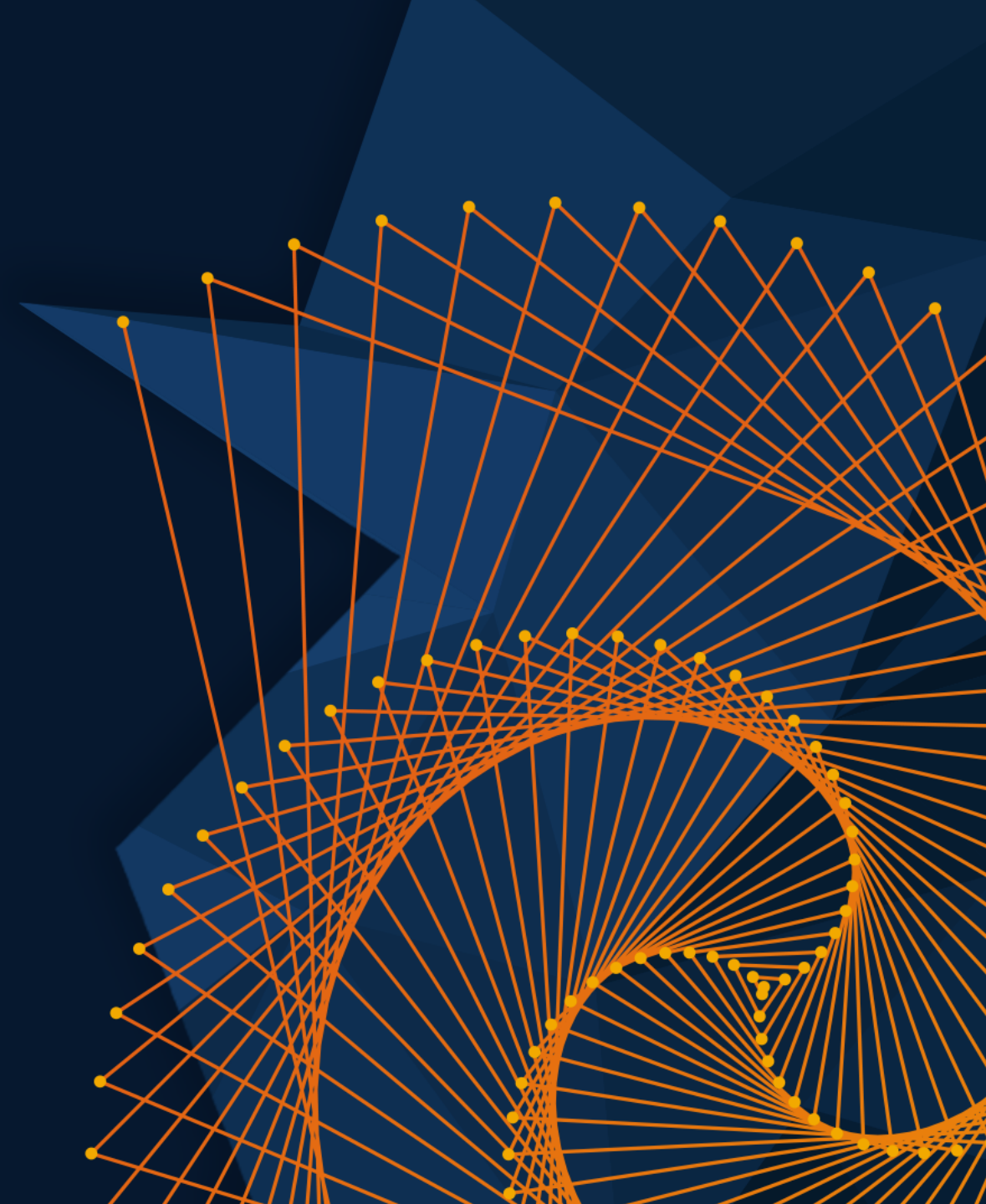


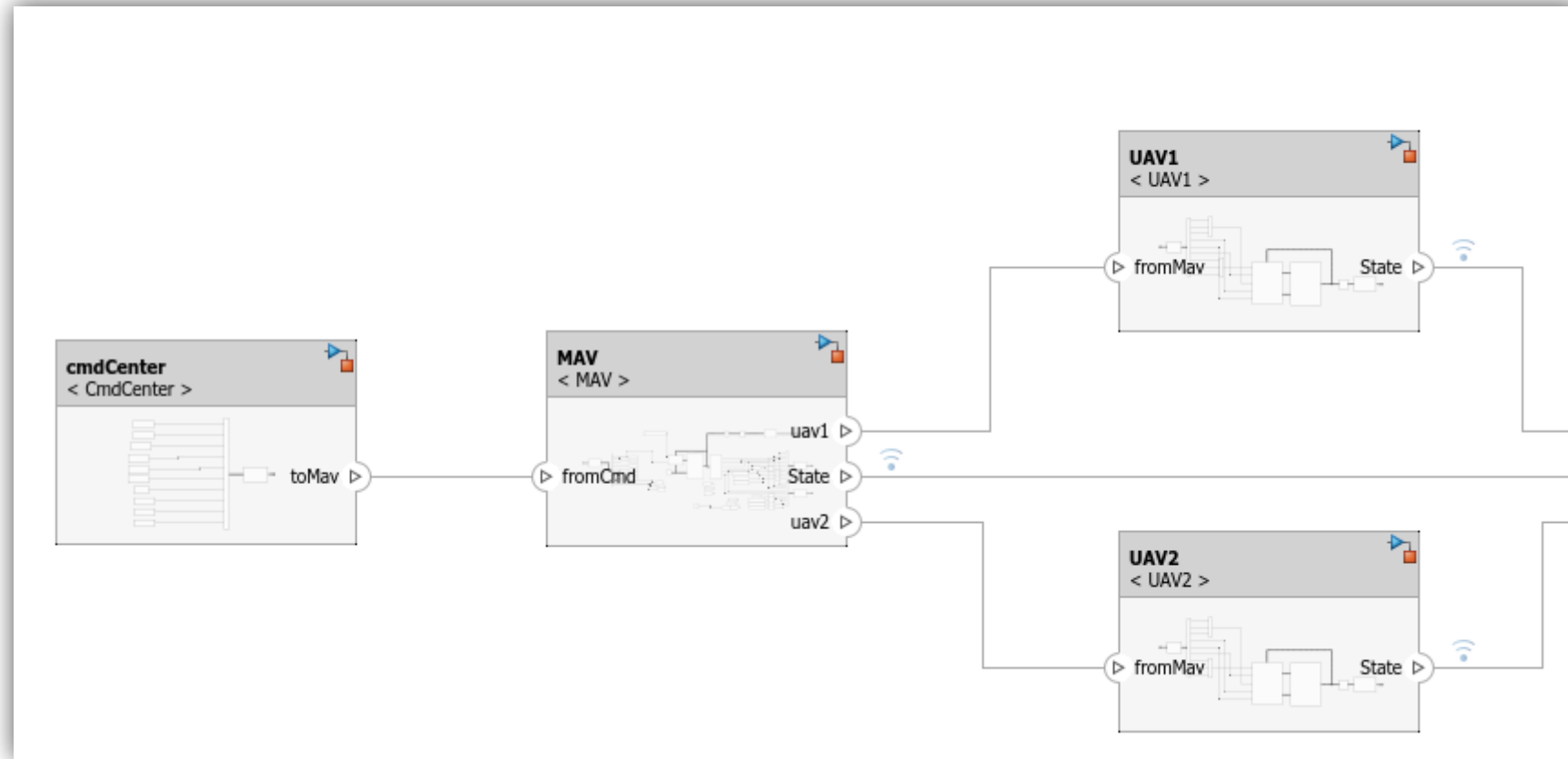
MATLAB EXPO

DDS Blockset을 활용한 무인기 분산 시뮬레이션

유성재, 매스웍스코리아

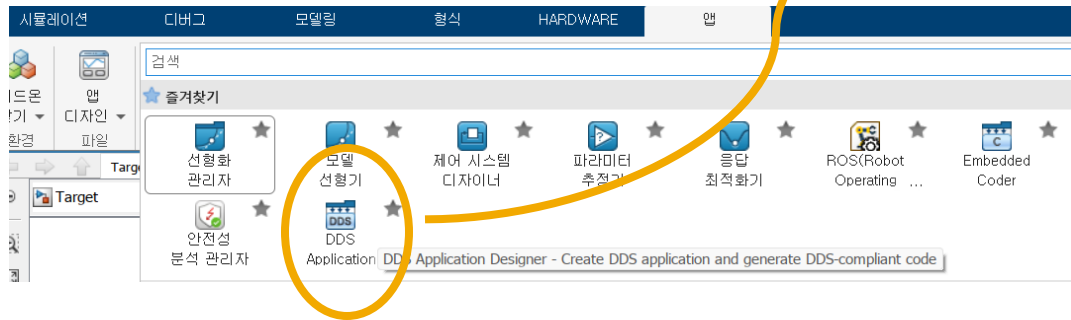


System Architecture of Multiple Aircraft Systems

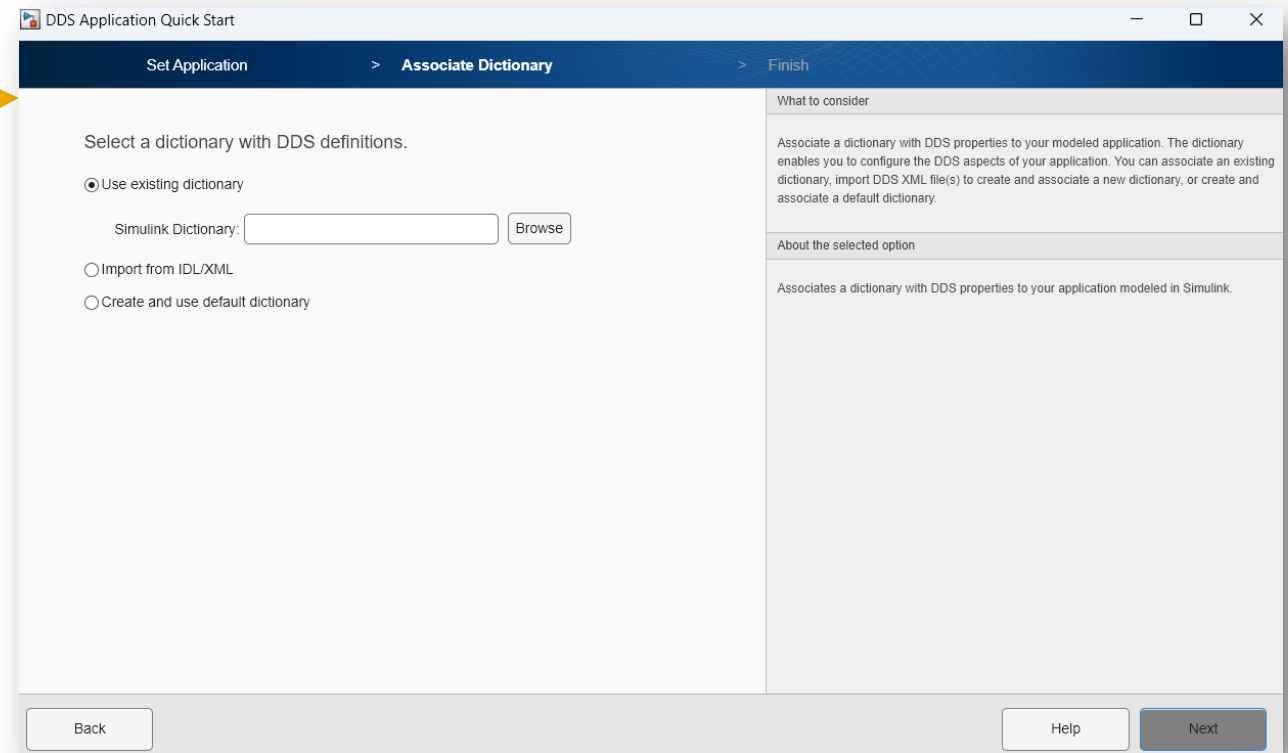


Component : Import DDS Interface definitions

- Import DDS definitions from XML or create new Definitions

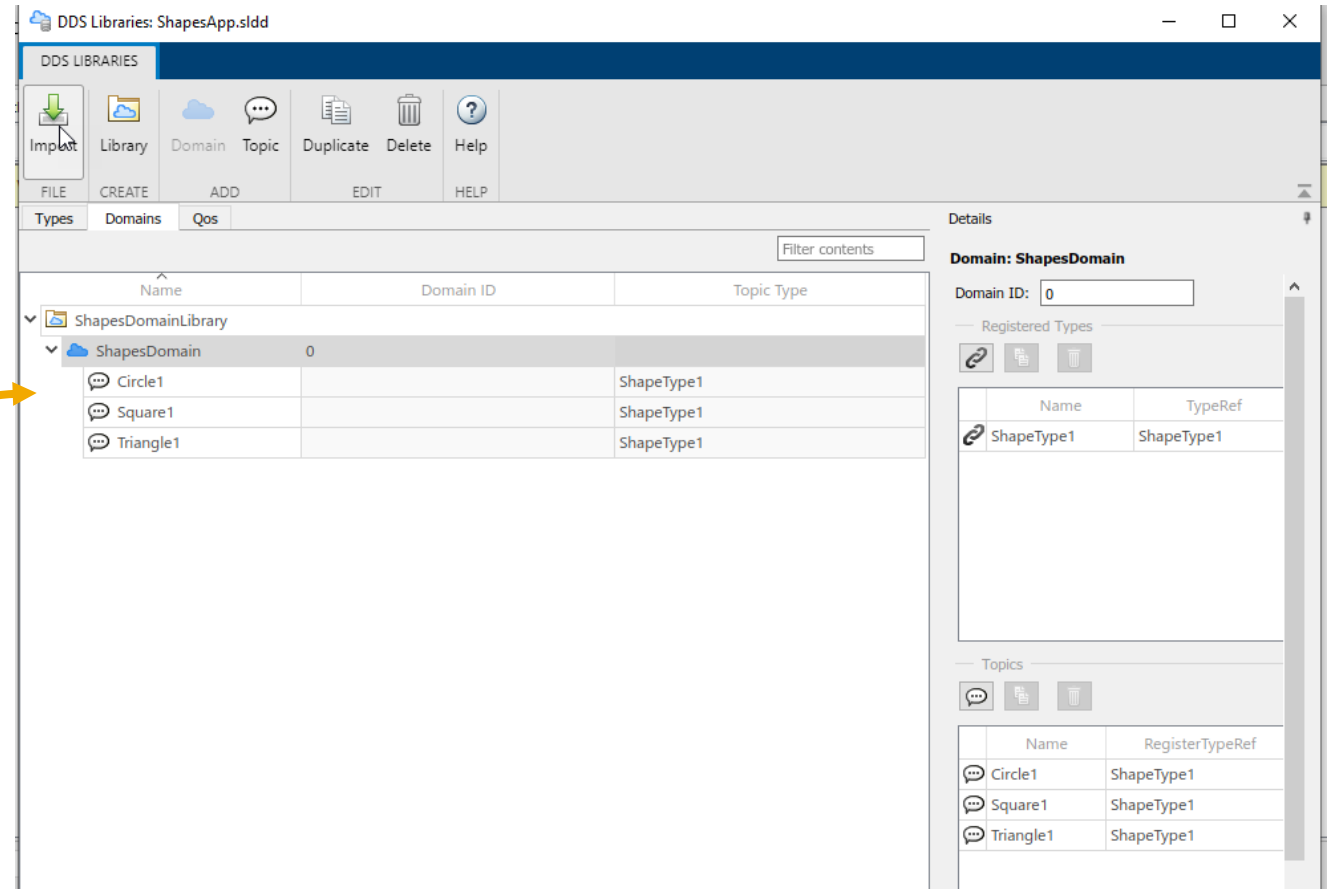
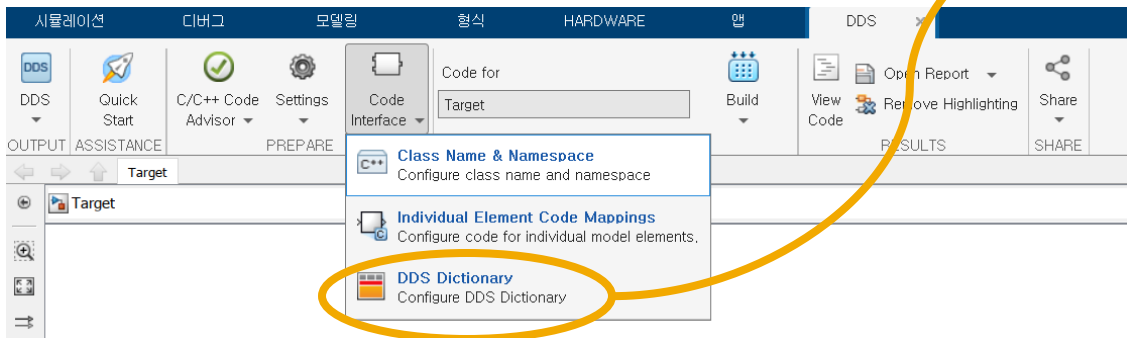


DDS Application Designer



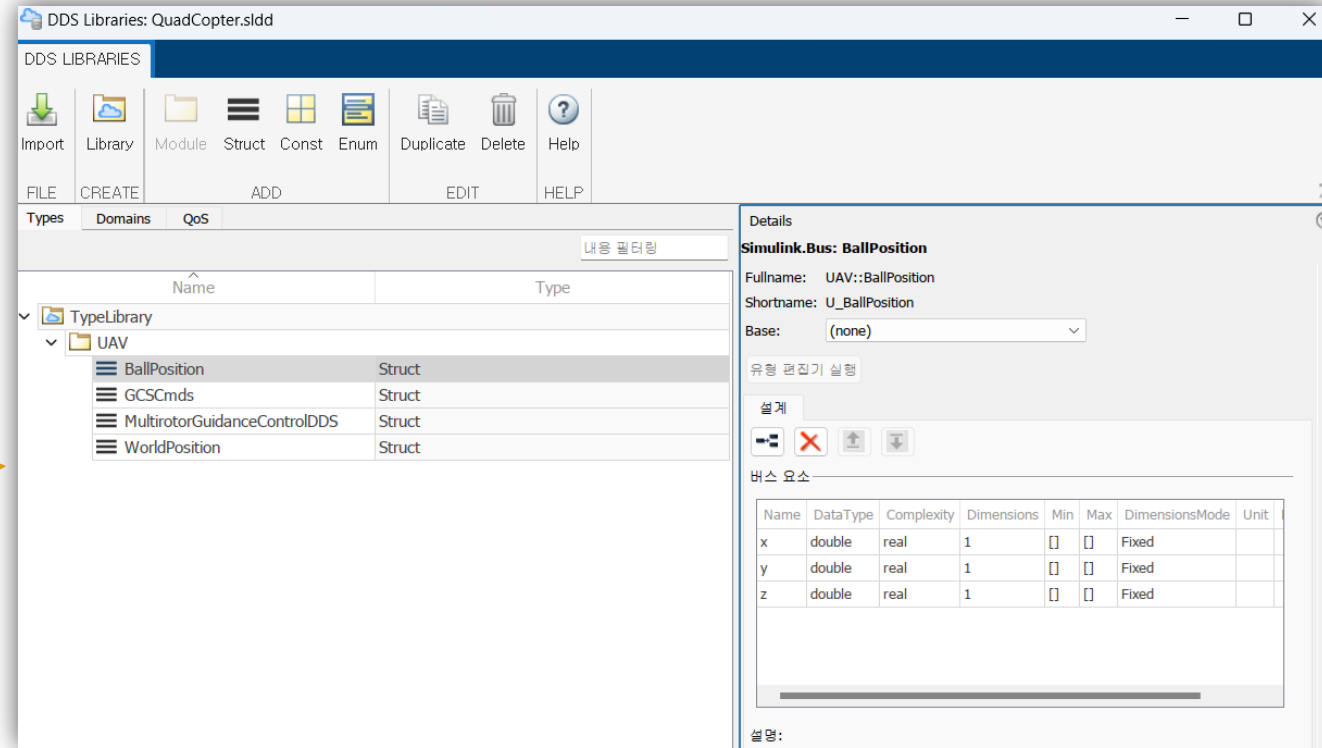
Component : Import DDS Interface definitions

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
 - Topic Types
 - Domains
 - QoS

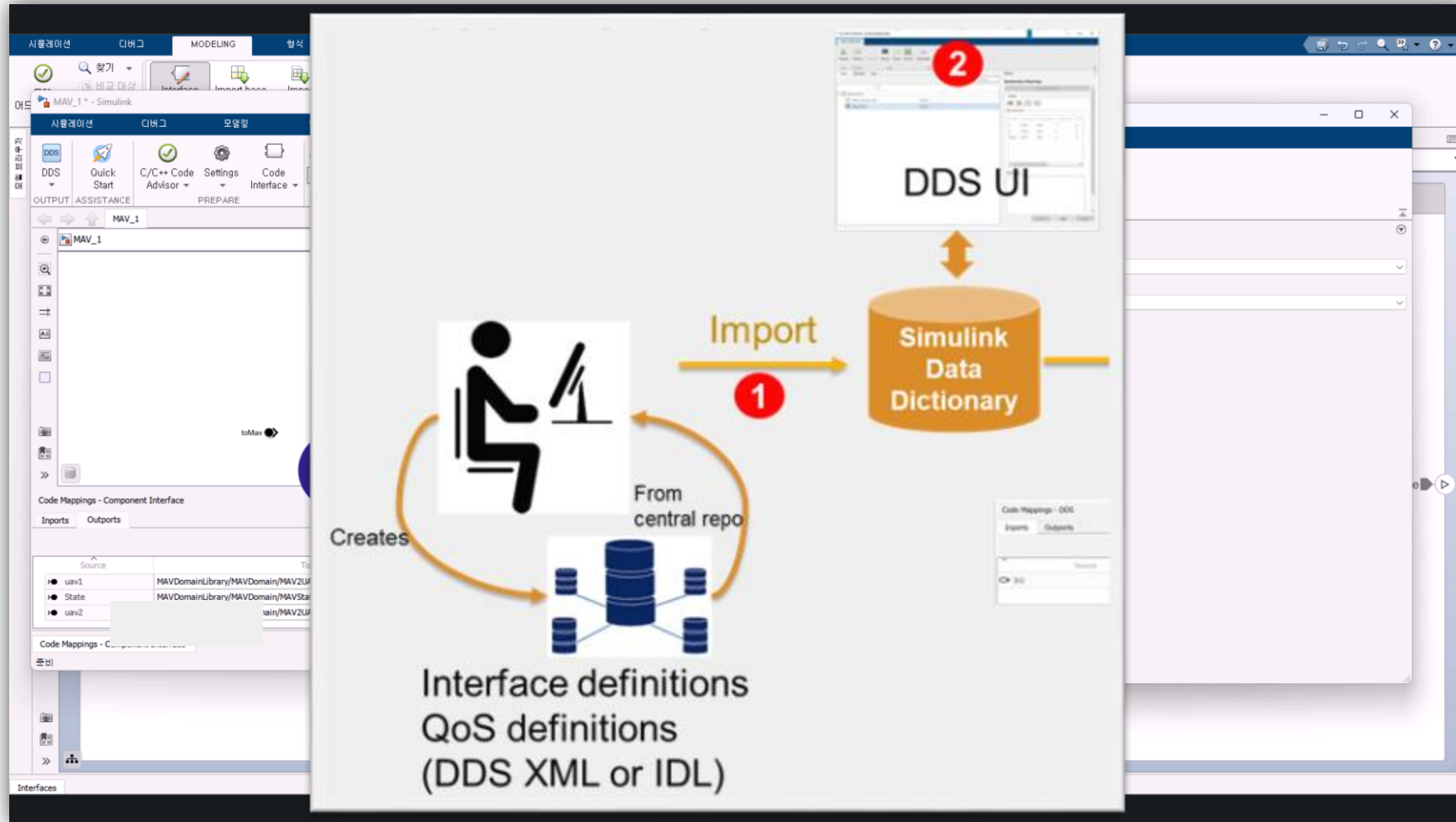


Component : Import DDS Interface definitions

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
 - Topic Types
 - Domains
 - QoS



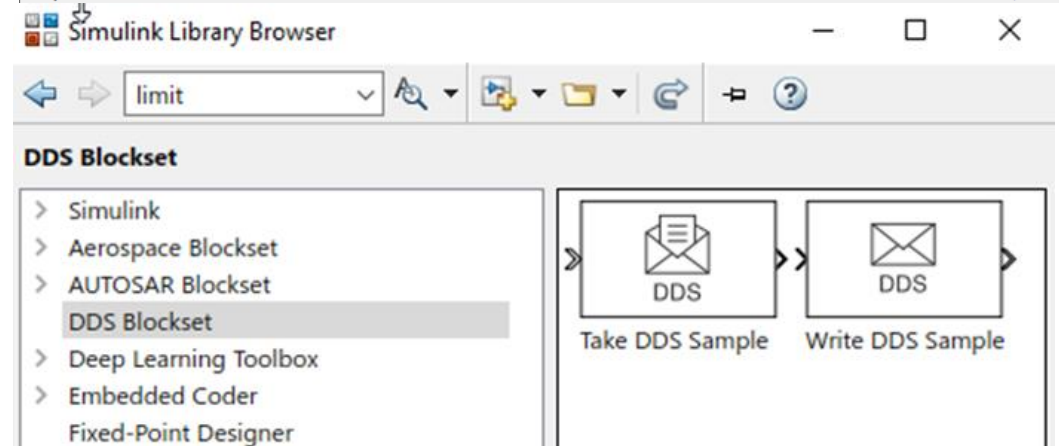
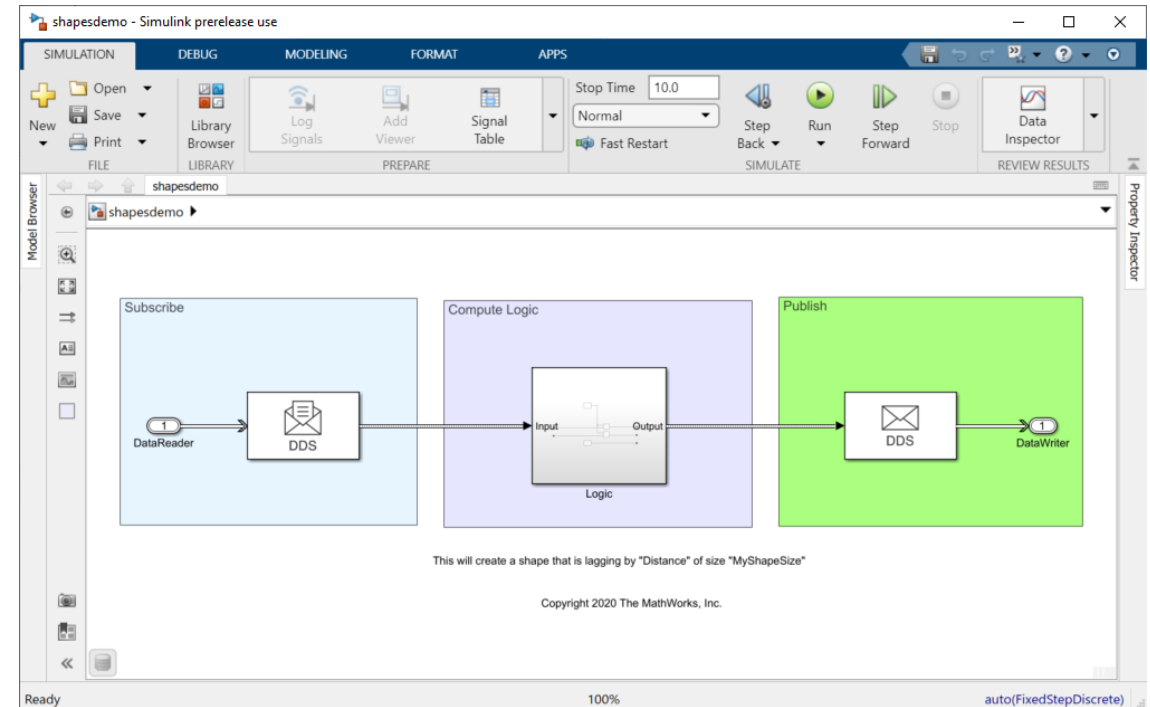
Component : Import DDS Interface definitions



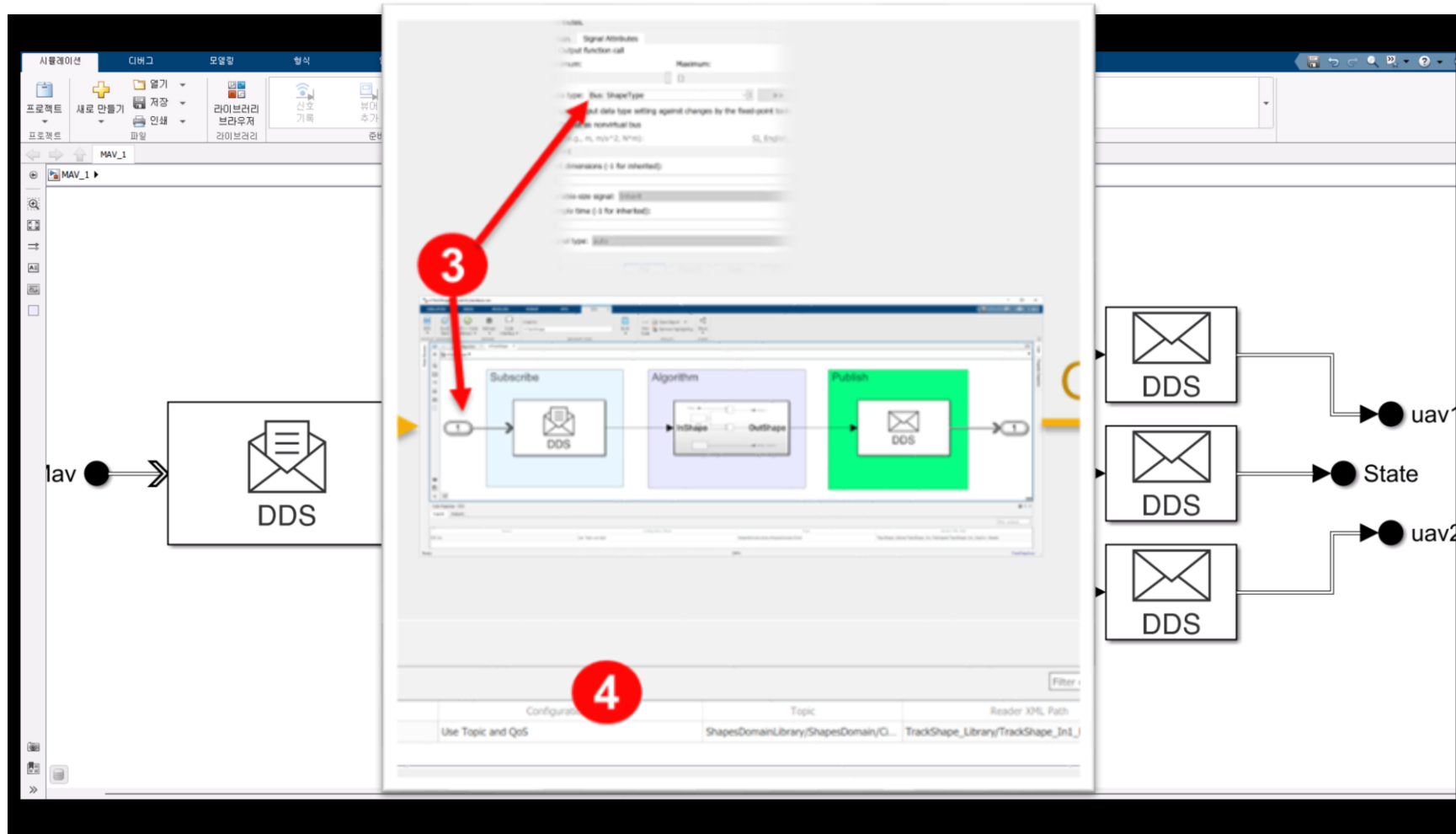
Component : Model DDS Application

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications

Use DDS Blocks to model a Publisher or Subscriber



Component : Model DDS Application



Component : Deployment of DDS Application

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications
- Simulate DDS models including QoS
- Generate DDS executables and deploy on a DDS network

```
bool writewithWriter(const PosType* data, std::string participantName, std::string writerName) {
    DDS_DataWriter* writer = getWriter(writerName, participantName);
    PosTypeDataWriter* fooWriter = PosTypeDataWriter_narrow(writer);
    if(!fooWriter) {
        return false;
    }
    const DDS_ReturnCode_t ret = PosTypeDataWriter_write((PosTypeDataWriter*)writer, data);
    return (ret == DDS_ReturnCode_t::DDS_RETCODE_OK);
};

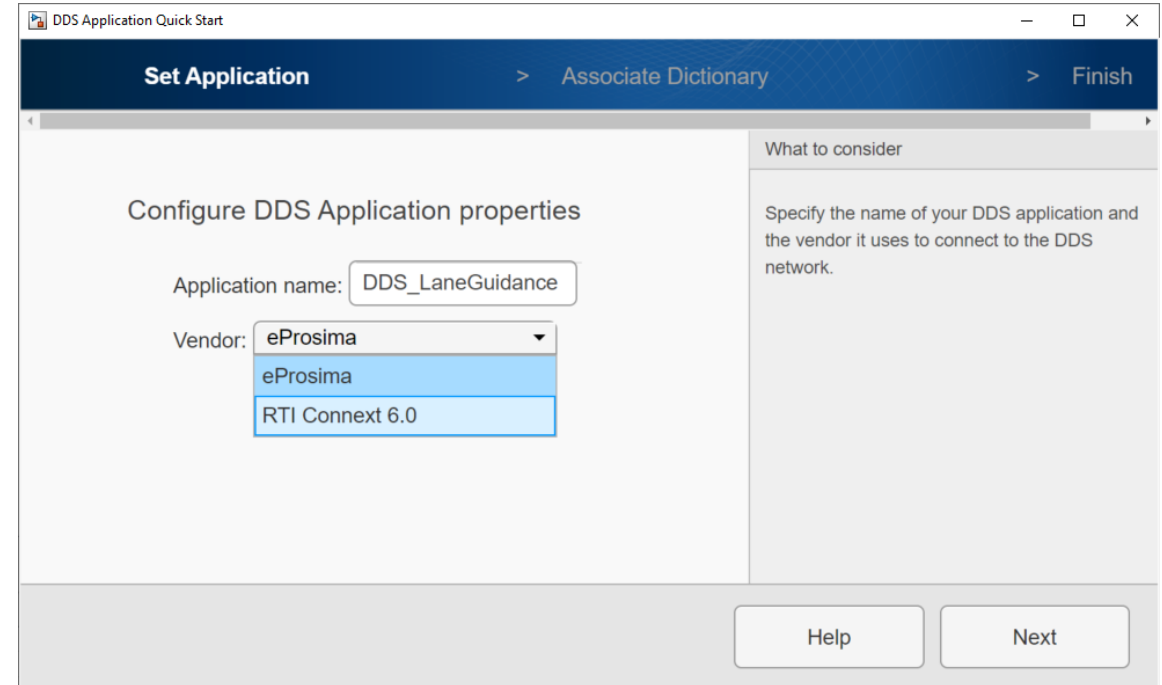
bool createParticipant(std::string participantName) {
    if (participants.find(participantName) == participants.end()) {
        DDS_DomainParticipant* participant =
            DDS_DomainParticipantFactory_create_participant_from_config(
                DDS_TheParticipantFactory, participantName.c_str());
        if(!participant) {
            return false;
        }
        participants[participantName] = participant;
    }
    return true;
};
```

With Embedded coder, generate

- C++ production code with DDS APIs
- XML or IDL files from Simulink models to deploy

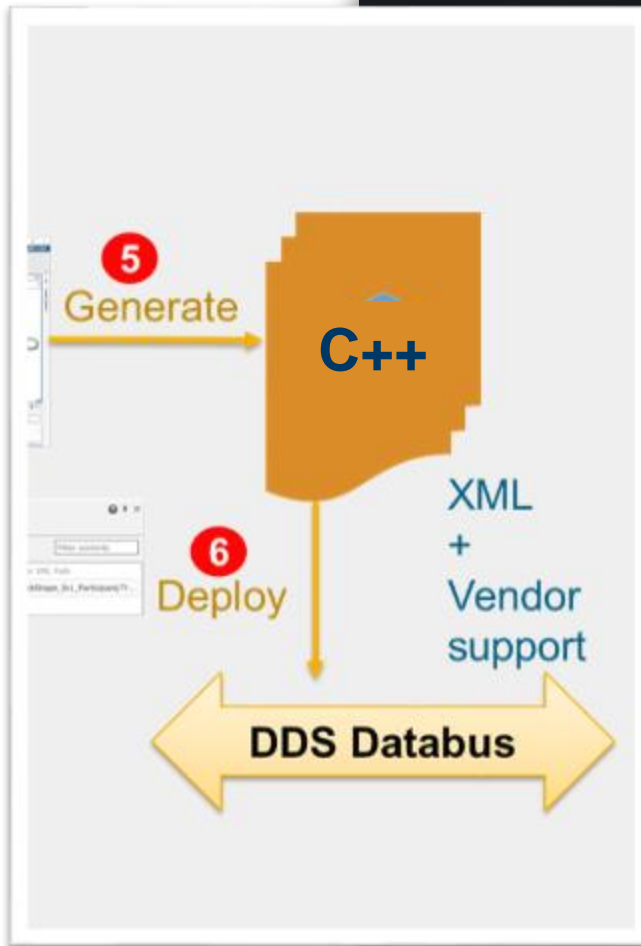
Component : Deployment of DDS Application

- Import DDS definitions from XML or create new Definitions
- Define/Modify DDS definitions in DDS Dictionary
- Model applications
- Simulate DDS models including QoS
- **Generate DDS executables and deploy on a DDS network**



Full integration with third-party DDS stacks including RTI Connnext and eProsima Fast DDS

Component : Deployment DDS Applications (Single Application)



Signals from Simulink

North	East	Height	publication_handle	instance_state
5595.875311837248	4482.914983359677	-16.041399999999943	01011fcd.fb5dc...	

Signal from DDS Message

Value

Time

In live mode

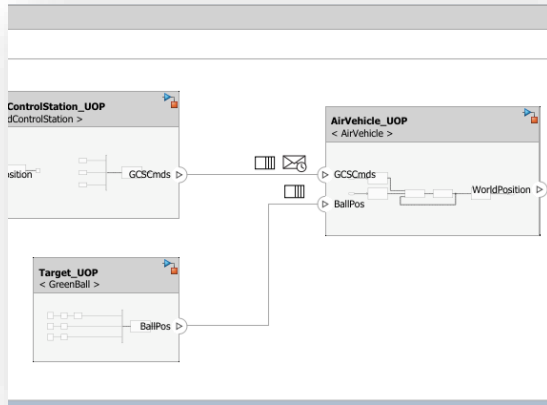
System : Deployment of DDS Applications

The screenshot displays the MATLAB Simulink environment for a system deployment. It is divided into several key sections:

- System Architecture:** A block diagram showing the interconnection of components including 'cmdCenter', 'MAV', 'UAV1', 'UAV2', and 'Visualization'.
- DDS Domain:** A configuration window for 'Domain 0' showing the network topology and application connections.
- Deployed DDS Applications:** A 'LinuxTarget1' window showing the deployment of applications like 'CmdCenter', 'UAV1', 'UAV2', 'Visualization', and 'MAV' to a target system.
- Signal from Simulink:** A plot showing time-series data for '<North>', '<East>', and '<Height>' signals, with a callout indicating the source is Simulink.
- Signal from DDS Message:** A plot showing a constant signal value of approximately 36.1 over time, with a callout indicating the source is a DDS message.
- Log Viewer:** A table at the bottom showing system logs with columns for Timestamp, AppID, CtxID, Type, and Message.

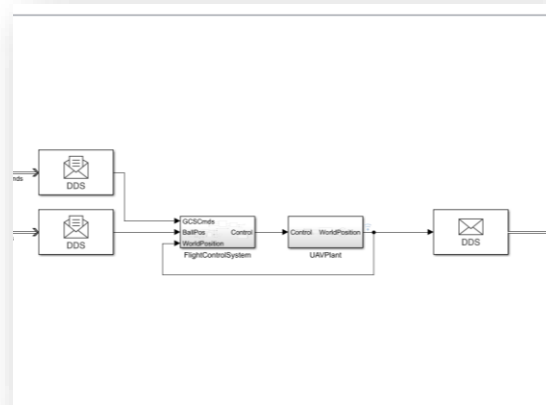
Timestamp	AppID	CtxID	Type	Message
2024/06/05 19:58:14	LNXD	LNXD	info	[ACTION: Terminating proc...
2024/06/05 20:00:50	LNXD	LNXD	info	[ACTION: Launching proce...
2024/06/05 20:00:50	LNXD	LNXD	info	[Process launch initiated: M...

Key Takeaways



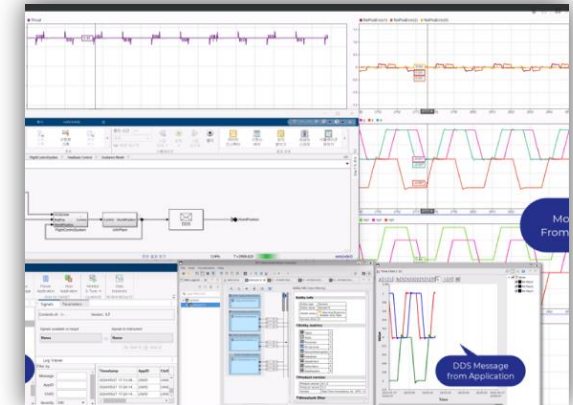
Architecture

System of Systems architectures are evolving, pushed by need for advanced, complex functions



DDS Model & Simulation

New, **service-oriented architectures** are required to **master complexity** and enable **frequent updates**



Deployment & Monitoring

You can **design, simulate and generate code** to deploy service-oriented applications in **Simulink**, reusing your **existing expertise and models**

MATLAB EXPO



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

