Holonic Manufacturing Execution System

Fan-Tien Cheng

Institute of Manufacturing Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.

July 14, 2000
Outlines

😊 Introduction

😊 Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

😊 Development of WIP Holons

😊 Implementation and System Integration

😊 Summary and Conclusions
Outlines

✓ Introduction

😊 Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

😊 Development of WIP Holons

😊 Implementation and System Integration

😊 Summary and Conclusions
Introduction

- Background
- Motivation and Purposes
Introduction

Manufacturing Objectives

- Cost Reduction
- Cycle-Time Reduction
- Yield Improvement
- Empowerment
- Agility
- WIP Visibility
- Asset Utilization
- On-Time Delivery

眦 Failure recovery
眦 Security certification
Introduction

Background - MES Framework

Object Request Broker

Applications

Components

MES Framework

CORBA Infrastructure

Object Services

Common Facilities

Application 1

Scheduler

System Management

Equipment Management

Labor Management

Material Management

Design Pattern A

Other Component

Design Pattern B

Labor Management

Material Management

Equipment Management

System Management

Scheduler

Other Component

Design Pattern A

Object Services

Common Facilities
Introduction

Motivation and Purposes -

Characteristics of Holonic Manufacturing Systems

• Intelligence
• Autonomy
• Cooperation
• Reconfigurability
• Extensibility
Introduction

Motivation and Purposes - Deployment Diagram for Holonic Manufacturing Execution Systems
Outlines

😊 Introduction

☑️ Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

😊 Development of Scheduling Holons

😊 Development of WIP Holons

😊 Implementation and System Integration

😊 Summary and Conclusions
Basic Foundations

- Holon
- UML
- CORBA
- MES Framework
- Development Procedure for OO Systems
Basic Foundations

Super-ordination to parts (sub-holons)

Co-operation

Co-ordination with peers

Co-ordination with environment

Subordination to whole

internal structure & rules

rules for external behavior

Holon boundary

Material Holons

Equipment Holons

Scheduling Holon

Job Holon_1

Job Holon_2
Basic Foundations

UML (Unified Modeling Language)

- Use Case Diagram
- Class Diagram
- State Diagram
- Sequence Diagram
- Collaboration Diagram
CORBA (Common Object Request Broker Architecture)

OMG Reference Model Architecture

The Structure of a CORBA ORB
**Basic Foundations**

**MES Framework Architecture**

- **Object Request Broker**
- **Applications**
- **Components**
- **CORBA Infrastructure**

Diagram showing the architecture with various components and patterns.
Development Procedure for OO Systems

1. Collect and study domain knowledge
2. Collect requirements
3. Analyze requirements
4. Define Use Cases
5. Identify each scenario in each Use Case
6. Find classes
7. Find classes' relationships and their operations/attributes
8. Refine sequence diagrams
9. Key mechanism
10. Refine sequence diagrams iterative
11. Design complete
12. System implementation
13. System integration and testing

Basic Foundations

System Analysis

OOA

OOD

Implementation

Integration and Testing
Outlines

😊 Introduction

😊 Basic Foundations

✅ Development of Holonic Manufacturing Execution Systems

😊 Development of WIP Holons

😊 Implementation and System Integration

😊 Summary and Conclusions
Development of Holonic Manufacturing Execution Systems

- Development Procedure for HMES

- Collect Domain Requirements and Analyze Domain Knowledge

- Holarchy Design
  - Construct Abstract Object Model
  - Partition Application Domain into Components
  - Identify Generic Functions
  - Develop the Generic Holon
  - Define Holarchy Messages
  - Holarchy Framework of HMES
Development Procedure for HMES

1. Collect Domain Requirements
2. Analyze Domain Knowledge
3. Construct Abstract Object Model
4. Partition Application Domain into Components
5. Identify Generic Functions among Functional Holons
6. Develop Generic Holon
7. Define Holarchy Messages
8. Define Holarchy Framework of HMES
9. Design Functional Holons Based on Generic Holon
10. Construct Applications
11. Integrate System and Test It

System Analysis

Holarchy Design

Functional Holons Design and Implementation

Application Construction

System Integration and Testing
System Analysis

- Manufacturing Objectives shown in pp. 5.
- Open Interfaces
- Modularization
- Distribution
- Configurability
- Interoperability
- Maintainability
  ★ Security Certification
  ★ Failure Recovery
Holarchy Design -- Construct Abstract Object Model

Data Warehouse
Support all objects to access data

Factory Area

Labor Holon
Material Holon
Equipment Holon
WIP Holon

Shop-Floor Holon
Dispatches orders

Scheduling Holon
Dispatches jobs

Labor
Material
Equipment
WIP

Manages
Manages
Manages
Manages

Data Warehouse
Holarchy Design -- Partition Application Domain into Components

- Factory Area
  - Labor
    - Labor Management Component
    - Labor Holon
      - Manages
        - 0..*
  - Material
    - Material Management Component
    - Material Holon
      - Manages
        - 0..*
  - Equipment
    - Equipment Management Component
    - Equipment Holon
      - Manages
        - 0..*
  - WIP
    - WIP Management Component
    - WIP Holon
      - Manages
        - 0..*

- Data Warehouse
  - Support all objects to access data

- Shop-Floor Holon
  - Manages
  - Dispatches orders
  - Scheduling Holon
  - Manages
  - Dispatches jobs

- Scheduling Component

- Material Management Component
- Labor Management Component
- Equipment Management Component
- WIP Management Component

0..*
Holarchy Design -- Identify Generic Functions among Functional Holons

Intelligence Mechanism
- Exception Detection
- Exception Diagnosis

Search Mechanism
- Collaboration
- Reconfigurability

Security Mechanism
- Security Certification
- Data Encryption
- Data Decryption
Holarchy Design -- Develop the Generic Holon

- Requirements of GH
- Use Case Diagram for GH
- Sequence Diagram for Diagnosing Exceptions
- Class Diagram of Generic Holon
- Generic Holon Internal Architecture
Holarchy Design -- Develop the Generic Holon

Requirements of GH

- It can construct the message backbone for cooperation and communication purposes.
- It provides the mechanism for security.
- It provides the ability of error diagnosis.
- It provides the ability of searching the desired services.
- It provides the ability of establishing a database service for information storage.
Holarchy Design -- Develop the Generic Holon

Use Case Diagram for GH

1. Construct communication backbone
2. Search desired services
3. Use security mechanism
4. Use intelligence mechanism
5. Connect to local database

<<USE>>

Functional Holon

Database
Sequence Diagram for Diagnosing Exceptions (OOA)

1: Send exceptions
2: Diagnose exception
3: Get extension rules
4: Reasoning

Return error level
Holarchy Design -- Develop the Generic Holon

Class Diagram of Generic Holon (OOD)

Holon Kernel
- SetInitialService()
- SetDBConnection()
- SetRegistration()
- SetEncrypt()
- SetDecrypt()
- SetExceptionTest()
- SetSearchData()

ORB
- use

CORBA Interface
- InitializeORB()
- CreatObject()
- CloseObject()
- SetInterceptor()

Security Mechanism
- Validate()
- Encrypt()
- Decrypt()

Local Database
- Connection()
- RetrieveData()
- UpdateData()

Knowledge Base
- SearchEngine()
- Diagnose()
- AddRule()
- Match()
- RuleFilter()

Holon Configuration

Construct CORBA ORB

manage/use

retrieve
Holarchy Design -- Develop the Generic Holon

Generic Holon Internal Architecture

- Security Mechanism
- Local Database
- Knowledge Base

HolonKernel

CORBAInterface
Holarchy Design -- Define Holarchy Messages

- **Fully-Automated Holarchy Messages**
  - Sequence Diagram
  - Collaboration Diagram

- **Semi-Automated Holarchy Messages**
  - Sequence Diagram
  - Collaboration Diagram
Fully-Automated Holarchy Messages - Sequence Diagram

- External User
- Shop Floor Holon
- Scheduling Holon
- AS/RS
- AGV
- Robot
- EM Holon
- WIP Holon
- Data Warehouse

- place an order
- save order information
- dispatch order
- get WIP status
- get item master, equipment status, and order status
- dispatch job
- dispatch job
- dispatch job
- dispatch job
- dispatch job
- dispatch job
- dispatch job
- retrieve lot from AS/RS
- track out (retrieving from AS/RS)
- return track-out result
- update lot status
- setup parameter
- get BOM and recipe
- move lot to AGV
- transport lot to equipment
- track in (in equipment)
- return track-in result
- update lot status
- Loop back if the order was not done.
Fully-Automated Holarchy Messages - Sequence Diagram (Cont’d)

1. **External User**
   - Shop-Floor Holon
   - Scheduling Holon
   - AGV
   - Robot
   - EM Holon
   - WIP Holon
   - Data Warehouse

2. **Process Flow**:
   - **Shop-Floor Holon**:
     - External User requests a job.
   - **Scheduling Holon**:
     - Schedules the job.
   - **AGV**:
     - Transport lot to next station.
   - **Robot**:
     - Notify robot to pick up the lot.
     - Validate lot and equipment start process.
     - Update equipment status.
   - **EM Holon**:
     - Track out job done.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to equipment.
     - Track out (in equipment).
     - Update lot status.
     - Track out job done.
   - **Robot**:
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
   - **Data Warehouse**:
     - Order done.
     - Report order done.
   - **Shop-Floor Holon**:
     - Move lot to next equipment.
     - Notify robot to pick up the lot.
     - Move lot to equipment.
   - **EM Holon**:
     - Validate lot.
     - Equipment start process.
     - Update equipment status.
   - **WIP Holon**:
     - Track in (storing in AS/RS).
     - Send storing information.
     - Store lot in AS/RS.
     - Update lot status.
   - **AS/RS**:
     - Move lot to AS/RS.
     - Track in (storing in AS/RS).
     - Update lot status.
**Holarchy Design -- Define Holarchy Messages**

**Fully-Automated Holarchy Messages - Collaboration Diagram**

- **External User**
  - Place an order

- **Shop-Floor Holon**
  - Dispatch order
  - Get WIP status
  - Save order information

- **Scheduling Holon**
  - Dispatch order

- **AS/RS**
  - Move lot to AGV
  - Dispatch job

- **AGV**
  - Move lot to AS/RS
  - Dispatch job
  - Notify robot to pick up the lot
  - Robot

- **Equipment Management Holon**
  - Dispatch job
  - Finish job
  - Get FOM and recipe
  - Update equipment
  - Get item master, BOM

- **WIP Holon**
  - Get item master, equipment status, and order status
  - Update lot status

- **Data Warehouse**
  - Save order information
  - Update lot status

- **Robot**
  - Robot

- **Robot**
  - Robot

- **Robot**
  - Robot

- **Robot**
  - Robot

- **Robot**
  - Robot
Semi-Automated Holarchy Messages - Sequence Diagram

- External User
  - Shop-Floor Holon
  - Scheduling Holon
  - Material Mover Holon
  - Equipment Operator Holon
  - WIP Holon
  - Data Warehouse

place an order

dispatch order

save order information

get WIP status

get item master, equipment status, and order status

scheduling

dispatch job

if different product

[if different product] assign setup job

assign setup job

assign equipment setup job

setup equipment

transport lot

lot arrives

do track-in job

get BOM and recipe

scan barcode

validate lot

get WIP status

get item master, equipment status, and order status

scheduling

dispatch job

[if different product] assign setup job

assign equipment setup job

setup equipment

transport lot

lot arrives

do track-in job

get BOM and recipe

scan barcode

validate lot

validate lot
Semi-Automated Holarchy Messages - Sequence Diagram (Cont’d)

- External User
- Shop-Floor Holon
- Scheduling Holon
- Material Mover
- LM Holon
- Equipment Operator
- EM Holon
- WIP Holon
- Data Warehouse

- validate lot
- track in
- update lot status
- start equipment
- equipment start process
- update equipment status
- equipment job done
- do track-out job
- scan barcode
- track out
- update lot status
- track-out job done
- transport lot to next station
- order done
- report order done

Loop back if the order was not done.
Holarchy Design -- Define Holarchy Messages

Semi-Automated Holarchy Messages - Collaboration Diagram
Holarchy Design -- Holarchy Framework of HMES

Application 1

- Exchange Holon
- Scheduling Holon
- Shop-Floor Holon
- Equipment Holon
- WIP Holon
- Material Holon

Object Request Broker

Common Facilities

Object Services

Applications

Functional Holons

Holarchy

CORBA Infrastructure
Outlines

😊 Introduction

😊 Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

✓ Development of WIP Holons

😊 Implementation and System Integration

😊 Summary and Conclusions
Development of WIP Holons

- Requirements for WIP Holons
- Object-Oriented Analysis
- Object-Oriented Design
Development of WIP Holons

Requirements for WIP Holon

→ Real-Time Update WIP
→ Track by Process of Lots
→ Provide Information for Other Holons
→ Provide Information for Users
→ Ability of Accessing DBMS
→ Open Interfaces
→ Error Recovery
→ Security Certification
Development of WIP Holons

Holarchy Messages of WIP Holon

- Scheduling Holon
  - get WIP status

- WIP Holon
  - track in, track out
  - return track-in result
  - return track-out result

- Data Warehouse
  - get BOM, item master
  - update WIP status

- AS/RS
  - return track-in result
  - return track-out result

- AGV
  - return track-in result
  - return track-out result
Development of WIP Holons

Use Case Diagram for WIP Holon

- External User
  - Track out
- User
  - Login System
  - Track in
  - Provide lot information
- Common Database
  - Handle Exceptions
- Generic Evaluator
  - Login System
Development of WIP Holons

Creating New WIP Object (OOA)

- : External User
- : WIP Manager
- : WIP
- : Common Database

- create new WIPs
- generate new WIPs
- get BOM
- get item master
- get order status
Development of WIP Holons

Regular Track in (OOA)

: External User

WIP Manager

WIP

- Functional Holon asks to track in.
- get specific WIP
- track in

- get BOM on Eq
- check BOM and process sequence
- update WIP status

- [if wrong Eq] send wrong Eq message
- [if wrong BOM] send wrong BOM message
- [if OK] send starting processing message

- return track-in result
Development of WIP Holons

User Login System (OOA)

: User

User Interface

enter user information

send user information

WIP Manager

check user's authority

[if granted] enable track-in functions

[if granted] enable track-out functions

[if granted] enable query functions
Development of WIP Holons

OOA-Stage Class Diagram

Generic Holon

- HolonKernel
- HolonConfiguration

User Interface

- WIP Manager
  - WIP
  - Recovery Manager
  - 0...*
  - use
  - manage
Development of WIP Holons

Creating New WIP Object (OOD)

: WIPManager
: WIP
: BOM
: ItemMaster
: Common Database

: External User

CreateNewWIPs(lotID, quality, count)

WIP()

NewVariables()

Constructor of WIP Class

get BOM

get Item Master

get order status
Development of WIP Holons

Regular Track in (OOD)

External User

DoTrackin(Barcode, Station)

Returns TRUE if track-in is ok.

Searches WIP by parameters

WIPManager

WIP

Trackin(Barcode, Station)

See Sequence Diagram of "Track in"
Development of WIP Holons

“Track in” Operation (OOD)

These operations happen when the track-in is on last station.

get BOM on Eq
CheckBOM&PS ()
Checks BOM and process sequence.

ShowCheckResult ()

[if this track-in is the last station] DeleteRepository(lotID)

save WIP status

[if this track-in is the last station] DeleteMe()
Development of WIP Holons

User Login System (OOD)

- User
- User Interface
- WIP Manager
- Security Mechanism

- Login(userID, password)
- SetRegistration(userID, password)
- Validate(userID, password)
- [if granted] EnableTrackin()
- [if granted] EnableTrackout()
- [if granted] EnableQuery()
Development of WIP Holons

OOD-Stage
Class Diagram

Holon Configuration

- CORBInterface
- SecurityMechanism
- LocalDatabase
  - CORBAInterface
  - Validate()
  - Decrypt()
  - InitialORB()
  - CreateObject()
  - CloseObject()

HolonKernel
- SetInitialService()
- SetDBConnection()
- SetRegistration()
- SetEncrypt()
- SetDecrypt()
- SetErrorTest()
- SetSearchData()

KnowledgeBase
- SearchEngine()
- Diagnose()
- AddRule()
- Match()
- RuleFilter()

LocalDatabase
- Connection()
- RetrieveData()
- UpdateData()

manage/use

UserInterface
- CreateNewWIP()
- DoTrackin()
- DoTrackout()
- Query()
- SendException()
- KillRepository()
- ValidateID()
- Recover()
- SaveLog()

WIPManager
- ReConnectLDB()
- TryConnection()
- AlarmAGV()
- AlarmASRS()
- AlarmRobot()

WIP
- LotID : String = initval
- Barcode : String = initval
- StorageX : Integer = initval
- StorageY : Integer = initval
- Quality : Integer = initval
- Type : String = initval
- Station : String = initval
- BOM : Object = initval
- ItemMaster : Object = initval
- OrderID : type = initval

BOM
- NewVariables()

ItemMaster
- NewVariables()

Generic Holon

- Trackin()
- Trackout()
- DestroyMe()
- GetStructure()
- CheckBOM&PS()

UserInterface

RecoveryManager

AutoLab

50
Development of WIP Holons

State Diagram of WIP Object Life Cycle

1. Start
   - Create WIP
     - do: WIP

2. Create BOM
   - do: BOM:NewVariables

3. Create Item Master
   - do: ItemMaster:NewVariables

4. Getting order status
   - All WIP initial information is collected
   - do: GetStructure

5. Wait for request
   - track-in request
   - track-out request

6. Do track-out
   - do: Trackout

7. Do track-in
   - do: Trackin

8. Store WIP status to Database
   - [not last process]

9. Store WIP status
   - to database
   - do: DeleteMe

10. Delete the WIP repository
    - do: WIP:DeleteRepository

11. end
State Diagram of WIP Holon

start

User Login the System
do: Login

[if granted]

Initiate CORBA Service
do: SetInitialService

Receives creating command

[If last process finished] End

[An Exception occurred]

Diagnose
do: SetExceptionTest

Do Recovery Operation
do: Recovery

[Recovery failed]
Send an Alarm

End

[Recovery successful]
Back to the last operation where the exception is occurred

WIP Object Life Cycle
Outlines

😊 Introduction

😊 Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

😊 Development of Scheduling Holons

😊 Development of WIP Holons

☑️ Implementation and System Integration

😊 Summary and Conclusions
Implementation and System Integration

✦ System Deployment Architecture

✦ Functional Holons Implementation Procedure

✦ Application Construction with System Integration
Implementation and System Integration

System Deployment Architecture

[Diagram with various components such as Data Warehouse, Shop-Floor Holon, WIP Holon, Scheduling Holon, Material Handling, Equipment Holon, Robot, AGV, AS/RS, and their connections]
Functional Holons Implementation Procedure

1. Generic Holon Development
   1. Generic Holon Development

2. HMES Framework
   2. Define Functional Holon Framework IDL
      - idl2java

   - 3B-1: Client Stub
     - 3B-2: Implement Functional Holon Client
       - javac
     - 3B-3: Locate Servers of other Functional Holons

   - 3A-1: Server Skeleton
     - 3A-2: Implement Functional Holon Server
       - javac
     - 3A-3: Register Functional Holon Server

3. Functional Holon Development
   4. Functional Holon Ready

Implementation and System Integration
Implementation and System Integration

Application Example for Getting WIP Status

Scheduling Holon
- SH:Scheduling Manager
- SH:Information Manager

GetWIPStatus(lotID)

WIP Holon
- WH:WIPManager
- WH:WIP

_query(Barcode, lotID, Station)

- [if WIP exists] GetStructure()
- [if WIP doesn't exist] Get WIP's record

Returns a structure of WIP Status including "Barcode", "lotID", "Station", "Quantity", and "Type"
Outlines

😊 Introduction

😊 Basic Foundations

😊 Development of Holonic Manufacturing Execution Systems

😊 Development of Scheduling Holons

😊 Development of WIP Holons

😊 Implementation

✔️ Summary and Conclusions
Summary and Conclusions

▪ Comparisons between MES and HMES

▪ Summary and Conclusion
## Summary and Conclusions

### Comparisons between MES and HMES

<table>
<thead>
<tr>
<th></th>
<th>Legacy MES</th>
<th>MES Framework</th>
<th>HMES Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architecture</strong></td>
<td>Centralization</td>
<td>Distributed OO</td>
<td>Holarchy</td>
</tr>
<tr>
<td><strong>Open Interfaces</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Modularization</strong></td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td><strong>Interoperability</strong></td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td><strong>Configurability</strong></td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Maintainability</strong></td>
<td>Difficult</td>
<td>Easy</td>
<td>Easy</td>
</tr>
<tr>
<td><strong>Security Certification</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Failure Recovery</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Legacy MES: Promis, WorkStream, etc., Framework MES: SiView, FACTORYWorks, etc.*
Summary and Conclusions

- Present a new concept for developing next generation Manufacturing Execution Systems.

- Provide a systematic procedure to develop holonic manufacturing systems.

- Demonstrate the feasibility and capability of HMES by designing and implementing a WIP Holon.
END

Thanks……..