GAIA ANALYSIS AND DESIGN OF A MULTI-AGENT-BASED SHOP FLOOR CONTROL SYSTEM
Agent based shop floor control systems

The Gaia methodology

Analysis phase

Design phase

Conclusions
Agent-based shop floor control systems

Multiagent Systems

Agent: autonomous and flexible software entity that is situated in an environment

To achieve its objectives, agents interact with each other by means of communication.

Multiagent System
Agent-based Shop Floor Control Systems integrate the programming, execution and control activities, in a software system made up of several autonomous agents, which can interact using their social abilities.

- **O₁**: Manager Order Agent
- **Mₘ**: Manager Machine Agent
- **mₘ**: Physical Machine

Manufacturing Shop Floor
Content

- Agent based shop floor control systems
- The Gaia methodology
  - Analysis phase
  - Design phase
- Conclusions
Software methodologies: frameworks to structure, plan, and control the process of developing information systems.

Agent-based software methodologies: software methodologies to develop multiagent systems.

Gaia methodology: MAS is a set of autonomous and interactive agents that live in an organized society in which each agent plays one or several roles and interact with other agents by means of different protocols.

Gaia supports the two levels of design in MAS: the individual agent structure and the agent society.

Procedure: Two Phases
Agent based shop floor control systems
The Gaia methodology
Analysis phase
Design phase
Conclusions
**Shop floor control (SFC)** is a system of computers and/or control tools used to schedule, dispatch and track the progress of orders through manufacturing system.

- **Inputs**
  - Machine features
  - Item *production process* (operations and precedence relationships)
  - Manufacturing *orders* (item and due date)

- **Functions**
  - Allocate and schedule tasks on machines
  - Dispatch the operation execution to the machines
  - Monitor the status of the plant
  - React to the disturbances and reschedule when necessary

- **Outputs**
  - Performance measurements

- **Special features**
  - The *allocation* is chosen by the system at *runtime*
  - The *schedule* is done by means of a *auction process*
## User roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MACHINE MANAGER</strong></td>
<td>It <strong>sets machine parameters</strong>: capacity and technical possibilities and specifications</td>
</tr>
<tr>
<td><strong>PROCESS DESIGNER</strong></td>
<td>It <strong>sets products can be manufactured</strong> in the system. For each product, it designs their manufacturing processes (operations and precedence relationships)</td>
</tr>
<tr>
<td><strong>PLANNER</strong></td>
<td>It <strong>defines and dispatches orders</strong>. Moreover, it analyzes the system efficiency</td>
</tr>
<tr>
<td><strong>SHOP FLOOR MANAGER</strong></td>
<td>It <strong>sets the system parameters and monitors</strong> the system</td>
</tr>
</tbody>
</table>
## System roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MACHINEOPS</strong></td>
<td><em>(Machine Operation Specification)</em>. It assists the user <strong>MACHINEMANAGER</strong> to set the machine. It also assists the role <strong>PROCESSDS</strong> for the process specification.</td>
</tr>
<tr>
<td><strong>PROCESSDS</strong></td>
<td><em>(Process Design Assistant)</em>. It assists the user to define the process manufacturing. It asks the role <strong>MACHINEOPS</strong> for information about the possibility of performing operations in the machines and the durations of these operations.</td>
</tr>
<tr>
<td><strong>PLANNERASS</strong></td>
<td><em>(Planer Assistant)</em> It assists the user Planner to define the work orders. It also calculates parameters for the system performance measurement.</td>
</tr>
<tr>
<td><strong>ORDERDISPATCHER</strong></td>
<td><em>(Order Dispatcher)</em> It dispatches the orders defined by the Planner.</td>
</tr>
<tr>
<td><strong>DISPLAYER</strong></td>
<td>It displays de system state (orders and machine state) and its evolution.</td>
</tr>
<tr>
<td><strong>PRICECALCULATOR</strong></td>
<td><em>(Order Manufacturing Manager)</em>. It calculates the price of resource usage in each time unit of the planning horizon.</td>
</tr>
<tr>
<td><strong>LOCALSCHEDULER</strong></td>
<td>It schedules the order operations according the prices that the role <strong>PRICECALCULATOR</strong> has calculated.</td>
</tr>
<tr>
<td><strong>ORDMANUFMANAGER</strong></td>
<td><em>(Order Manufacturing Manager)</em>. It manages the execution of operations of each production order according to the local schedule.</td>
</tr>
<tr>
<td><strong>OPEXMANAGER</strong></td>
<td><em>(Operation Execution Manager)</em>. It is responsible of the operation execution in each machine. It controls the physical machine.</td>
</tr>
</tbody>
</table>
### Role detail example

**Role:** MACHINEOpsPC

**Description:** It assists the user MACHINEMANAGER to set the machine. It also assists the role PROCESSDsAss to the process specification by stating whether a machine is able to perform operations and by calculating its duration.

**Protocols and Activities:** ShowGUI (Show Graphic User Interface), RegisterSp (Register Specifications in DB), RegisterOp (Register operation in DB), RequestOpParam (Request to ProcessDsAss for the operation parameters)

**Permissions:** create, read and modify  
*Machine Specification List, Machine Operations List*

**Responsibilities**

**Liveness:**  
\[ \text{REGISTERSpOp} = (\text{AIDMACHINEMANAGER} | \text{REGISTEROp})^w \]

\[ \text{AIDMACHINEMANAGER} = \text{ShowGUI} \cdot \text{RegisterSp} \]

\[ \text{REGISTEROp} = \text{RequestOpParam}^+ \cdot \text{RegisterOp} \]

**Safety:** true

---

- **Protocols (interaction templates)**
- **Operators**
- **Complex activities**
- **Activities**
## Protocol example

<table>
<thead>
<tr>
<th>RequestOpParam</th>
<th>Operation Identifier</th>
<th>Operation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MACHINEOpSPC</strong></td>
<td><strong>PROCESSDsAss</strong></td>
<td></td>
</tr>
</tbody>
</table>

## Protocol list

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Initiator/Responder</th>
<th>Description (Initiator Intention)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestOpParam</td>
<td>MACHINEOpSPC / PROCESSDsAss</td>
<td>Knowing the parameters that define an operation to design the execution and calculate the process time.</td>
</tr>
<tr>
<td>RequestOpProc</td>
<td>PROCESSDsAss / MACHINEOpSPC</td>
<td>Generating the operation execution process in a machine and knowing the process time.</td>
</tr>
</tbody>
</table>
## Protocol list (continued)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Initiator/Responder</th>
<th>Description (Initiator Intention)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestItems</td>
<td>PLANNER / PROCESSDSASS</td>
<td>Knowing the items with a defined process to create manufacturing orders.</td>
</tr>
<tr>
<td>RequestOrManuf</td>
<td>PLANNER / ORDERDISPATCHER</td>
<td>Creating the ORDERMANUFMANAGER to manage the manufacturing order.</td>
</tr>
<tr>
<td>RequestProcess</td>
<td>ORDERDISPATCHER / PROCESSDSASS</td>
<td>Knowing the order manufacturing process to create the ORDERMANUFMANAGER.</td>
</tr>
<tr>
<td>RequestOpEx</td>
<td>ORDERMANUFMANAGER / OpEXMANAGER</td>
<td>Execute an operation.</td>
</tr>
<tr>
<td>RequestLocSch</td>
<td>PRICECALCULATOR / LOCALSCHEDULER</td>
<td>With this protocol, the role PRICECALCULATOR gets local schedules.</td>
</tr>
<tr>
<td>RequestMachEv</td>
<td>DISPLAYER / OpEXMANAGER</td>
<td>Knowing the state changes of the machines to monitor the system. The DISPLAYER has to know the operation start time and conclusion time.</td>
</tr>
</tbody>
</table>
Agent based shop floor control systems

The Gaia methodology

Analysis phase

Design phase

Conclusions
## Agent model

<table>
<thead>
<tr>
<th>Agent</th>
<th>Planner</th>
<th>Process Designer</th>
<th>Coordinator</th>
<th>Order Manager</th>
<th>Machine Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instances</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Number of orders in the system</td>
<td>Number of machines</td>
</tr>
<tr>
<td>Roles</td>
<td>PLANNERASS, PROCESSDSASS, DISPLAYORDERDISPATCHER, PRICECALCULATOR</td>
<td></td>
<td>ORDMANUFMANAGER, LOCALSCHEDULER</td>
<td></td>
<td>OPEXMANAGER, MACHINEOPS</td>
</tr>
</tbody>
</table>

### Design phase

- **Agent model**
- **Acquaintance model**
Design phase

Agent model

- Planner
  - RequestOrManuf
  - RequestLocSch
  - RequestItems

- Coordinator
  - RequestMachEv
  - RequestOpProc

- Process Designer
  - RequestProcess
  - RequestOpParam

Acquaintance model

- Order Manager
  - RequestOpEx

- Machine Manager
Content

- Agent based shop floor control systems
- The Gaia methodology
- Analysis phase
- Design phase
- Conclusions
The Gaia process was the **first phase of a development process** that finished with a successful **JADE implementation**.

- The **process of developing our system was agile and robust**.
- Gaia allow an **incremental development process**,  
- Gaia allow **documenting** the different models  
- Gaia **facilitate the subsequent JADE implementation**.