

자동차 가상플랜트 분산시물레이션을 위한 미들웨어 활용에 관한 연구

손창영*·노상도*·홍성원*

A Study on the Distributed Simulation of the Virtual Automotive Plant for a Virtual Manufacturing

Key Words: Virtual Manufacturing(가상생산), Virtual Automotive Plant(자동차 가상 플랜트), Middle ware, Distributed Simulation(분산시물레이션)

Abstract

본 연구는 자동차 회사를 대상으로 주요 생산공정중 하나인 차체 공장을 대상으로 한 가상플랜트를 설계하고, 이를 구축, 운영함으로써 생산준비의 각 부문에서 사전 검증을 통한 오류 방지, 의사결정의 최적화, 각종 데이터의 관리 및 실제 생산 설비와의 연계를 위한 자동차 Virtual Plant의 모니터링 및 운영 기술에 필요한 기능을 보유한 middle ware s/w 인 SysLink Integrator를 활용해 Virtual Plant를 분산 환경에서 구현해 봄으로, 향후 자동차의 가상플랜트구축에 따른 다양한 응용프로그램과의 통합을 위한, 유용한 하나의 middle ware로 그 활용성을 검토하였다.

1. 서론

1.1 개요

Virtual Plant구축과 관련하여, Delphi사에서 개발한 Virtual Plant Integration S/W인 SysLink Integrator의 기능 검토를 통해 향후 가상플랜트의 운영 및 모니터링을 위한 유용성을 살펴보았다.

1.2 SysLink 소개

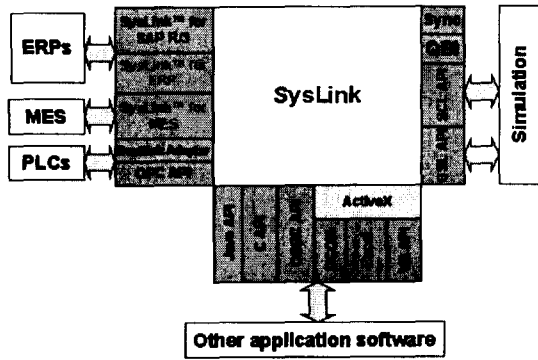
SysLink는 [그림 1.1]에서 나타낸 바와 같이 일종의 middleware s/w로서 기존 s/w들과 같이 사용되어 독립적으로 발달하고 있는 여러 영역들을

함께 묶어줌으로써 시너지 효과를 낼 수 있는 제품이다. SysLink는 크게 SysLink Integrator와 SysLink Planner의 2가지로 나뉘어 지며, 각각의 특징은 다음과 같다.

① SysLink Integrator

SysLink Integrator를 이용하면 시물레이션에 의한 실제 시스템 설계를 효율적으로 할 수 있다. 즉, SysLink Integrator를 이용하면, 하나의 컴퓨터 또는 다른 컴퓨터들에 설치되어있는 Delmia Products(Quest, IGRIP 등)간의 정보교환 뿐만 아니라 다른 Application들(Excel, ODBC, Java, OPC(PLC), MES 등)과도 Quest등이 데이터를 주고 받을 수 있다.

* 고등기술연구원



[그림 1.1] SysLink의 활용

따라서 한 컴퓨터에서 다루기 어려운 큰 모델을 작게 나눌 수도 있으며, 여럿이 각기 다른 부분을 모델링하여 같이 실행시키는 협업도 가능하다. 또한 시물레이션에 필요한 각종 입력 및 상태 조건들을 다양한 Application들로부터 그대로 입력받음으로써, 시물레이션 모델과 실제 플랜트 사이의 차이를 최소화할 수 있다. 결론적으로 현존의 시물레이션 기술의 한계를 극복할 수 있는 분산 협력 기반의 효율적인 모델의 구축이 가능하게 된다.

② SysLink Planner

SysLink Planner는 Web 기반의 Production Planning 및 Scheduling Solution으로서, SAP/R3 등의 ERP s/w와 시물레이션모델사이에 위치하여 정보전달 뿐만 아니라 자체적인 Scheduling Engine을 이용한 효과적인 생산계획을 수립할 수 있도록 함으로써, Agile Production을 가능케 하는 s/w이다. 즉 현재의 정적으로 다루어지는 scheduling 문제를 시물레이션 모델을 이용하여 내/외부의 변화(자재공급, 라인정지 등)를 고려하여 동적으로 풀어냄으로써 보다 더 정확한 결과를 얻을 수 있으며, 또한 Web 기반이기 때문에 장소 및 장비에 구애받지 않게 된다.

1.3 연구 내용

Evaluation은 크게 3 가지로 진행되었다. 첫째로, 기본 예제 정도의 단순한 Quest 및 IGRIP 모델에 대한 SysLink Integrator(이하 SysLink) 적용 결과 검토, 둘째로 실제 차체 공장을 대상으로 구축되어 있는 복잡한 Quest 및 IGRIP 모델에의 적

용, 그리고 마지막으로 OPC Adaptor를 이용한 PLC I/F 등을 수행중이고, 본 논문에서는 그 일부의 내용을 정리하였다.

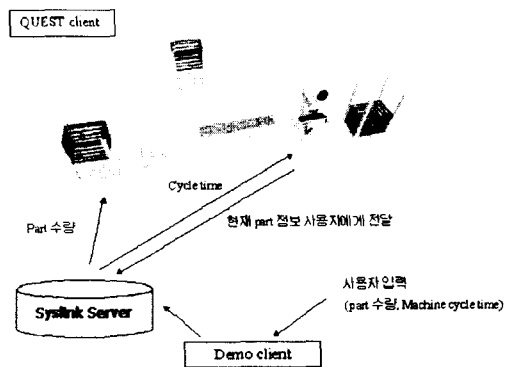
Evaluation 대상은 SysLink Integrator v2.2 이며, SysLink Server는 Client 와 TCP/IP를 통한 Socket connection을 기본으로 하여 통신하므로, 여러 대의 컴퓨터 또는 Client가 동시에 data을 주고받을 수 있다. 따라서 본 Evaluation에서는 모두 같은 컴퓨터상에서 실행하였지만, 각각의 컴퓨터(원격지)에서 실행하여 결과 값을 서로 주고받으며 시물레이션을 실행할 수도 있다.

2. SysLinker 기능 검토 - 예제 편

Quest 모델(Client)과 SysLink Server 와의 통신을 위해서는 library 의 path 설정, 초기화, SysLink configuration 파일의 작성, Quest Element구성 등에 대한 작업이 선결되어야 하며, 이에 대해서는 “3장 SysLink 기능 검토 - 응용편”에서 자세히 서술해 놓았다.

2.1 Demo Client에 의한 Quest 모델 제어

본 예제는 Quest 모델을 사용자가 직접 제어할 수 있음을 나타낸다. SysLink에서 제공하는 Demo Client를 통하여 Part의 정보를 받아들인 후 Quest 모델의 시물레이션이 실행된다. ([그림 2.1] 참조)

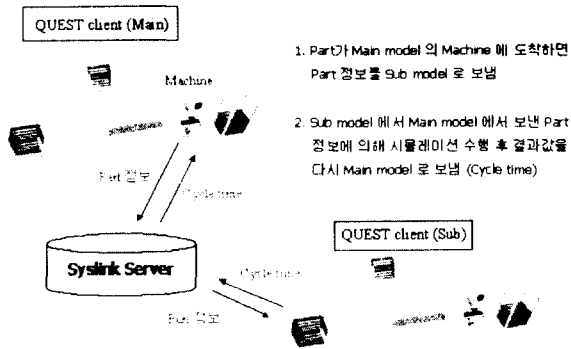


[그림 2.1] Demo Client에 의한 Quest 모델 제어

2.2 Quest 모델의 분리(Main, Sub)

두 개의 Quest 모델(Main, Sub)을 만든 후, SysLink를 통해 Main 과 Sub 간의 통신을 할 수 있도록 한다. 즉, [그림 2.2]와 같이 Main 모델의 Machine Cycle time 을 Sub 모델의 시뮬레이션 결과값으로 부터 읽어들이는다.

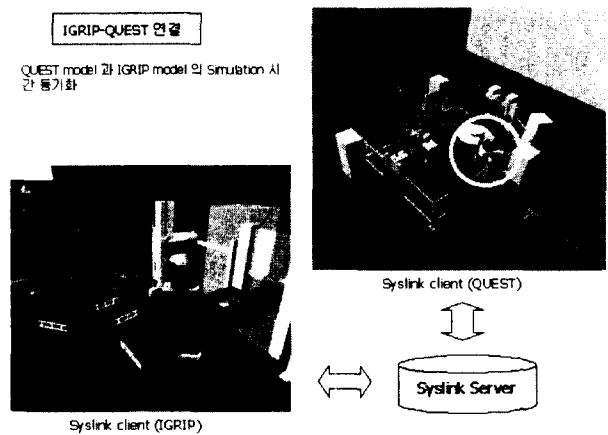
본 예제는 향후 가상 공장 구축시 각 공장간의 연결(프레스, 차체, 도장, 조립 공장) 하는 데에 유용하게 쓰일 수 있으리라 판단되며, 3.1절에서 차체의 1,2층을 대상으로 실현하였다.



[그림 2.2] Quest 모델의 분리(Main, Sub)

2.3 Quest-IGRIP 모델의 Integration

SysLink 를 통해 IGRIP-QUEST Integration 하는 예제로서, [그림 2.3]과 같이 QUEST 에서 모델링된 컴퓨터 조립라인 중 한 공정을 IGRIP 에서 상세 모델링하여 두 모델을 연결시켰다. 따라서 IGRIP-QUEST의 시뮬레이션 시간을 동기화하여 고정된 Cycle time 을 이용하지 않고 IGRIP Workcell 에서 시뮬레이션을 결과값을 매순간 QUEST 모델에 반영하여 보다 정확한 시뮬레이션 결과값을 얻을 수 있게 된다. 본 예제의 결과를 응용하여 3장 2절에서 차체의 Quest 모델과 OLP용 IGRIP Workcell 모델의 Integration을 실현하였다.



[그림 2.3] Quest-IGRIP 모델의 Integration

3. SysLinker 기능 검토 - 응용 편

3.1 차체 Quest 모델의 분리(1층 및 2층)

1. Syslink Server와의 통신을 위한 설정

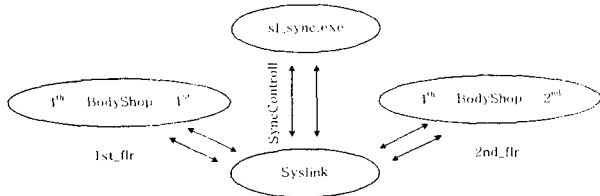
두 모델간의 연결을 위해서는 먼저 Input/Output 에 대한 논리적 연결관계를 정해야 한다. 본 모델에서는 차체 공장의 1/2층을 대상으로 하였으므로 [표 3.1]과 같이 U/BODY 대차를 기준으로 설정하였다.

[표 3.1] 두 Quest 모델의 논리적 연결관계

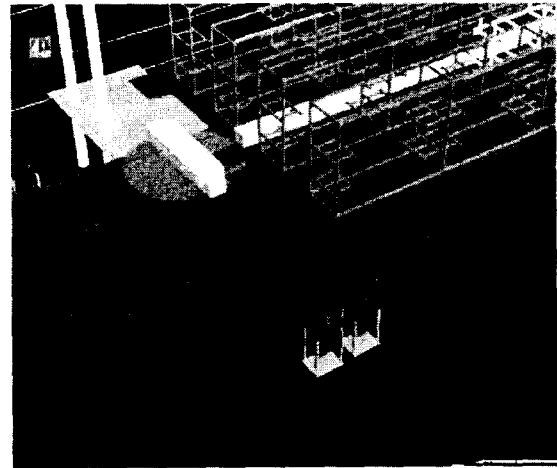
	Input	Output
모델 1 (차체공장 1층)	U/BODY rack storage로 들어가는 빈 대차	Main body를 실은 대차
	src_1st_flr_1	sink_1st_flr_1
모델 2 (차체공장 2층)	Main body를 실은 대차	U/BODY rack storage로 들어가는 빈 대차
	src_2nd_flr_1	sink_2nd_flr_1

QUEST 에서 Syslink server 와의 통신을 위한 설정은 크게 두가지가 있다. *.cfg 파일에서는 통신에 관한 것(Server&Client)과 QUEST element (RESOURCE)를 정의한다. *.inc 파일에서는 Syslink Server와 QUEST element와의 메시지 규칙을 정의하게 된다.

Syslink Server 를 통해 message를 주고받기 위해서는 QUEST 모델 내에 SL_Server_1 라는 element를 만든 후 QUEST 모델을 초기화 한다 (init_md1()). QUEST 모델의 Syslink Client 로 등록시에 각각 1st_flr, 2nd_flr 라는 이름을 사용하였다(RECEIVER, SENDER 항목). 동일한 컴퓨터에서 두 모델이 실행되므로 SOCKET 항목은 “localhost:5100” 으로 설정한다. 원격지 컴퓨터의 경우 IP 어드레스를 적어주면 된다. 나머지 설정은 기본값을 사용하면 된다. USER CONFIGS/SYSLINK MESSAGE CHANNELS 항목에서 Syslink Server 와 통신을 위한 element를 설정한다. 본 모델에서는 1층의 Drop lifter를 통해 Body를 실은 대차가 2층으로 올라가는 경우 Sink 에서 Part를 소멸시킴과 동시에 신호를 2층의 Source에 전달하여 Part를 생성시키는 방법을 사용하였다. 2층에서 1층으로 빈 대차가 Drop lifter를 통해 이동하는 경우에도 같은 방법을 사용하였다. 결과적으로 Part소멸/생성을 위한 1층의 Sink(sink_1st_flr_1) - 2층의 Source (src_2nd_flr_1), 1층의 Source (src_1st_flr_1) - 2층의 Sink(sink_2nd_flr_1)” 를 연결하여 두 모델을 통합하였다. 두 모델간의 동기화(Synchronization)를 위해서 각 모델내에 SL_Sync_1이란 이름의 element를 만든 후 SyncController(메시지를 전달하는 채널이름)를 통해 sl_sync.exe와 일정시간(기본값 : 1000ms) 간격으로 메시지를 주고받으며 동기화를 하게 된다. [그림 3.1]에 상관관계를 간략하게 나타내었다. [그림 3.2]는 QUEST 모델내에 Syslink Server와의 통신을 위한 SL_Server_1 과 sl_sync.exe (동기화) 과 통신을 위한 SL_Sync_1 element를 보여준다.



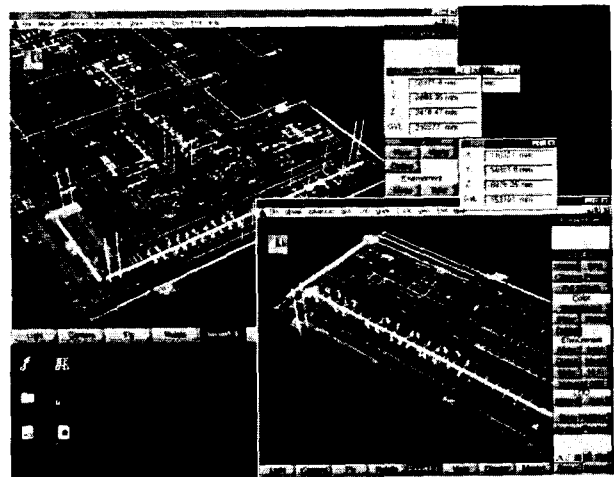
[그림 3.1] Syslink를 이용한 통합모델 구축시 상관관계



[그림 3.2] QUEST 모델내의 SL_Server, SL_Sync element

3.2 모델구축 결과

[그림 3.3]은 2개의 QUEST 모델이 동기화하면서 동시에 실행되고 있는 모습이다. 첫번째 모델이 먼저 실행되면 sl_sync.exe(background로 실행되어 화면에는 안 나타남) 는 두번째 모델이 실행될 때까지 대기한다. 두 모델의 time step에 관계없이 동기화되어 실행된다.



[그림 3.3] Syslink 통해 연결한 두 모델 (차체 1/2층)

3.3 결과

차체 공장을 대상으로하여 기구축된 모델의 1/2층을 분리하여 두개의 모델을 만든 후 SysLink를 이용하여 연결하였다. SysLink Server와는 별도로 Synchronization을 위한 약간의 SCL 코드의 변경이 필요했으며, sl_sync.exe(SysLink™ Synchronizer)를 구동하여 Syslink Server를 통한 Message와 별도로 Synchronizing을 위한 Message 처리를 하게 하였다. 따라서 QUEST 모델내에 기본적으로 SL_Server와 SL_Sync라는 두개의 element를 설정해야한다.

Syslink Server와의 메시지 교환을 위한 element의 설정파일 작성이 QUEST와는 무관하게 별도의 Text 파일에서 이루어지기 때문에 모델이 커지고, 설정해야 할 element의 수가 많을 경우에 초기설정 및 디버깅에 많은 시간이 소요되리라 예상된다. 차후에 SCL, BCL 등의 매크로를 이용해서 QUEST 프로그램 내에 버튼/메뉴 형태로 지원된다면 사용자 편의 향상 및 프로그래밍 시간을 단축시킬 수 있을 것으로 사료된다.

현재 Syslink Server를 통한 Client 간의 Message를 모니터링하기 위한 기능이 매우 빈약한 상태이다. Text파일의 형태로 제공되며 시물레이션이 완료된 후에 확인이 가능하다. 차후에 Message의 실시간 모니터링 기능이 구현되면 유용하리라 생각된다.

단일모델의 경우보다 QUEST 실행시의 시스템 부하에 Syslink와의 통신에 따른 부하가 더해지므로 전체적인 실행속도는 비슷하거나 약간 느렸으며, 네트워크 속도가 충분히 빠른 경우에 서로 다른 컴퓨터에서 실행하면 좀 더 향상된 performance를 보이리라 생각된다.

기존모델이 구축되어 있는 경우에 비교적 적은 양의 SCL 코드의 수정만으로 두 모델을 연결-동기화하여 시물레이션을 수행하는 것이 가능하였으며, 차후에 프레스/차체/도장/조립의 Virtual Plant 구축되었을 경우 Input/Output만 정확히 설정된다면 Syslink를 통한 Integration이 충분히 가능하리라 판단된다.