



ZyntRTI

DATA SHEET



★★★★★
Zyntelligent

398 หมู่ที่ 20 ตำบลขามเรียง อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150

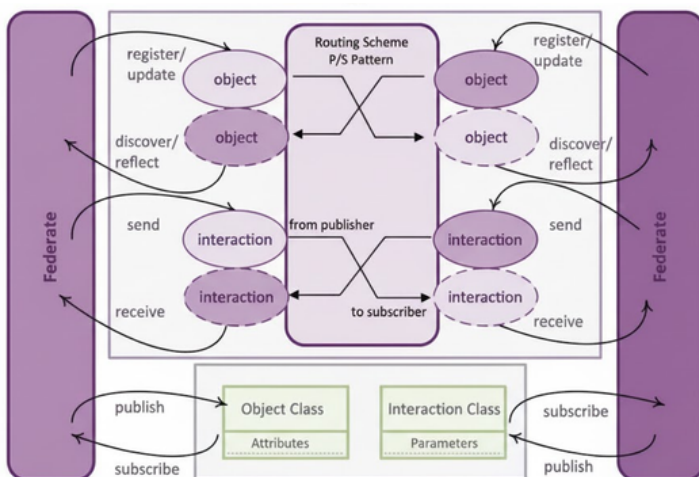
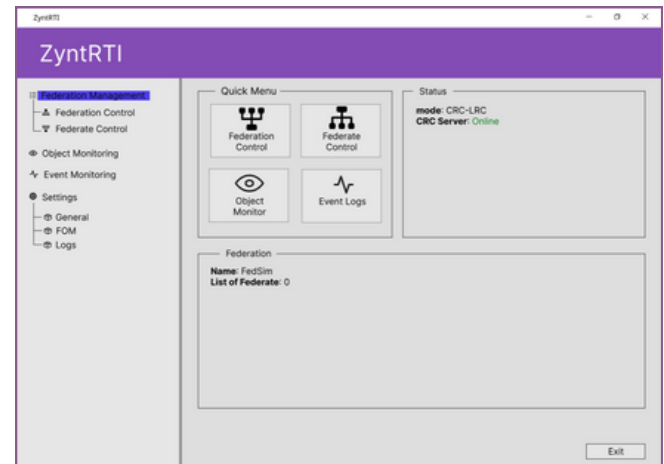
☎ 083 036 4720

✉ contact@zyntelligent.com



OVERVIEW

โปรแกรมจัดการระบบจำลองแบบกระจายศูนย์ ด้วยโซลูชันการเชื่อมโยงข้อมูลที่สมบูรณ์แบบของ ZyntRTI ซึ่งพัฒนาต่อยอดจาก Open-Source ยอดนิยมสำหรับงาน High Level Architecture (HLA) โดยได้พัฒนามุ่งเน้นไปที่การรองรับมาตรฐาน IEEE Std 1516-2010 (HLA Evolved) มีระบบจัดการที่เป็น Graphic User Interface (GUI) เพื่อให้ใช้งานได้ง่าย และให้ผู้ใช้ได้มั่นใจในการทำงานร่วมกัน และความแม่นยำสูงสุดในการจำลองร่วมกัน



ครบถ้วนทุกกระบวนการเชื่อมต่อแลกเปลี่ยนข้อมูลผ่าน HLA Service เช่น

- การรับ-ส่งแลกเปลี่ยนข้อมูล Object และ Interaction
- การสมัครรับข้อมูล (**subscribe**) และ เผยแพร่ข้อมูล (**publish**) ของ Object Class และ Interaction Class
- การลงทะเบียน (**register**) และ การปรับปรุง (**update**) ข้อมูลของ Object และ Interaction
- การตรวจพบ (**discover**) และ สะท้อน (**reflect**) ข้อมูลของ Object

ครบทุก Service ที่จำเป็นต่อระบบจำลอง (Simulation)

TECHNICAL

Software

- Name: ZyntRTI
- Type: HLA/RTI Management

System Requirements:

- CPU: 1 GHz or faster with 4 or more cores
- RAM: 16GB or higher
- Storage: 64GB or higher
- O/S: Windows 8/8.1/10/11 x86, 64Bit
- Network: Gigabit Ethernet (GigE)
- Program: Java 17 or higher

Programming Languages

- Java
- C++





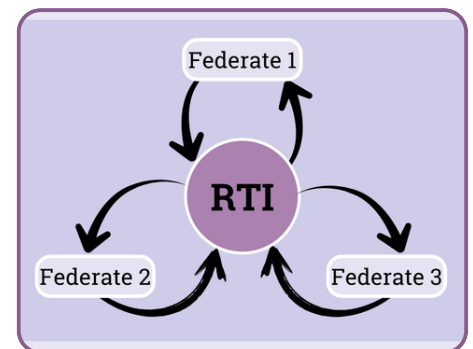
FEATURES

ระบบจัดการ Federation (Federation Management)

- ระบบจัดการ Federation และ Federate ภายในเครือข่ายระบบจำลอง
- เชื่อมต่อและทำงานร่วมกับ Run-Time Infrastructure (RTI) ภายใต้มาตรฐาน IEEE Std 1516-2010 (HLA Evolved)
- ออกแบบระบบและทำงานภายใต้ข้อบังคับมาตรฐานของ IEEE Std 1516-2010 (HLA Evolved)
- รองรับโหมดการเชื่อมต่อได้ทั้งแบบ Decentralized (LRC-LRC) และ Centralized (CRC-LRC)

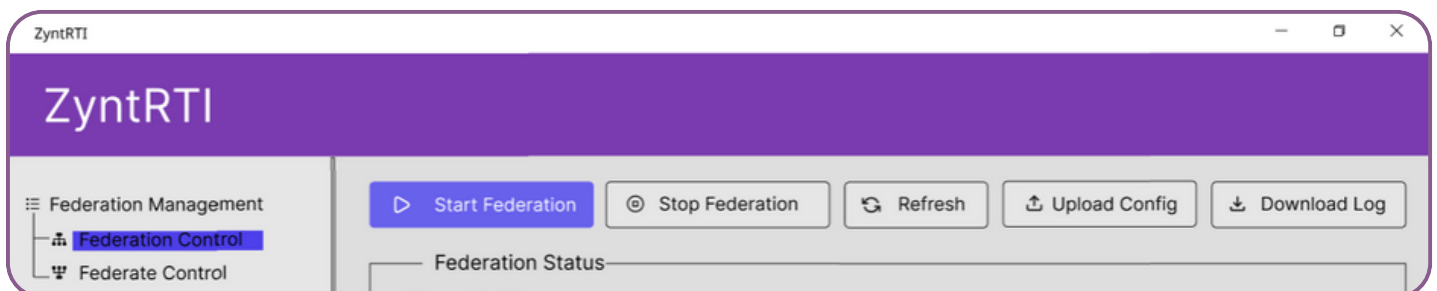
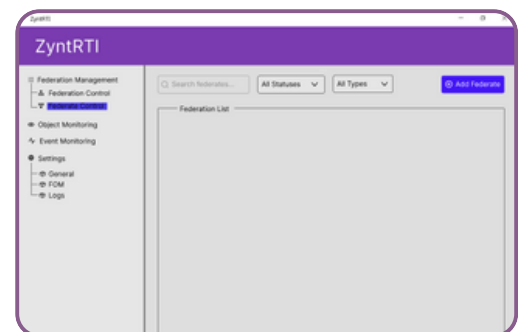
ระบบควบคุม Federation (Federation Control)

- ควบคุมการทำงาน และ สามารถสร้าง หรือ ทำลาย Federation ได้
- เรียกดูสถานะ และ รายละเอียดของ Federation
- ตั้งค่าและเรียกดู Federation Object Model (FOM) ที่ใช้ในการจำลอง
- แสดงรายชื่อ Federate ที่กำลังเชื่อมต่อกับ Federation ได้แบบเรียลไทม์



ระบบควบคุม Federate (Federate Control)

- ควบคุมการทำงานของ Federate
- เรียกดูสถานะ และ รายละเอียดของ Federate
- เรียกดู Registered Objects และ Interaction ของ Federate
- ส่งคำสั่ง Interaction ที่กำหนดไว้ใน Federation Object Model (FOM) เช่น การสั่ง Resign Federation, Shutdown, การ Announce หรือ Achieve Synchronization Point (Checkpoint) เป็นต้น





FEATURES

ระบบติดตามข้อมูล (Object Monitoring)

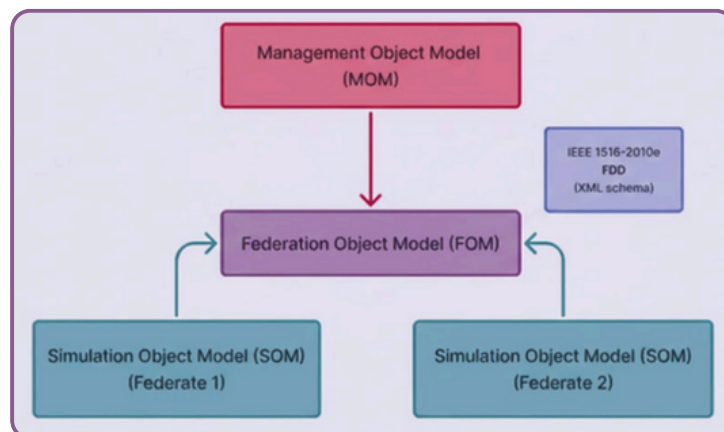
- ติดตามการเปลี่ยนแปลงของข้อมูล Object และ Attributes ภายใน Federation แบบเรียลไทม์
- ติดตามการส่ง Interaction ภายใน Federation แบบเรียลไทม์
- เรียกดู Registered Objects และ Interaction ภายใน Federation แบบเรียลไทม์

ระบบติดตามเหตุการณ์ (Event Monitoring)

- สามารถบันทึกข้อมูล Event ได้หลากหลาย เช่น วัน-เวลาที่เกิด Event, Federation และ Federate ที่เกิด Event, ข้อความ Event เป็นต้น
- ระบบบันทึก Event ที่เกิดขึ้นภายใน Federation และ ที่เกิดจาก Federate (Event Log) สามารถเรียกดูข้อมูล Event ทั้งแบบเรียลไทม์ และ ย้อนหลัง
- มีระบบกรองข้อมูล Event Log เพื่อการวิเคราะห์ที่รวดเร็ว
- บันทึก Event ข้อผิดพลาดที่เกิดขึ้นภายใน Federation และ ที่เกิดจาก Federate
- มีระบบบันทึกไฟล์ Event Log (text file) แบบ Automatic และ Manual

Federation Object Model (FOM)

- รองรับารออกแบบ Federation Object Model (FOM) และ Simulation Object Model (SOM) ที่เป็นไปตามมาตรฐาน IEEE Std 1516-2010 (HLA Evolved)





STANDARD

โปรแกรมออกแบบระบบและทำงานภายใต้ข้อบังคับมาตรฐาน 10 ข้อ ของ
IEEE Std 1516-2010 (HLA Evolved) – Framework and Rules

Summary of the HLA rules

กฎของ Federations

- 1) Federations shall have an HLA FOM, documented in accordance with the HLA OMT.
- 2) In a federation, all simulation-associated object instance representation shall be in the federates, not in the RTI.
- 3) During a federation execution, all exchange of FOM data among joined federates shall occur via the RTI.
- 4) During a federation execution, joined federates shall interact with the RTI in accordance with the HLA interface specification.
- 5) During a federation execution, an instance attribute shall be owned by at most one joined federate at any given time.

กฎของ Federates

- 6) Federates shall have an HLA SOM, documented in accordance with the HLA OMT.
- 7) Federates shall be able to update and/or reflect any instance attributes and send and/or receive interactions, as specified in their SOMs.
- 8) Federates shall be able to transfer and/or accept ownership of instance attributes dynamically during a federation execution, as specified in their SOMs.
- 9) Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of instance attributes, as specified in their SOMs.
- 10) Federates shall be able to manage local time in a way that will allow them to coordinate data exchange with other members of a federation

อ้างอิง: IEEE STD 1516™-2010, IEEE Standard For Modeling and Simulation (M&S)
High Level Architecture (HLA) - Framework and Rules, Section 4,5,6 (page 20-22)





STANDARD

Federation Design

- Federation structure A federation structure depicts the participating federate applications in a federation execution.
- Object model development (FOM, SOM using tables based on OMT tabular format) A object model development to develop the object models, which are a FOM, the shared data among federates, and SOMs, one per federate, which specify the capabilities of each federate in terms of the data it publishes and subscribes to.
- Behaviors of participating federates must be reflected, demonstrating their participation and role in the collaborative simulation, such as sending data, receiving data, interactions, or updating objects.

Federate Design

- Federate structure defines the architectural style of an application. Many software architectures may be used in order to develop a federate. the architecture of the federate structure must be consistent with the HLA Standard.
- Observable Behavior The behavioral specification of a federation encompasses all the observable interactions among participating federates from the viewpoint of the RTI in a federation execution.
- Federate internal behavior The applications do simulate an abstract model and share the state of the objects in the model with other members in a distributed simulation execution.

