A Model Transformation Approach to Constructing Agent-oriented Design Models for CPS/IoT Systems

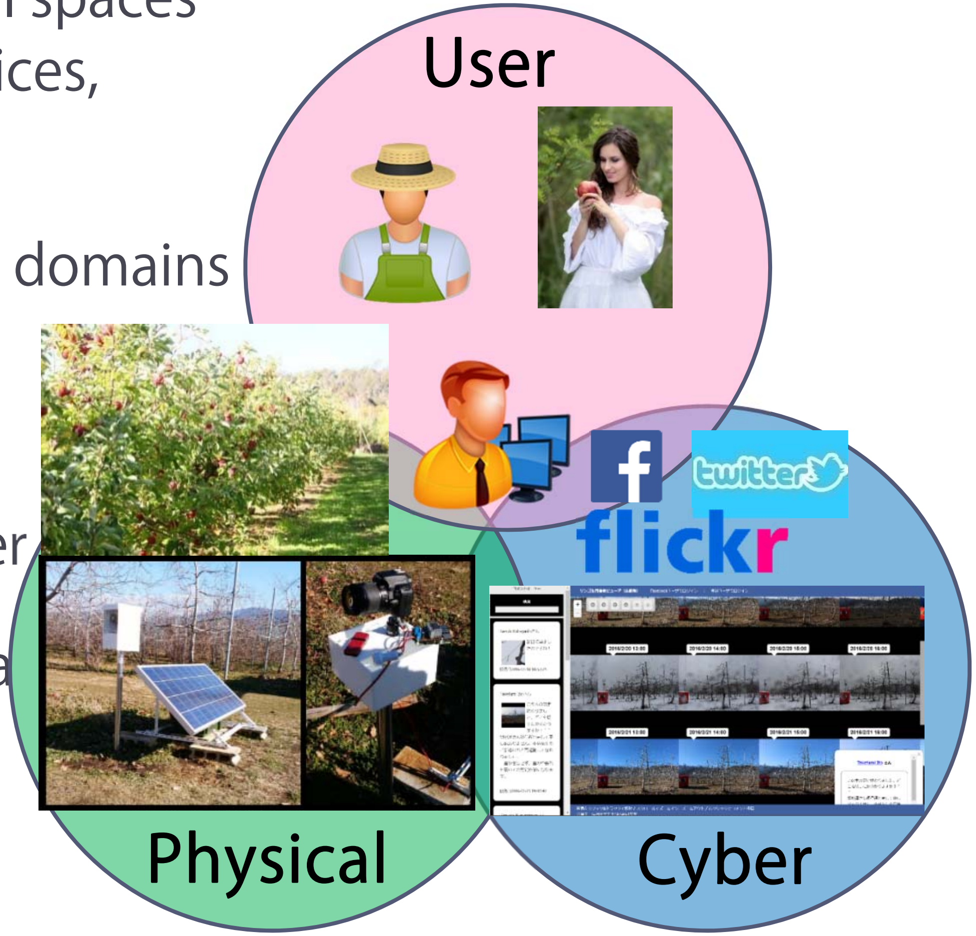
Hiroyuki Nakagawa (Osaka Univ.), Shinpei Ogata(Shinshu Univ.)
Yoshitaka Aoki(Nihon Unisys, Ltd), Kazuki Kobayashi(Shinshu Univ.)

March 30 - April 3, 2020 SAC2020

[Background] CPS, IoT systems

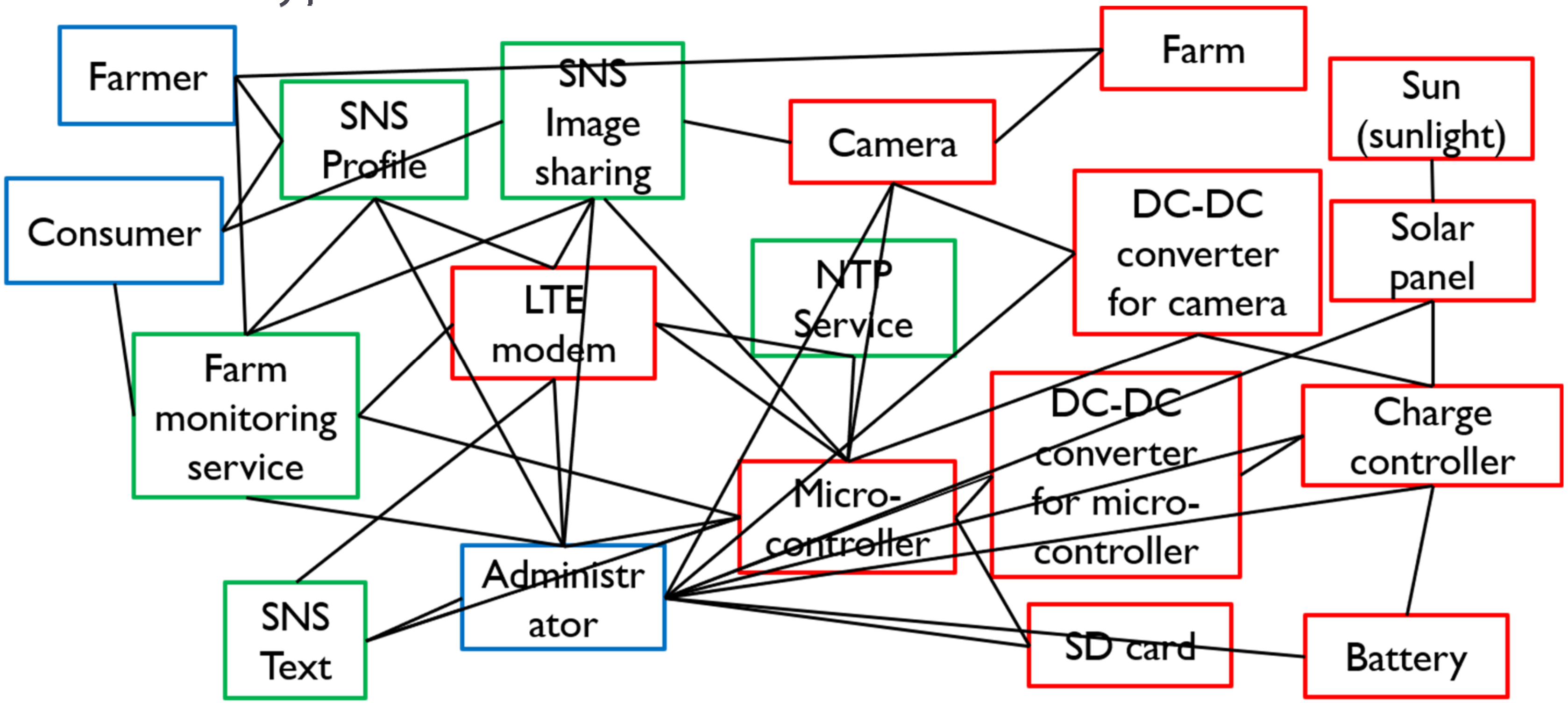
- CPS (Cyber-Physical systems), IoT systems
  - Involve in cyber and physical spaces
  - Components: hardware devices, software components, network devices
  - Rapidly spreading in various domains

- Example: farm monitoring system
  - provides a web-based viewer containing images of crops periodically taken by camera
    - Main Components:
      - camera, solar panel, charge controller, battery, SNS applications, …



Complicated relations: farm monitoring system

- Design for CPS/IoT systems tends to be considerably complicated
  - Devices in physical space and software in cyber space
  - Various types of relations



→ Develop a CPS/IoT system based on the MAS design

Agent-oriented design for CPS/IoT systems

- Develop CPS/IoT system as MAS (Multi-Agent System)
  - Abstraction of components based on autonomy
  - Separation of concerns using organizations
- Gaia [1]: an agent-oriented design methodology
  - Focuses on the early phases of MAS development
    - System analysis, architectural design, detailed design
    - does not impose a limitation on the implementation
  - Many methodologies are based on Gaia
    - ROADMAP, IMPULSE, …
  - Introduces the organization concept
    - Two-step abstraction, i.e., agent and organization

→[GOAL] Construction of an agent-oriented design model for a CPS/IoT system under the Gaia model

[1] F. Zambonelli, N. R. Jennings, and M. Wooldridge, "Developing multi-agent systems: The Gaia methodology", ACM Trans. on Software Engineering and Methodology, Vol. 12, No. 3, 2003.

Design process in Gaia

(1) Define the environment of MAS

- Listing resources and their access permissions.

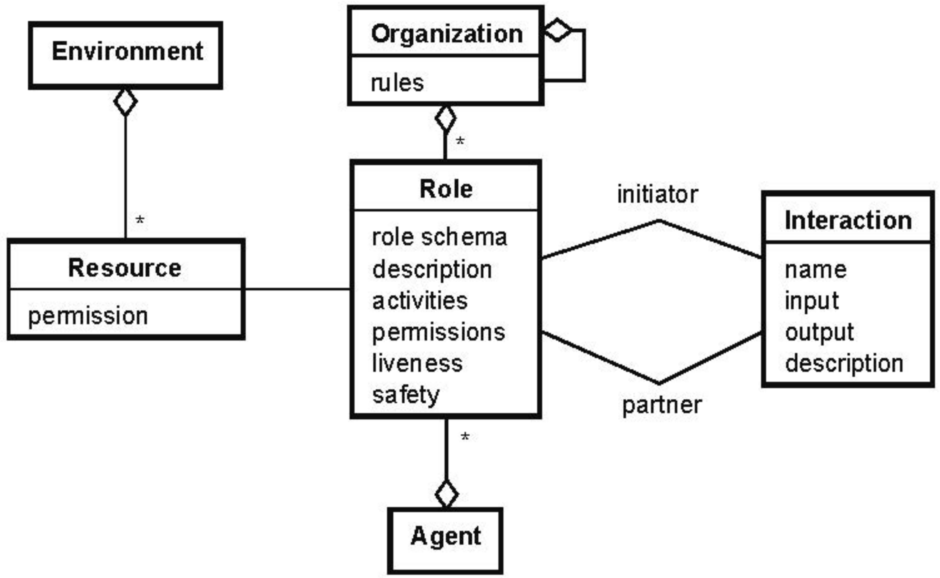
(2) Extract roles and construct the role model

- Roles are assigned to agents

(3) Identify interactions between roles

(4) Find organizations and define their rules (constraints).

Gaia metamodel



Gaia role model

Attributes	Explanations
Role Schema	Name of the role. e.g.)Tracker
Description	Summary. e.g.) Tracker chases the user by using sensing data and a reasoning mechanism.
Activities	Tasks that the role can complete by itself. e.g.) FindTarget, Migrate
Protocols (Interactions)	Relevant interactions with other roles. e.g.) RequestTracking, RequestSensingData,…
Permissions	Access permissions to environmental resources. e.g.) changes Log
Liveness	Operational sequence, which is composed of protocols and activities e.g.) L1 =RequestTracking. RequestSensingData. Migrate
Safety	Constraints to be satisfied at all times e.g.) $\Box\Diamond(Area_{Tracker} = Area_{Target})$

Three difficulties in designing agent model

- Strength
  - Abstraction of components based on autonomy
  - Separation of concerns using organizations

- However, three difficulties exist in the model construction
  - Difficulty 1: Role extraction
    - Which agents (roles) should we define and implement?
  - Difficulty 2: Role model construction
    - The role model construction requires adequate understanding of the role.
  - Difficulty 3: Organizational model construction
    - Organizations can not be defined until we understand individual roles' activities.

MAS design using model transformation

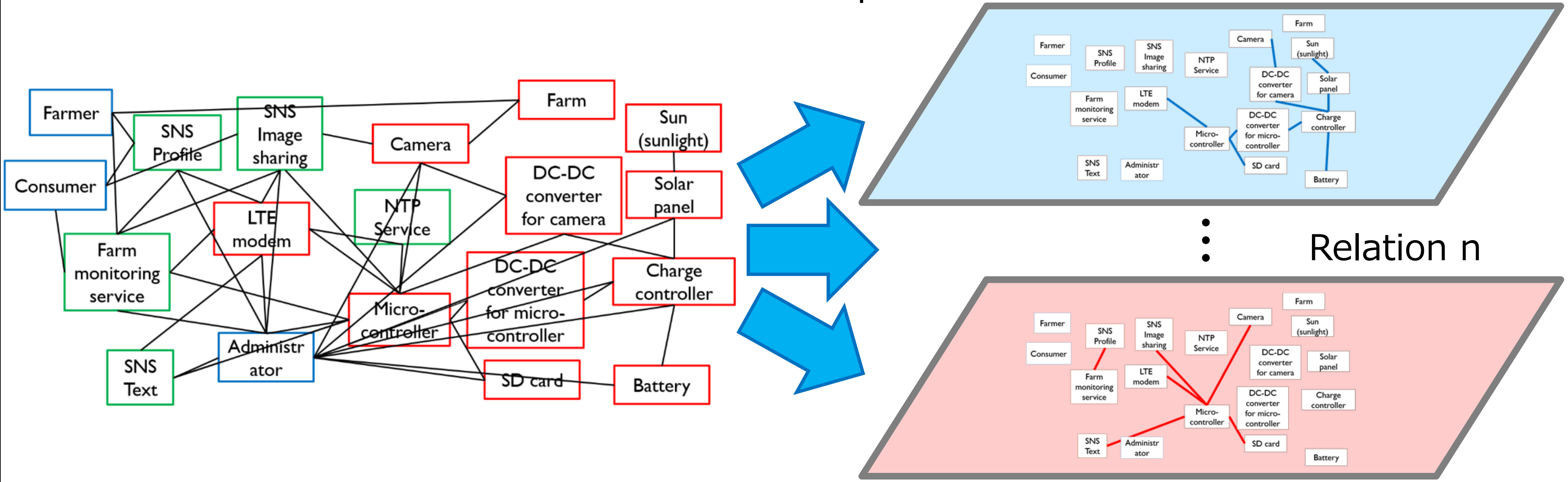
- Three difficulties in MAS design:
  - Difficulty 1: Role extraction
  - Difficulty 2: Role model construction
  - Difficulty 3: Organizational model construction

- Introduce a model transformation technique
  - [Source] ???
  - [Target] MAS model compliant with Gaia model

Source model: TORTE [2]

Architectural model/modeling method for CPS/IoT systems

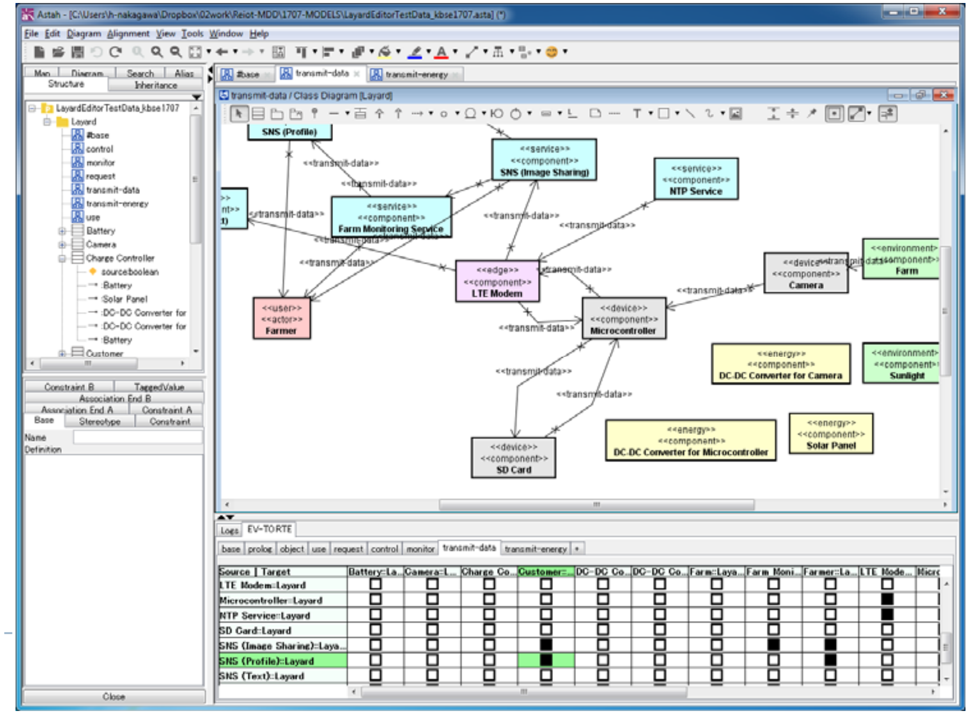
- Defines representative types of components and relations
  - Component: user, device, service, edge, energy, environment
  - Relation: use, request, control, monitor, transmit-data/energy
- Has a modelling editor
  - Layered views corresponding to individual relation types
    - Realizes a visualization based on separation of concerns



 [2] S. Ogata, H. Nakagawa, Y. Aoki, K. Kobayashi, Y. Fukushima, "A tool to edit and verify IoT system architecture model", MODELS 2017, 2017.

TORTE editor

- A UML editor implemented as a plugin of Astah [3]
- Main characteristics
  - Visualization using layered views
    - Layers correspond to types of relations, one to one
  - Verification mechanism using Prolog rules
  - Template SMV code generation for behavioral verification



 [3] http://astah.net/

MAS design using model transformation

- Three difficulties in MAS design:
  - Difficulty 1: Role extraction
  - Difficulty 2: Role model construction
  - Difficulty 3: Organizational model construction
- Introduce a model transformation technique
  - [Source] TORTE model
  - [Target] MAS model compliant with Gaia model

Overcome Difficulty 1: Role extraction

- Primary role (agent) extraction guideline:
  - extract a component *c* as a primary role of a layer l, if and only if c satisfies all of the following conditions:
    - (1) Component c has a relation to another component in the layer l
      - $\exists c2 \in$ Comp: r(l, c, c2)
    - (2) Component c is not an actor, such as a user or administrator
    - (3) Component c should equip the autonomy for satisfying the relation extracted in Condition

- Primary roles extracted in the farm monitoring system

Layers (organizations)	Primary roles
control	Micro controller
monitor	Micro controller, Charge controller, Camera
request	Micro controller, Farm monitoring service
transmit-data	Micro controller, SNS (for profile/image sharing/text), Farm monitoring service, Camera
transmit-energy	Micro controller, Charge controller

Overcome Difficulty 2: Role model construction

Attributes of Gaia role model	Corresponding elements in the TORTE model
Role Schema	*component*. name
Description	Task summary of *component* in the layers where the component is regarded as a role.
Activities	Important methods (functions) of *component*
Protocols (Interactions)	*component's* important associations (relations) to other components.
Permissions	A set of access permissions of each object that has an association link to *component*.
Liveness	Operational sequences of *component*. A sequence is composed of protocols and activities.
Safety	Constraints imposed on *component*.

Constructed role model for **Charge controller**

Attributes	Values
Role Schema	Charge controller
Description	This role checks other components' voltage and provides energy supply if necessary. If other role request the energy supply, this role supplies energy with managing supply and demand balance.
Activities	BalanceSupplyDemande
Protocols (Interactions)	CheckMCVoltage, EnergySupplyToMC, RequestEnergySupply, CheckCameraVoltage, EnergySupplyToCamera
Permissions	**reads** SolarPanel.ElectricityOutput **writes** Battery.EnergyLevel
Liveness	P_MC1 = CeckMCVoltage. BalanceSupplyDemand. EnergySupplyToMC P_MC2 = RequestEnergySupplly. BalanceSupplyDemand. EnergySupplyToMC P_Cam = CheckCameraVoltage. BalanceSupplyDemand. EnergySupplyToCamera
Safety	Battery.EnergyLevel ≥ 0, MC.Voltage=Vc1, Camera.Voltage = Vc2, LTE modem.Voltage=Vc3

Overcome Difficulty 3: Organizational model construction

- We have to define organizations and their members and rules

- Solution:
  - Organization: assigned to each layer of TORTE
    - TORTE has six representative layers → six organizations
    - Designer can define new layers in the TORTE model
  - Members: roles extracted at the corresponding layer
    - Primary roles and partners of interactions
  - Rules: defined with reference to the liveness/safety properties of the members' role models

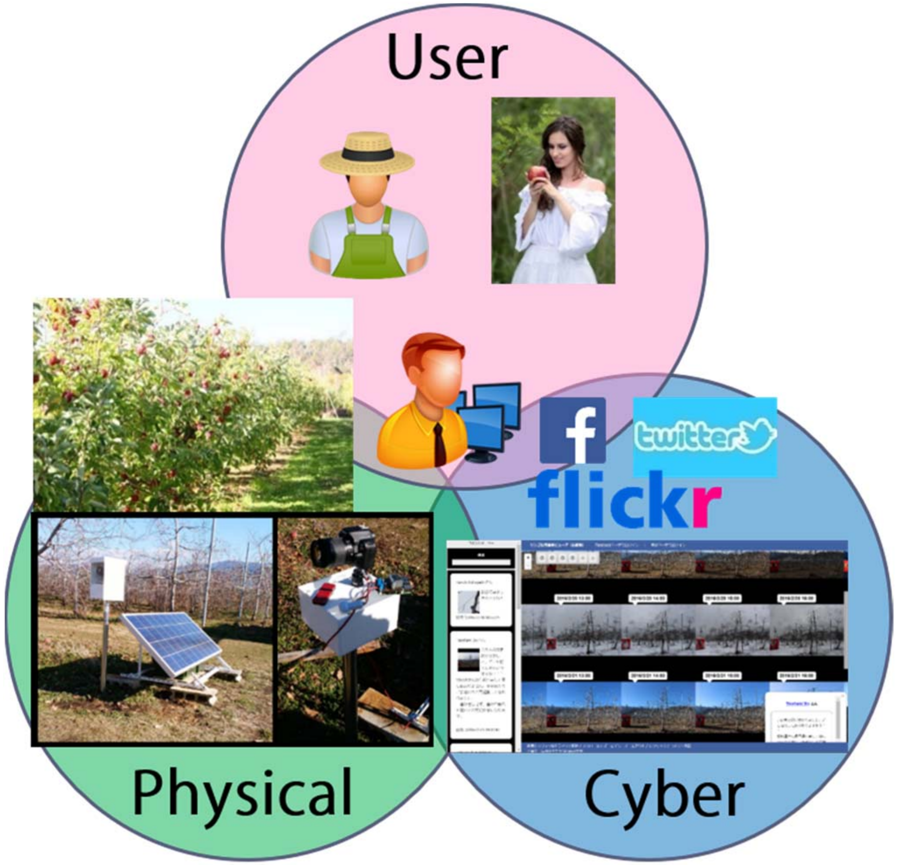
Organization corresponding to the transmit-energy layer

Attributes	Values
Name	transmit-energy
Members	MC (Micro controller), Charge controller, Camera
Rules (Liveness)	- CheckMCVoltage→EnergySupplyToMC - RequestEnergySupply→EnergySupplyToMC - CheckCameraVoltage→EnergySupplyToCamera
Rules (Safety)	- Battery.EnergyLevel ≥ 0 - MC.Voltage=Vc1 - Camera.Voltage=Vc2 - LTE modem.Voltage＝Vc3

Case study: Farm monitoring system

- Apply the model transformation to the farm monitoring system [4]
  - [Source] TORTE model:
    - #components: 18
    - #relations: 51
    - #layers: 6
  - [Target] Gaia model:
    - #role models: 7
    - #interactions: 14
    - #organizations: 6

  Gaia model was systematically constructed



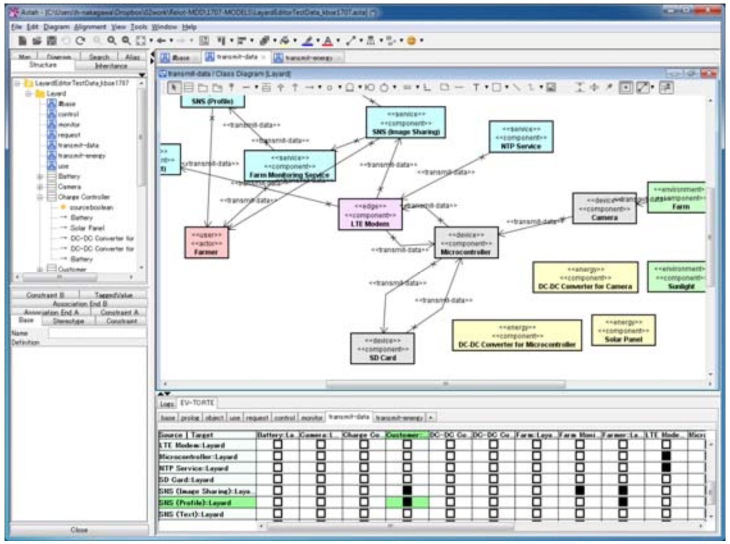
[4] H.Genno, K. Kobayashi, "Apple growth evaluated automatically with high-definition field monitoring images", Computers and Electronics in Agriculture, Elsevier, 2019.

Discussion

- Difficulty 1: Role extraction
  - Semi-automatically determined primary roles and other roles
  - All of the extracted roles are important in the actual system
- Difficulty 2: Role model construction
  - Most of the attributes in the Gaia role model can be formally generated
  - Others are determined with reference to the TORTE model
- Difficulty 3: Organizational model construction
  - Organizations and their members can be automatically determined
  - Rules can be defined with reference to the corresponding views
  - Some rules can be obtained from liveness/safety of role models

Conclusions

- A model transformation technique to construct MAS design model for CPS/IoT systems
  - Uses the TORTE model to construct the Gaia model
- Experimentally constructed a Gaia model for a farm monitoring system
  - Enable to construct appropriate Gaia model



- Future work
  - Help semi-automatic model construction
    - Liveness/safety in role model
  - Support implementation
    - Model transformation to code
    - Provide programming framework
  - Case studies on complicated domains