

지능형 교통 시스템을 위한 동적 정보관리 시물레이션

조규철[†]

Dynamic Interest Management in Web Simulation for Intelligent Transportation System

Kyu-Cheol Cho[†]

ABSTRACT

Simulation methods are being studied for various service models that can be used in industries as the development of web technology. DEVS is being used as a tool to simulate through scenarios using various information. In addition, DEVS/WS integrates web-based DEVS models through a distributed computing environment and can be used as tools for high-performance computing and data distribution. This paper describes a method for simulating an intelligent transportation system in DEVS/WS environment. This paper propose dynamic interest management model(DIMM) applied the genetic algorithm, manage efficiently road information of moving node. And, this paper evaluates performance of the dynamic interest management model in comparing to the none interest management model(NIMM). The transmitted messages and simulation time of the DIMM is saved than that of the NIMM.

Key words : DEVS, Simulation, Web Service, ITS

요약

웹 환경 기반 모델링 기술의 발전으로 다양한 산업분야에서 활용할 수 있는 서비스 모델에 대한 시물레이션 연구가 진행되고 있다. DEVS 형식론은 시간의 흐름에 따라 다양한 정보를 사용하여 시나리오를 통해 시물레이션 할 수 있는 도구로써 활용되고 있다. 또한 DEVS/WS는 웹 기반의 DEVS모델들을 분산컴퓨팅환경을 통해 통합하여, 고성능 컴퓨팅과 데이터 분산 등이 요구되는 시물레이션 환경으로 활용이 가능하다. 본 연구에서는 이동 노드 간 공간에 대한 정보를 효율적으로 관리하기 위해 Dynamic Interest Management Model(DIMM)를 제안하여 DEVS/WS 환경에서 시물레이션 하였다. DIMM에서 시물레이션 중 노드의 효율적인 위치 이동을 위해 Genetic 알고리즘을 통하여 최적의 경로를 유도하게 된다. 본 연구에서는 DIMM의 성능을 평가하기 위해 None Interest Management와 전송메시지 수, 시물레이션 시간을 비교하였다. DIMM은 공간정보에 대한 인지도관리를 통해 메시지의 수와 시물레이션 시간을 절약하는 성능을 제공하였다.

주요어 : DEVS, 시물레이션, 웹 서비스, 지능형 교통체계시스템

1. 서론

웹 환경 기반의 시물레이션 기술은 시물레이션 모델의

개발, 문서화, 분석을 지원하며, 시물레이션 분야에 의미 있는 정보를 만들기 위해 중요한 역할을 해왔다. 웹기반 시물레이션은 인터넷을 통해 전통적인 시물레이션 모델의 연결에 대한 한계를 극복하였고 시물레이션의 접근성과 제어, 효과적인 관리 및 통합을 용이하게 하고 하드웨어와 시스템에 독립적이라는 장점이 있다^[1].

DEVS/WS(Discrete Event System Specification/Web Service)^[2]는 DEVS^[3]를 확장하여 분산되어 있는 컴퓨팅 자원을 활용하여 고성능의 웹 서비스를 시물레이션 할 수 있는 도구이다. 서비스 모델을 웹 서비스로 정의하고

* This work was supported by INHA TECHNICAL COLLEGE Research Grant.

Received: 23 May 2018, Revised: 26 November 2018,
Accepted: 17 January 2019

[†] Corresponding Author: Kyu Cheol Cho

E-mail: kccho@inhac.ac.kr

Dept. of Computer Science, Inha Technical College

서비스 참여 모델을 DEVS 모델을 활용함으로써 단위 모델에 대한 메시지 송수신과 모델 간의 네트워크 통신을 통해 상호작용을 통해 고성능 계산, 데이터 분산, 서비스 분산을 시뮬레이션 할 수 있다.

산업발전과 더불어 제한적인 도로에서 지속적으로 차량이 증가하고 있고 효율적인 교통 흐름을 유도하기 위한 대안으로 지능형 교통체계(ITS: Intelligent Transportation System)^[4]와 다양한 정보를 제공하는 연구가 진행되고 있다. ITS는 교통시설의 이용 효율의 극대화와 교통 이용편의를 고려하고 있다. 본 연구에서는 복잡한 도로환경에서 이동 중인 노드들이 효율적으로 움직이기 위해 도로상에서 목적지가 있는 노드의 최적 경로를 이용한 이동을 제공하기 위해 동적 인지도 관리방법을 제안한다. 교통상황에 대한 시나리오와 수많은 이동차량들이 생산하는 데이터를 계산하여 최적의 경로를 제공하기 위해서는 고성능의 컴퓨팅 자원과 차량들의 다양한 서비스를 시뮬레이션 할 수 있는 환경이 필요하다.

본 연구 구성은 다음과 같다. 2장에서는 본 연구의 배경이 되는 웹 서비스기반 시뮬레이션에 대하여 기술하고, 3장에서는 웹 서비스 시뮬레이션을 위한 DEVS/WS를 소개한다. 4장에서는 DEVS/WS를 활용한 최적경로 이동을 위한 시뮬레이션을 설명한다. 5장에서는 실험과 결과를 논하고, 마지막으로 결론을 맺는다.

2. 관련연구

웹 서비스 기반 시뮬레이션은 다양한 서비스를 제공하고 최적의 서비스를 제공하기 위한 분석을 위한 목적으로 연구되고 있다. Simjava^[5]는 웹 환경에서 기본적인 이산사건 시뮬레이션 엔진과 자바를 이용하여 시뮬레이션 하였고 자바를 이용하여 멀티스레드를 이용한 분산 시뮬레이션 하였다. 그리고 하이브리드 서버/클라이언트 시뮬레이션은 서버 호스트와 클라이언트 실행 시뮬레이션간의 장점을 결합하려는 시도를 하였다^[6]. 지리적으로 분산된 모델들을 에이전트 기반으로 시뮬레이션 흐름 관리를 통해 웹기반의 시스템에 의사결정을 지원하였다^[7]. 그러나 웹 기반 시뮬레이션은 서버와 클라이언트들의 각각의 시뮬레이션 동기화 및 통합 그리고 컴포넌트들 간의 동기화와 상호작용의 어려움이 있다.

3. 웹 서비스 시뮬레이션을 위한 DEVS/WS

DEVS/WS에서 실행단위로써 웹 서비스 시뮬레이션

환경은 시뮬레이션 표현 영역과 메시지 통신영역으로 나뉜다. 시뮬레이션 표현 영역은 참여모델에게 시뮬레이션 진행 정보를 출력하고 메시지 통신 영역은 DEVS/WS에서 참여모델간의 외부 메시지 및 정보 송수신을 위한 영역으로 Time Server(TS)와 정보 송수신 및 참여객체들과 상호작용을 위한 메시지를 전달할 수 있는 영역이다.

3.1 DEVS/WS 시뮬레이션 관제

DEVS/WS 시뮬레이션 관리자는 DEVS 참여모델들을 통합하여 전반적인 시뮬레이션 운영을 관장하는 프로세스이다.

- *시뮬레이션 관제*: DEVS/WS에서 실행되고 있는 시뮬레이션 시간의 동기화 및 참여객체 관리를 진행하고 시간을 제어하며 순차적인 시뮬레이션을 진행할 수 있다. 그리고 참여 참여객체의 웹 서비스 참여 정보를 확인하고 참여 참여객체를 참여 여부를 확인할 수 있다.
- *DEVS 모델의 시뮬레이션 참여*: 시뮬레이션에 참여하게 되는 DEVS/WS 참여객체를 관리정보 및 연결 정보를 관리한다.
- *시뮬레이션의 실행과 운영*: 시뮬레이션에 참여한 모든 참여객체가 확인되면 참여객체들 간의 상호 운용 및 상호작용을 지원하고, 참여객체들 간의 연결 정보를 전달하여 상호작용의 활성화를 유도하고 시간을 통제하여 시뮬레이션의 흐름을 제어할 수 있다.
- *시뮬레이션 시간관리*: 시뮬레이션 시간 동기화구조를 통해 시뮬레이션 시간을 관리하여 병렬처리 뿐만 아니라 분산 환경 시뮬레이션이 가능하도록 지원한다.

3.2 DEVS/WS 실행 구조

DEVS/WS는 DEVS/WS 시뮬레이션 관리자를 통해 웹 서비스를 이용하려는 참여객체들로 하나의 시뮬레이션 단위가 형성되는데, 그 참여객체들의 실행구조는 그림 1과 같이 표현할 수 있다.

DEVS/WS 모델링단위가 되는 DEVS 참여객체들은 시뮬레이션 객체 생성과 초기화를 통해 참여 객체로써 실행 이후에 각자의 역할을 수행하게 된다. 시뮬레이션을 진행을 위해서는 시뮬레이션 단위의 모든 참여객체들을 연결하여 웹 서비스 형태로써 상호간의 연결을 통하여 웹 서비스 시뮬레이션 실행구조를 구성할 수 있다.

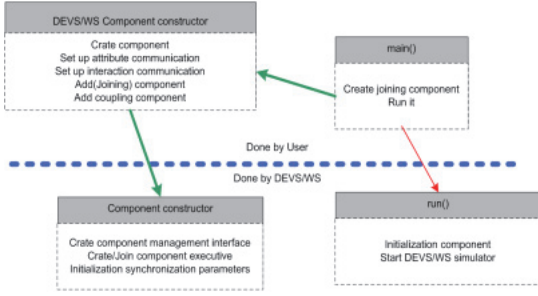


Fig. 1. Simulation model connection and run in DEVS/WS

3.3 DEVS/WS에서의 웹 서비스 시뮬레이션 환경

그림 2는 웹 서비스의 참여객체로서 약속된 메시지 형식으로 전송할 데이터를 XML 메시지 생성기(Generator)를 통해서 만들어 내고, 다른 웹 서비스 참여객체로 부터 데이터를 수신할 때 XML로 수신하여 XML 분석기(Parser)를 통해 필요한 데이터를 추출하게 된다.

DEVS/WS는 서비스 객체 검색정보들을 정의하는 서비스 질의, XML 파일을 이용하여 웹 서비스 참여객체들 간의 정보를 송수신하는 메시지를 정의하는 서비스 간 메시지가 있다. 웹 서비스 형태를 운영하기 위한 운영 메소드를 정의하고, XML로 표현되는 인자 메소드를 정의함으로써 웹 서비스 정보를 정의하게 된다.

4. DEVS/WS를 활용한 최적경로 이동 시뮬레이션

DEVS/WS는 웹 서비스에 대한 시뮬레이션 환경을 지원한다. DEVS/WS로 웹서비스 참여객체는 각각의 참여객체는 인터넷상의 다른 참여객체와 상호 작용을 통해 작업, 정보, 자원을 공유 할 수 있다. 이때, 참여객체는 상

호 작용하는 명령, 메시지, 데이터 수신 등을 정보를 송수신 할 수 있다. 최근 차량의 이동을 위한 도로, ITS, 교통 정보 등의 인프라의 발전으로 차량 이동 간 많은 정보를 활용하여 최적의 경로를 제공하는 시스템에 대한 요구뿐만 아니라 해당 시스템의 성능향상을 위한 노력을 기울이고 있다^{[8][9]}.

4.1 참여객체 간 정보공유를 위한 프로토콜

DEVS/WS에 참여한 웹 서비스 구성요소들인 차량정보와 교통상황 정보를 통해 이동 중인 노드들에게 최적 경로를 제안하는 기지국(Base station)과 이동하는 노드들은 역할공유, 참여객체 검색, 객체간의 연결 등 3가지 상호작용이 필요하다. 이 3가지 상호작용은 웹 서비스 환경과 독립적으로 통신하는데, DEVS/WS은 분산 컴퓨팅 환경이 지원되고 있고 약속된 방법의 통신규약을 통하여 독립적인 환경에서 일관된 방법으로 공개, 검색이 가능하다.

4.2 DEVS 웹서비스 모델의 시뮬레이션 참여

DEVS/WS의 참여객체는 서비스를 요구하는 역할을 수행하는 참여객체 단위로서의 시뮬레이션 객체이다. 기지국, 이동 노드들은 DEVS/WS의 시뮬레이션 관리자라 통해서 시뮬레이션에 참여할 수 있고 참여객체로서 이벤트를 전달하는 것뿐만 아니라 다른 참여객체들 간의 상호작용을 통해서 분산된 컴퓨팅자원을 활용하여 병렬 및 분산 컴퓨팅 시뮬레이션이 가능하다.

- ① 웹 서비스 사용자로서 시뮬레이션 참가: DEVS/WS에서 최적경로 이동 시뮬레이션에 참여하는 모든 참여객체들은 서비스 사용자로서 자신의 접속정보를 등록함으로써 다른 참여객체와 상호작용 및 정보공유가 가능하다. 그렇기 때문에 참여객체들은

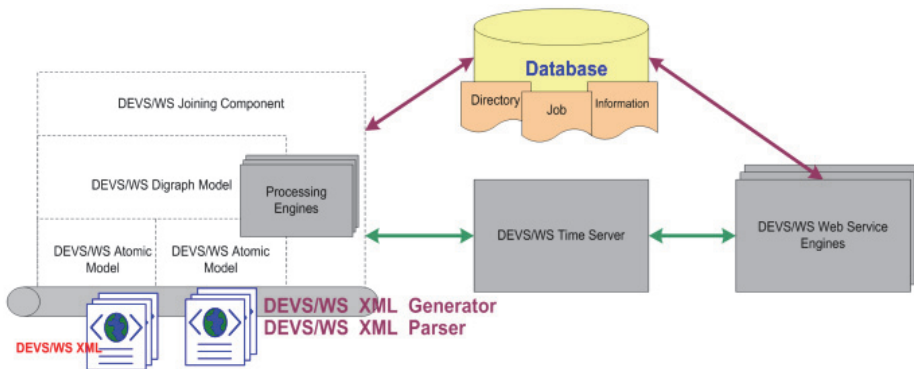


Fig. 2. DEVS/WS component modeling and communication structure of web service

자신의 정보를 자세하게 설명하고 설정할 필요가 있다.

```
<param name="ComponentIndex" value="0">
<param name="DigraphClassName" value="DEVS/WS-BaseStation">
<param name="ListeningPort" value="1973">
<param name="TimeServerIP" value="165.246.44.002">

<param name="ComponentIndex" value="1">
<param name="DigraphClassName" value="DEVS/WS-Node#1">
<param name="ListeningPort" value="1974">
<param name="TimeServerIP" value="165.246.44.002">
```

Fig. 3. Self description of the DEVS/WS components

그림 3은 참여객체의 Digraph Class Name이 “DEVS/WS-BaseStation”, “DEVS/WS-Node#N”인 참여객체의 참여 정보를 나타낸 것으로, 시뮬레이션에 참여하는 참여객체의 순번(Component Index)은 0, 1이다. Listening Port는 다른 참여객체와 정보공유와 상호작용을 위한 연결할 포트를 의미하고, Time Server IP는 시뮬레이션을 관장하는 관리자의 IP 주소를 표현한 것이다. 환경설정에 대한 최적경로 이동 시뮬레이션을 위한 참여객체들의 구성 및 연결모델^[10]을 표현하면 그림 4와 같이 표현할 수 있다.

② DEVS/WS 시뮬레이션 관리자로부터 웹서비스 참여자 정보 수신: 웹 서비스에 참여한 모든 참여객

$$TM_{sim} = \langle IX, IY, EX, EY, D, \{Md | d \in D\}, d \in D, EIC, EOC, IIC, IOC, IC, EC \rangle,$$

TM_{sim} : a coupled simulation model of the Traffic management

IX : the set of internal input ports and values

IY : the set of internal output ports and values

EX : the set of external input ports and values

EY : the set of external output ports and values

D : the set of the component

EIC : External Input Coupling - external inputs to component inputs

EOC : External Output Coupling - outputs to component inputs

IIC : Internal Input Coupling - internal inputs to component inputs

IOC : Internal Output Coupling - internal outputs to component outputs

IC : Internal Coupling

EC : External Coupling

where

$Internal\ Inport = \{ "start" \}, External\ Inport = \{ "ESTART" \},$

$IX = \{ ("start", start) \}, EX = \{ ("ESTART", start) \}$

$Internal\ Outport = \{ "out" \}, External\ Outport = \{ Auto\ allocation \}$

$IY = \{ ("out", (optimal\ path, threshold\ time, expected\ moving\ time, traffic\ information)) \}$

$EY = \{ ("result_out", (optimal\ path, expected\ moving\ time, traffic\ information)) \}$

$D = \{ base_station, node\#1, \dots, traffic_agent \}$

$EIC = \{ (TM_{sim}, "start"), (base_station, "start") \}$

$EOC = \{ (TM_{sim}, "out"), (TM_{sim}, "finish") \}$

$EC = \{ ((base_station, "threshold_time_out"), (agent, "threshold_time_in"), \dots \}$

Fig. 4. Simulation connection model

체는 객체간의 상호작용과 정보 공유 및 데이터 전달을 위해 상대의 참여객체들의 위치를 알아야 하는데, 웹서비스 참여 참여객체들은 정보 및 작업을 전달하고 수행하는 상대 참여객체들의 정보를 DEVS/WS 시뮬레이션 관리자로 부터 정보를 수신할 수 있다. 이때 DEVS/WS에서 DEVS 모델들에게 정보를 전달하는 것은 참여객체가 참여할 때 서비스 모델들과 공유할 참여객체 정보들이다. DEVS/WS는 참여 웹서비스 모델자신의 위치를 노출하지 않더라도 다른 참여객체와 투명성 있게 상호작용할 수 있는 장점이 있다.

4.3 동적 인지도관리

인지도관리(Interest Management)는 화산지역 관리, 교통사고 위험 지역 관리, 통계적 그리드 자원관리 등에서 송수신 메시지나 많은 데이터양에서 일부 데이터나 불필요한 데이터를 여과하여 실시간의 대규모의 시뮬레이션의 콘텐츠를 활용하는 것이다¹⁷⁾.

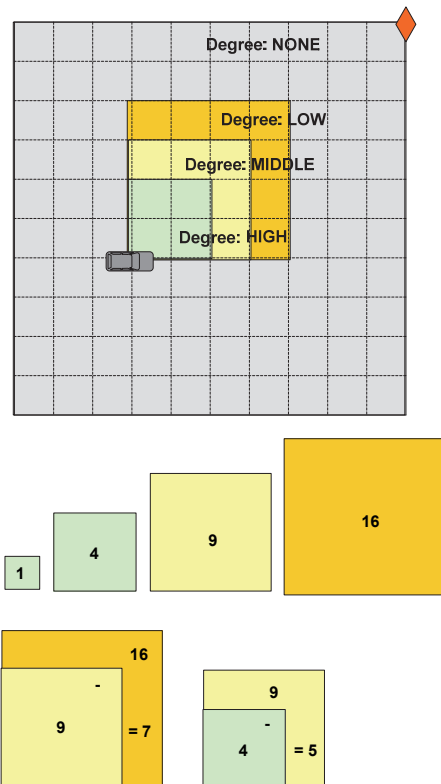


Fig. 5. Interest cell degree space management

다음 그림 5는 일정한 거리를 기준으로 인지도 셀을 상세하게 분류한 것이다. 특정 셀 단위로 구성된 공간에 대한 인지도가 시간, 상황, 환경 등의 요소에 따라서 인지도 공간의 크기나 모양이 변경되는 것을 관리하는 것이 인지도 관리이다. 인지도 관리기술은 인지도 표현이나 인지도 도메인을 이용하여 관리하는데, 보통 지역적으로 넓은 범위의 공간을 다차원으로 표현하여 지역단위의 데이터 전송 관리를 위해서 사용된다. 특히 시뮬레이션 중에 인지도 도메인을 상세하게 관리되어야 하는 경우는 인지도 수치를 활용하여 관리한다.

ITS에서 기지국은 복잡한 도로로 지나가는 차량을 위해 교통상황을 잘 반영하고 정확한 정보를 지속적으로 갱신해야 한다. 그런데 교통상황을 지속적으로 갱신하기 위해서 기지국은 교통상황의 정보를 빈번하게 수집하고 전달해야하기 때문에 정확하고 원활한 정보를 만들기 위해서 많은 메시지와 전송데이터가 요구된다. 그런데 기지국에 매 시간, 모든 상황마다 많은 정보를 전송하여 갱신하는 것은 비효율적이기 때문에 시간과 교통의 상황을 고려하고 교통상황을 갱신하여 메시지를 줄일 필요가 있다.

실시간으로 교통정보를 전송하는 경우, 기지국에 교통정보를 전달함에 있어서 실시간으로 교통정보를 메시지 화하여 바로 네트워크로 전송하고 적용해야 하므로 현실성이 없다. 그리고 차량의 이동성과 도로의 차량이동을 위한 통과 한계가 있기 때문에 교통정보의 실시간 전송은 짧은 시간간격으로 정보를 전송하는 것이 교통상황에 큰 변화가 없다면 비효율적이게 된다. 반면에 교통상황 정보를 전송하는 시간간격이 클 경우 도로의 교통상황에 대한 정확도 오차가 크기 때문에 정확한 교통정보를 제공할 수 없게 된다. 그리고 차량의 흐름이 많아서 교통상황이 빈번하게 바뀌는 경우는 교통상황을 자주 갱신하여 해당 구간의 교통정보를 정확하게 할 필요가 있으며, 차량 흐름이 드문 경로에서는 정보 갱신 횟수를 줄이더라도 교통상황 정보에 대한 오차와는 상관관계가 적을 것이다.

주행 중인 차량은 도로상에서 차량의 목표지점까지 도달하기 위해서 차량의 경유 지점을 포함한 주위 도로의 교통상황에 대한 교통정보를 바탕으로 주행하게 된다. 이때 차량은 목표지점까지 주행하기 위한 경유 예약지역이 아닌 곳의 교통정보나 이미 지나간 지역의 교통정보는 앞으로 목표지점까지 주행하기 위한 경로설정에 거의 영향을 주지 않는다.

그렇기 때문에 차량은 경유 예약지역이 아닌 곳, 현재 지점에서 멀리 떨어진 곳, 이미 지나간 경로에 대한 교통

정보를 자주 수집하는 것은 정보를 낭비하게 됨을 의미하기 때문에 이러한 정보를 절약하여 최적의 경로로 목표지점까지 이동함에 있어 이동시간의 변화의 오차가 적다면 정보의 송수신함에 있어서 메시지의 수의 절약과 메시지 양을 절약하게 될 것이다.

시뮬레이션 모델은 차량 이동이 많은 도심의 차량 정체, 도로상태, 교통 혼잡 제어 등을 목적으로 현재의 교통 상황을 표현하는 최적경로 이동모델이다. 그림6은 교통 혼잡 제어를 위한 도로모델을 디자인한 것으로 도심의 블록단위의 도로 공간을 하나씩 나누어 단위 맵 상에서 주행 중인 차량상태를 노드로 표현하여 노드들의 현재 위치 및 속도 등을 통하여 교통상황을 모델링할 수 있다. 전체의 도로에서 주행 중인 모든 노드들을 표현하는 것은 표현하는 데이터뿐만 아니라 최적경로로 차량 이동을 위한 의사결정 하는 계산량도 요구되기 때문에 분산화하고 관리하기 용이하게 적당한 구역을 정하여 단위맵을 기준으로 웹 서비스 참여객체 단위로 나누게 된다. 이는

인접한 4개의 방향인 맵 정보를 업데이트하며 상호작용하게 된다.

도로에서 하나의 블록을 하나의 모델로 구분해 M_{ij} 로 표현할 수 있고, 시간에 따라서 노드는 도로를 따라 이동할 것이다. 블록을 지형의 방향을 x(가로),y(세로)로 나누고 시간(t)에 따라 노드가 움직이는 것을 표현한 것으로 $M_{00}(t_{start}) \rightarrow M_{10}(t_0) \rightarrow M_{11}(t_1) \rightarrow M_{12}(t_2) \rightarrow M_{22}(t_3)$ 순으로 이동하고 있다.

그림 7은 최적경로를 제공하기 위한 시뮬레이션 구성도를 도식화한 것으로 웹 환경에서의 메시지 전송을 위한 아키텍처는 3단계로 구성되어 있다. 시뮬레이션 참여객체는 도로상황을 갱신하고 이동 중인 노드로서 교통상황에 대한 정보를 필요로 한다. 교통정보 등의 상호작용 메시지는 네트워크 및 DEVS 내부 메시지를 통해 기지국과 송수신하고 기지국은 현재의 교통정보를 유지하기 위한 데이터가 수신되면 교통 혼잡에 관한 현 상황 정보를 적용하여 교통정보 요청 시 해당 데이터를 참여모델에게

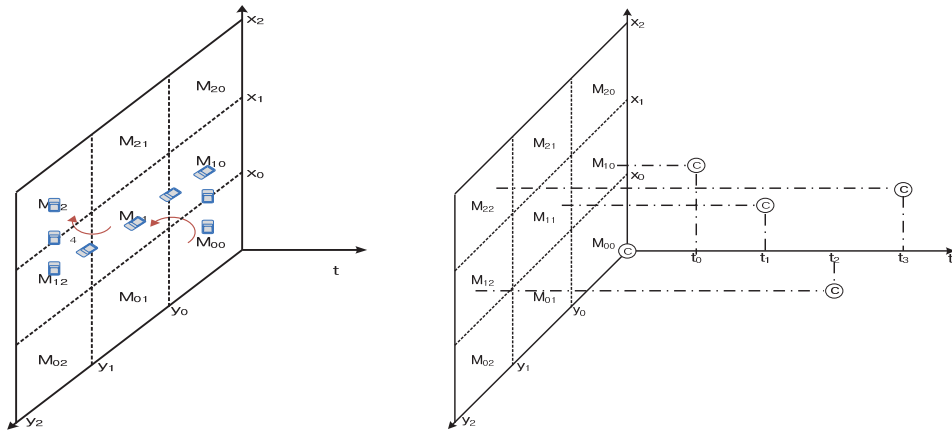


Fig. 6. Map distribution and the example of node moving in DEVS/WS

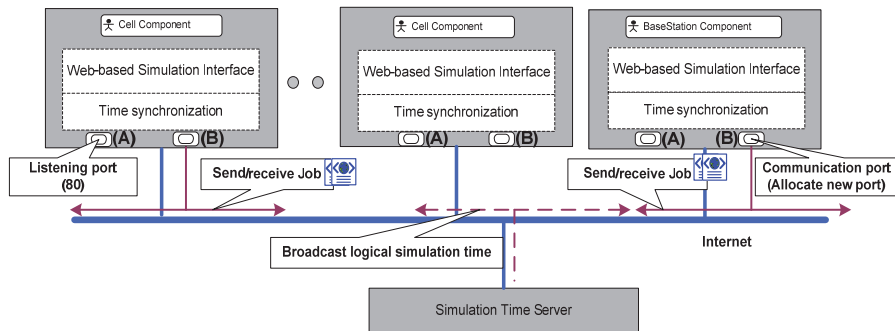


Fig. 7. Simulation model of the optimal path information

전송하게 된다. 참여모델은 최적 경로 정보 요청 시 최적 경로에 대한 의사결정을 담당하는 전문가 모듈을 통해 현 상황의 교통정보를 기준으로 최적의 경로를 도출하여 서비스를 제공받는 참여모델에게 제공한다.

5. 실험 및 결과

본 연구에서 Dynamic Interest Management Model (DIMM)이 복잡한 도로상에서 이동 중 노드에게 기지국이 현재의 교통상황을 고려하여 최종목적지까지 최적의 이동경로를 추천하기 위해 Genetic 알고리즘을 적용하였다^[11]. 그리고 현재의 교통상황의 원활한 정보를 갱신을 위해 인지도관리 셀을 통해 최소의 메시지 수와 데이터 양으로 주요 지점의 교통상황을 원활하게 표현하였다. 이에 대해 실험은 여러 요소를 적용하여 시뮬레이션 하였고 결과 그래프를 통해 제안된 방법들의 성능과 효과를 살펴본다.

5.1 실험 1: 전송메시지 수

실험 1은 노드의 이동 공간(Map)의 크기가 20x20인 환경에서 인지도 관리를 통한 전송메시지의 수에 대한 변화를 측정하였다. 본 실험은 ITS의 교통정보에 대한 송수신시 DIMM모델의 네트워크 밴드위스와 효율에 대한 성능을 비교하기 위해 NIMM(None Interest Management Model)과 비교하였다. NIMM은 모든 셀이 인지도 셀이 동일한 것을 의미한다. 그림 8은 DIMM과 NIMM에 대한 노드의 출발부터 도착까지 필요한 시뮬레이션 메시지의 수에 대한 그림이다.

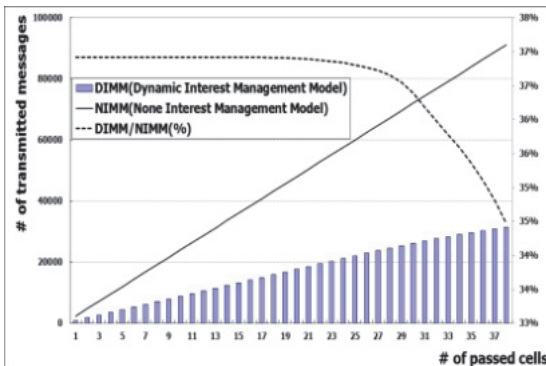


Fig. 8. Number of transmitted messages

DIMM는 인지도 셀을 관리함으로써 교통정보가 필요한 셀에 대한 메시지를 송수신함으로써 NIMM보다 75%

의 메시지를 절약한 것을 알 수 있고 셀이 많아질수록 메시지 전송률이 확실히 줄어드는 것을 확인할 수 있다.

5.2 실험 2: 시뮬레이션 시간

실험 2는 이동 공간의 크기가 20x20인 환경에서 이동 중인 노드가 인지도 관리를 적용하여 목적지까지 도착하기 위한 누적 시뮬레이션 시간을 측정하였다. 본 실험은 실험 1에서 발생한 메시지의 양이 줄어들 경우 교통상황에 대한 정확성이 떨어지게 된다. 하지만 메시지를 절약 하면서 교통상황의 정확성에 대한 오차도 최소화한다면 네트워크에 효율적이면서 안정된 시스템이라고 할 수 있다. 아래 그림 9는 20번의 실험을 통해 이동 중인 노드에 대한 DIMM과 NIMM의 목적지까지 평균 누적 도착시간을 표현한 것이다.

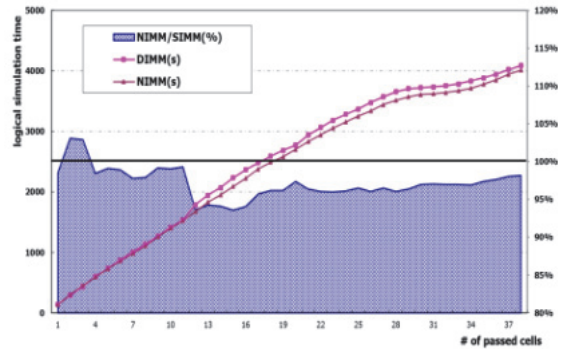


Fig. 9. Logical simulation time (DIMM vs. NIMM)

NIMM은 모든 셀에 대해서 인지도 셀이 높아 모든 셀의 정보갱신횟수가 빈번히 발생하여 정보의 정확도가 높지만, DIMM는 이동시간의 비율은 4초 이후부터는 NIMM의 약 90% 내외의 절약된 시간을 활용하여 목적지까지 통과하여 도착 시에는 약 95% 시간이 소요하여 시간이 절약된 것을 확인할 수 있다.

6. 결론

웹 기술은 시뮬레이션 분야에서 웹 서비스를 활용하여 모델의 분석, 개발, 서비스 제공에 활용되고 있다. 기술의 발달과 사람들의 생활수준 향상으로 차량은 지속적으로 증가하고 원활한 교통흐름을 관리하기 위한 도로, ITS 등의 인프라도 증가하고 있다. 본 연구에서는 복잡한 도로 환경에서 목적지가 있는 노드의 최적경로를 효율적으로 유도하기 위한 시뮬레이션을 진행하였다. 본 연구에서는

웹 서비스 시뮬레이션이 가능한 DEVS/WS 환경에서 차량 이동을 최적화하는 경로에 대한 정보를 제공하고 관리할 수 있는 시뮬레이션 모델을 제안하였다. 실험을 위해 분산 컴퓨팅이 가능한 웹기반의 Dynamic Interest Management Model(DIMM)을 디자인하였다. 셀 단위로 나뉘는 맵 내부의 도로에서 교통상황을 수집하기 위한 메시지의 수와 전송하는 전체 메시지의 양은 요구되는 정보 양을 절약하였고 이때 인지도 셀을 동적으로 조정하여 교통정보의 정확성의 오차를 최소화하는 범위에서 교통정보를 갱신하여 메시지 및 데이터양을 절약했다. DIMM은 인지도 관리의 결과와 전송된 메시지, 시뮬레이션 시간 절약을 통해 효율적인 노드의 이동을 유도함을 알 수 있었다. 본 연구에서는 웹 환경에서 이동 공간에 대한 인지도관리를 위한 시뮬레이션을 통하여 교통정보를 효율적으로 활용하여 최적 경로를 유도하여 시스템의 계산시간을 절약하는 결과를 얻을 수 있었다.

References

1. Veith, T. L., Koza, J. E. and Keolling, C. P.(1998), World Wide Web-Based Simulation, International Journal Engineering Education, Vol. 14, No. 5, pp. 316-321
2. 조규철,(2016), Operating Simulation of RPS using DEVS W/S in Web Service Environment, 한국컴퓨터정보학회 논문지, 21(12), pp. 107-114.
3. Zeigler, B. P. (2017), Emergence of Human Language: A DEVS-Based Systems Approach, SpringSim Multi-conference, Virginia Beach, VA, USA
4. Shah, A. A. and Lee, J. D.(2007), Intelligent Transportation Systems in Transitional and Developing Countries, IEEE A&E Systems Magazine, 27-33.
5. McNab, R. and Howell, F. W.(1996), Using Java for Discrete Event Simulation, Proceedings of the Twelfth UK Computer and Telecommunications Performance Engineering Workshop, University of Edinburgh, UK, pp. 219-228.
6. Healy, K. J. and Kilgore, R. A.(1997), Silk: A java-based process simulation language. Proceedings of the 1997 Winter Simulation Conference, pp. 475-482.
7. Li, W. D., Fuh, J. Y. H., and Wong, Y. S. (2004). An Internet-enabled integrated system for co-design and concurrent engineering. Computers in Industry, 55, pp. 87-103.
8. Fu, L., Sun, D. and Rilett, L. R., (2006), Heuristic shortest path algorithms for transportation applications: state of the art, Computers and Operations Research, v.33 n.11, p.3324-3343.
9. Lin, C. H., Yu, J. L., Liu, C. H., and Lee, C. J., (2008), Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems, Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering, pp.402-406.
10. Eddie, M., Mittal, S., Moon, H. H. and James, N. (2007), Automated Link 16 Testing using DEVS and XML, Journal of Defense Modeling and Simulation (JDMS).
11. Eshelman, L. J., Caruana, R. A. and Schaffer, J. D., Bases in the crossover Landcape, Proc. 4rd Int. Conf. on Genetic Algorithm. J. Schaffer(Ed.), Morgan Kaufmann Publishers, LA, 1989, pp. 10-19



조 규 철 (kccho@inhac.ac.kr)

2005 인하대학교 컴퓨터공학과 학사
 2007 인하대학교 컴퓨터정보공학과 석사
 2013 인하대학교 정보공학과 박사
 2011.1~2016.2 한국전자통신연구원
 2016.3~현재 인하공업전문대학 컴퓨터정보과 조교수

관심분야 : 클라우드 컴퓨팅, 모델링 & 시뮬레이션, 역공학, 신재생에너지