진행된다. 각 절차마다 사용자로부터 군함 온톨로지에 등록된 웹서비스들이 실행된다. 적에 따른 무기 선택 절차에서는 적 식별과 거리에 따른 정보를 고려하여 무기를 선택하고 적이 파괴될 때까지 각 절차가 반복 수행된다. 그림 6과 같이 온톨로지 군함에 SM-2, ExtendedSM-2, ChunYong, GoalKeeper, HeaSung, Cannon, Torpedo가 장착되어 있는 경우, 교전 절차는 다음과 같다. 먼저, 탐지 및 식별 단계에서 적이 구별되고, 거리를 계산한다. 다음 단계로 온톨로지 질의를 통해 상황에 따른 적절한 무기가 선택된다.

```

<?xml version="1.0" ?>
<description xmlns= .... >
<documentation>
  <xsd:schema xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:ship="http://selab.mju.ac.kr/ship/missile/schema#">
    <ship:KindOfWeapon rdf:ID="ChunYongMissile">
    <ship:missileType rdf:datatype="http://www.w3.org/2001/XMLSchema#String">
      Missile</ship:missileType>
    <ship:minRange rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
      500</ship:minRange>
    <ship:maxRange rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
      1000</ship:maxRange>
    <ship:speedOfWeapon rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
      1260</ship:speedOfWeapon>
    <ship:successfulRate rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
      95</ship:successfulRate>
    </ship:KindOfWeapon>
  </xsd:schema>
</documentation>
<interface name="ChunYongMissileInterface">
<binding name="ChunYongMissileInterfaceHttpBinding"
  interface="tns:ChunYongMissileInterface"
  type="http://www.w3.org/ns/wsdl/http"
  wssoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">
<service name="ChunYongMissile" interface="tns:ChunYongMissileInterface">
</description>
  
```

그림 7. 시멘틱을 포함한 WSDL의 예

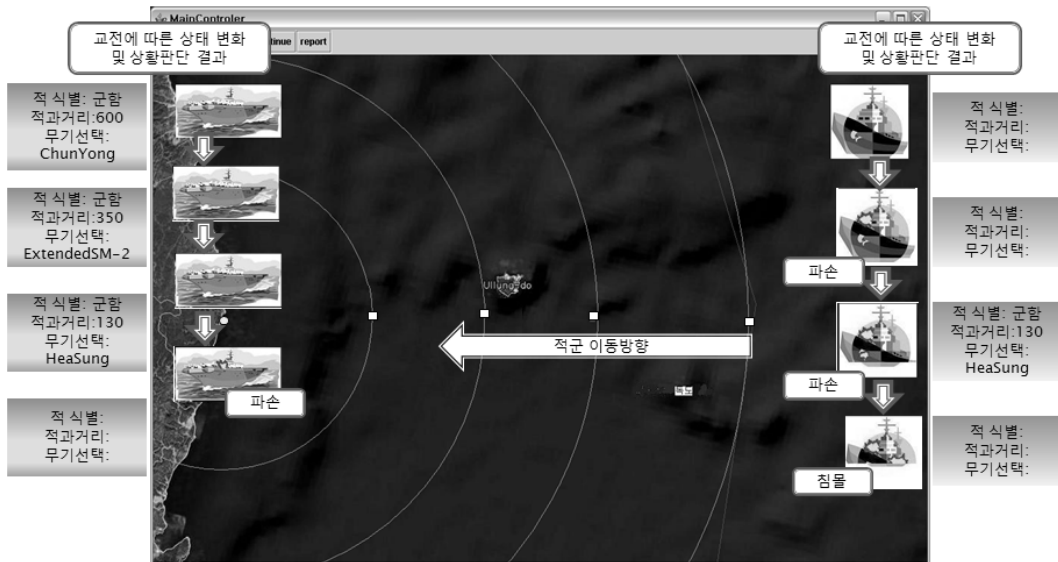


그림 8. 모니터링 페더레이트 화면에서 본 교전 수행결과

예를 들어, 적이 전함인 경우 전함을 공격할 수 있는 무기는 미사일, 포 등이며 적함의 종류와 거리를 고려하여 미사일이 선택된다. 위에서 제시한 질의는 적절한 무기를 선택하기 위해 사용한 질의의 구문과 예를 나타낸다. 그림 8은 수상전의 예제에서 교전 중 초기단계에서는

ChunYong이, 초기 파손단계에서는 ExtendedSM-2이, 후기 파손단계에서는 HeaSung 미사일이 무기로 선택된 결과를 보이고 있다.

그림 8은 본 논문에서 구현한 모니터링 페더레이트 화면에서의 교전 수행결과를 나타낸 것이다.

4. 결론과 평가

웹 서비스는 위게임 시뮬레이터 구축 시 웹상에서 상용화된 여러 자원을 재사용하고 WAN 환경에 분산된 여러 페더레이트간의 모의 연동을 지원할 수 있는 매우 효과적인 기술로 인식되고 있다. 본 논문에서는 웹서비스 기반 페더레이트와 기존의 RTI 기반 페더레이트 간의 연동 시 의미 수준의 상호운영성(semantic-level interoperability)을 달성하기 위한 시멘틱 명세 방법을 제안하고 함대함 수상전 예제를 통해 군함을 이루는 엔진, 탐색 및 식별 장비, 무기를 위한 온톨로지를 구축하였다. 구축된 온톨로지는 교전 결과 분석에 활용되어, 아함 및 적함에 서 현재 상태에 맞는 최상의 무기를 동적으로 선택한 후 대응되는 웹서비스를 자동 실행시킴으로써 시멘틱 수준의 plug-and-play 가능성을 보여주었다.

미래 네트워크 기반 LVC(Live, Virtual and Constructive) 시뮬레이터 구현을 위해서는 다양한 이벤트 및 방대한 실시간 데이터를 분석하여 동적으로 페더레이트를 재구성하는 기법이 필요하다. 웹서비스는 기본적으로 요청/응답 방식으로 수행되기 때문에, 이벤트 발생 여부를 판단하기 위해서는 지속적인 감시가 이루어져야한다는 문제가 있다. 향후에는 이러한 문제점을 개선하기 위해 웹 서비스 이벤트 기술을 적용하여 이벤트 발생을 통지(notification)함으로써, 지속적 감시로 인한 overhead를 해결할 계획이다.

참 고 문 헌

1. 최인수. (2007), “NCW환경에서의 국방정보보호 발전방향”,

정보화학회지, 제25권 9호, pp. 81-88.
 2. Booth, D. et. al. (2004), Web Services Architecture, W3C Working Group Note, <http://www.w3.org/TR/ws-arch/>
 3. Katherine L, Morse PD, Drake DL, Brunton RPZ. (2003), “Web enabling an RTI - an XMSF Profile[C]”, EURO Simulation Interoperability Workshop, Stockholm, Sweden : Simulation Interoperability Standards Organization, 03E-SIW-046.
 4. Lee W., Lacy. (2005), OWL:Representing Information Using the Web Ontology Language, Trafford Publishing.
 5. McIlraith, S.A. et al. (2001), “Semantic Web Services”, IEEE Intelligent Systems, Vol. 16, No. 2, pp. 46-53.
 6. Möller B. (2006), “A First Look at the HLA Evolved Web Service API”, Simulation Interoperability Standards Organization, 06E-SIW-061.
 7. Möller B, Löf S. (2005), “Mixing Service Oriented and High Level Architectures in Support of the GIG[C]”, Spring Simulation Interoperability Workshop. San Diego, California : Simulation Interoperability Standards Organization, 05S-SIW-064.
 8. Möller B, Löf S. (2006), “A Management Overview of the HLA Evolved Web Service API”, Simulation Interoperability Standards Organization, 06F-SIW-024.
 9. OWL Web Ontology Language Overview, (2004), <http://www.w3.org/TR/owl-features/>
 10. W3C Member. (2004), OWL-S:Semantic Markup for Web Services, <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>
 11. Web Services Description Language (WSDL) Version 2.0: RDF Mapping, (2007), <http://www.w3.org/TR/wsdl20-rdf/>
 12. Wenguang WANG. (2008), “Service-Oriented High Level Architecture”, Simulation Interoperability Standards Organization, 08E-SIW-022.



김 병 철 (blue7928@mju.ac.kr)

2007 명지대학교 컴퓨터공학과 학사
현재 명지대학교 컴퓨터공학과 석사과정

관심분야 : 컴퓨터시뮬레이션, 웹서비스, 온톨로지



이 강 선 (ksl@mju.ac.kr)

1992 이화여자대학교 전자계산학과 학사
1994 이화여자대학교 전자계산학과 석사
1998 미) University of Florida, 박사
현재 명지대학교 컴퓨터공학과 부교수

관심분야 : 컴퓨터시뮬레이션, 국방 M&S, 웹서비스