RoDaFlow: A Framework for Development of Dataflow Network Agents in Smart-M3 with Substitution Method

Denis Laure
den.a.laure@gmail.com

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University
Primary agent

Initialization
1. join the smart space
2. insert description triples
3. insert protection for description, output and state triples
4. calculate and inserts initial output and state triples
5. subscribe to input triples

Operation
- Receive new input triples
- Calculate new output and state triples
- Update calculated triples

Denis Laure
P.G. Demidov Yaroslavl State University

Yaroslavl FRUCT Lab
Substitute agent

Initialization
1. join the smart space
2. insert description triples
3. insert protection for description triples
4. subscribes to triple that indicates what primary agent is substituted by this one (‘Substitutes’ triple)

Operation

Receive new input triples → Receive substitution program → Calculate new output and state triples → Subscribe for input triples → Update calculated triples

Denis Laure

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University
Choosing the Substitute Agent

Substitute Agent
- type = signal_processing

Primary Agent
- type = calculation

Substitute Agent
- type = calculation

Substitute Agent
- type = condition
Choosing the Substitute Agent

Substitute Agent
  type = signal_processing

Primary Agent
  type = calculation

Substitute Agent
  type = calculation

Substitute Agent
  type = condition

Denis Laure
Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University
Choosing the Substitute Agent

Primary Agent
- type = calculation

Substitute Agent
- type = calculation

Substitute Agent
- type = signal_processing

Substitute Agent
- type = condition
Motivation

• The behavior and basic operations of the agents are always the same
• The agents differs only in their programs
RoDaFlow Framework

- Allows to create dataflow network agents for Smart-M3 platform
- Allows to create agents by implementing only their programs
- Created agents support substitution mechanism

- Written in Java
- Uses Java KPI
Implementing Primary Agent

To create primary agent:
- Implement AgentProgram interface
- Create instance of PrimaryAgent class
- Pass implemented agent program to created instance
- Call the joinSIB method on the PrimaryAgent instance
Implementing Substitute Agent

To create substitute agent:
• Implement SubstituteAgentProgram interface
• Create instance of SubstituteAgent class
• Pass implemented substitute agent program to created instance
• Call the joinSIB method on the SubstituteAgent instance
RoDaFlow Framework Benefits

• Saves the agents developer's time
• Simplifies the development of the agents
• Hence simplifies the development of dataflow network based systems
• Does not require from the developer any additional knowledge of substitution mechanism implementation
• Allows to create agents for:
  • Popular desktop platforms
  • Mobile devices
  • Oracle's Internet of Things platform
Thank You!

Q&A

RoDaFlow framework homepage: http://yar.fruct.org/projects/rodaflow

Denis Laure
den.a.laure@gmail.com

Yaroslavl FRUCT Lab
P.G. Demidov Yaroslavl State University
Agent Classes

I AgentProgram
- `InTriplesTemplates : Collection<Triple>`
- `StateTriplesTemplates : Collection<Triple>`
- `OutTriplesTemplates : Collection<Triple>`
- `calculateOutTriples() : Collection<TriplesCouple>`
- `calculateInitialOutTriples() : Collection<Triple>`

C PrimaryAgent
- `joinSIB() : void`
- `shutdown() : void`
- `suspendOperation() : void`

I SubstituteAgentProgram
- `parseProgramBody(String) : void`
- `calculateOutTriples() : Collection<TriplesCouple>`

C SubstituteAgent
- `joinSIB() : void`
- `shutdown() : void`