

SYSTEM-OF-SYSTEMS THAT ACT **LOCALLY** FOR OPTIMIZING **GLOBALLY**

EU FP7 - SMALL/MEDIUM-SCALE FOCUSED RESEARCH PROJECT (STREP)
FP7-ICT-2013.3.4: ADVANCED COMPUTING, EMBEDDED AND CONTROL SYSTEMS
D) FROM ANALYZING TO CONTROLLING BEHAVIOUR OF SYSTEM OF SYSTEMS (SOS)

3rd Consortium meeting (Munich)

The Local4Global System (product)

February 10-11, 2014

Local⁴Global

Contact Information

For information regarding this Project: Check the Project Web-Site: <http://local4global-fp7.eu>

Participants	
1	CERTH - Centre for Research and Technology
2	ETHZ – Eidgenössische Technische Hochschule Zürich
3	RWTH – RWTH Aachen University
4	IK4 – IK4 TEKNIKER
5	TRV – TRANSVER GmbH
6	TUC – Technical University of Crete
7	TUM – Technische Universität München

Project Acronym: Local4Global

Project Number: 611538

Project Start Date: October 2013

Duration: 3 Years

Funded by: EU FP7

Program Name:

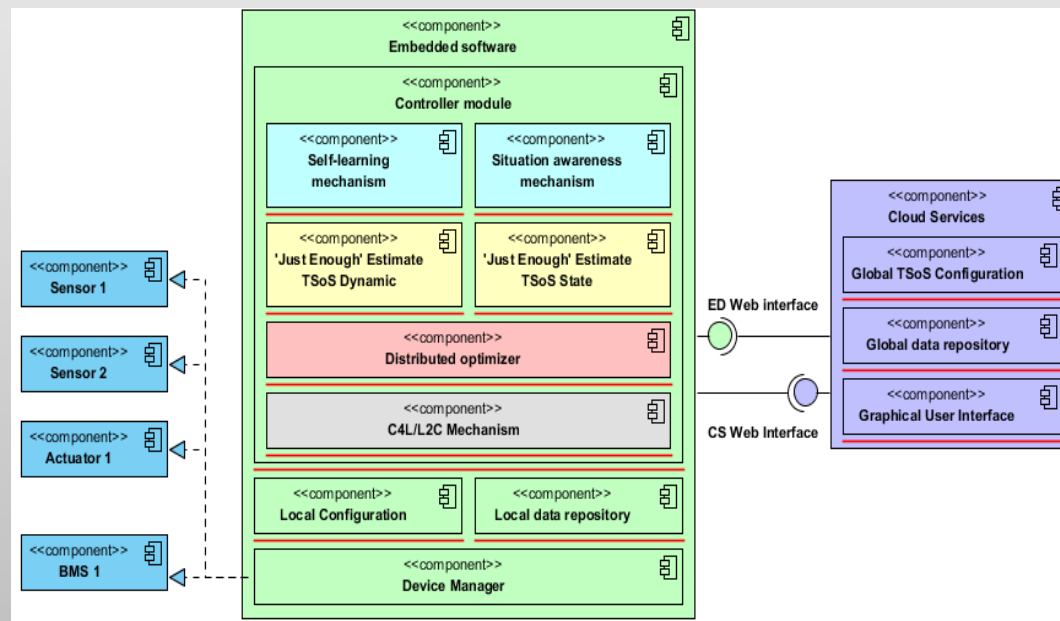
EU FP7 - SMALL/MEDIUM-SCALE FOCUSED
RESEARCH PROJECT (STREP)
FP7-ICT-2013.3.4: ADVANCED COMPUTING,
EMBEDDED AND CONTROL SYSTEMS
D) FROM ANALYZING TO CONTROLLING
BEHAVIOUR OF SYSTEM OF SYSTEMS (SOS)

Presentation Overview

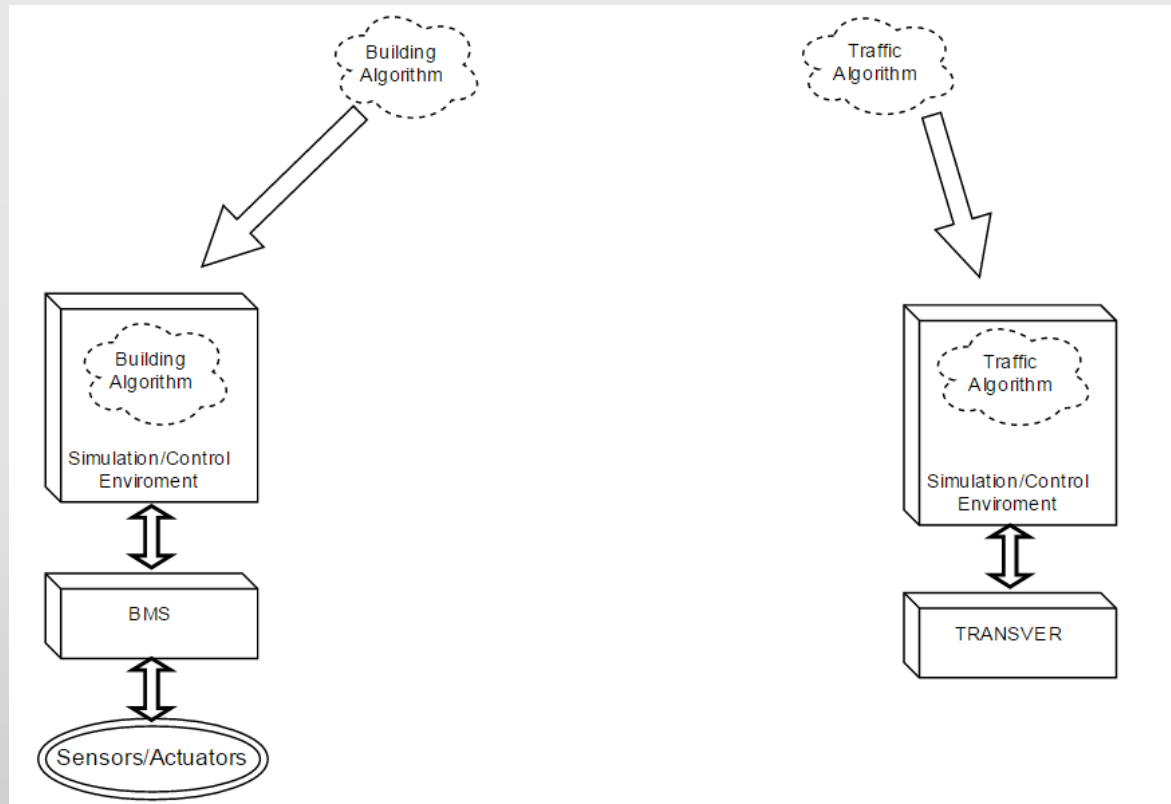
- Previous work
- Algorithm Integration
- Resulting products
- Common Interfaces
- Product testing
- Work to do

Previous work

- Product architecture design
- Product deployment discussions
- Algorithm simulation activities

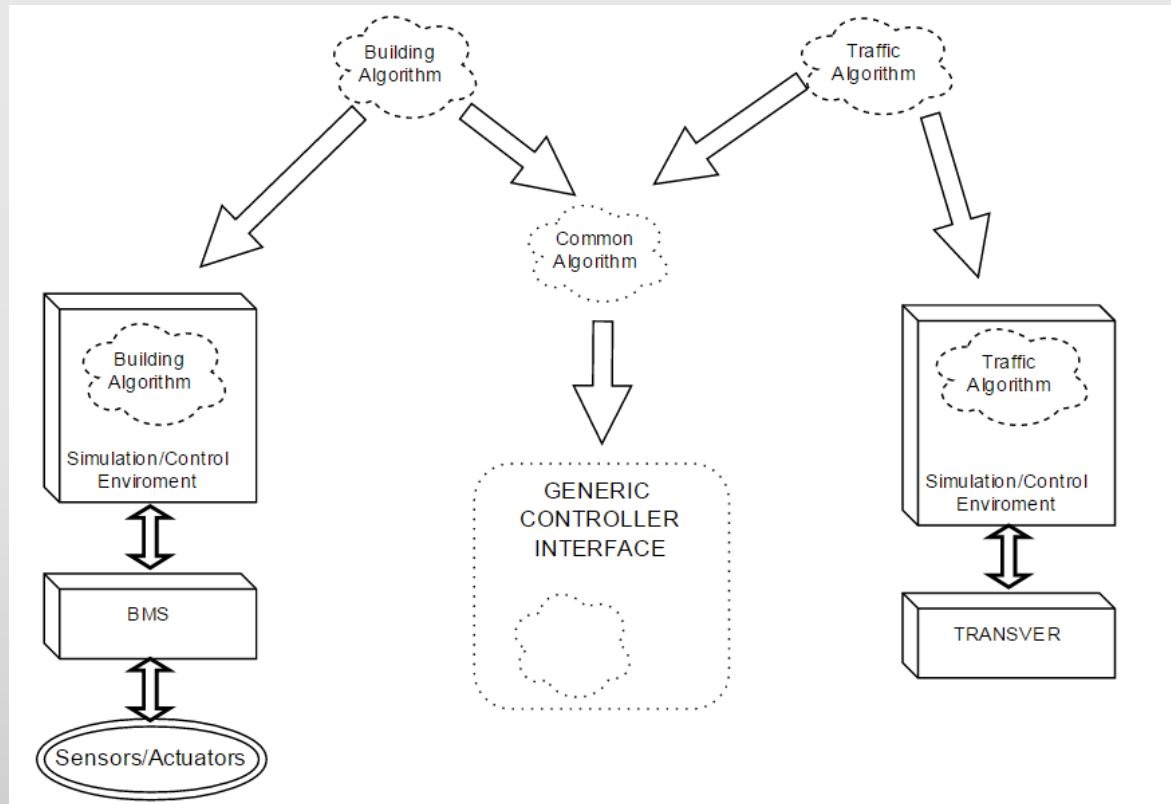


Algorithm integration plan



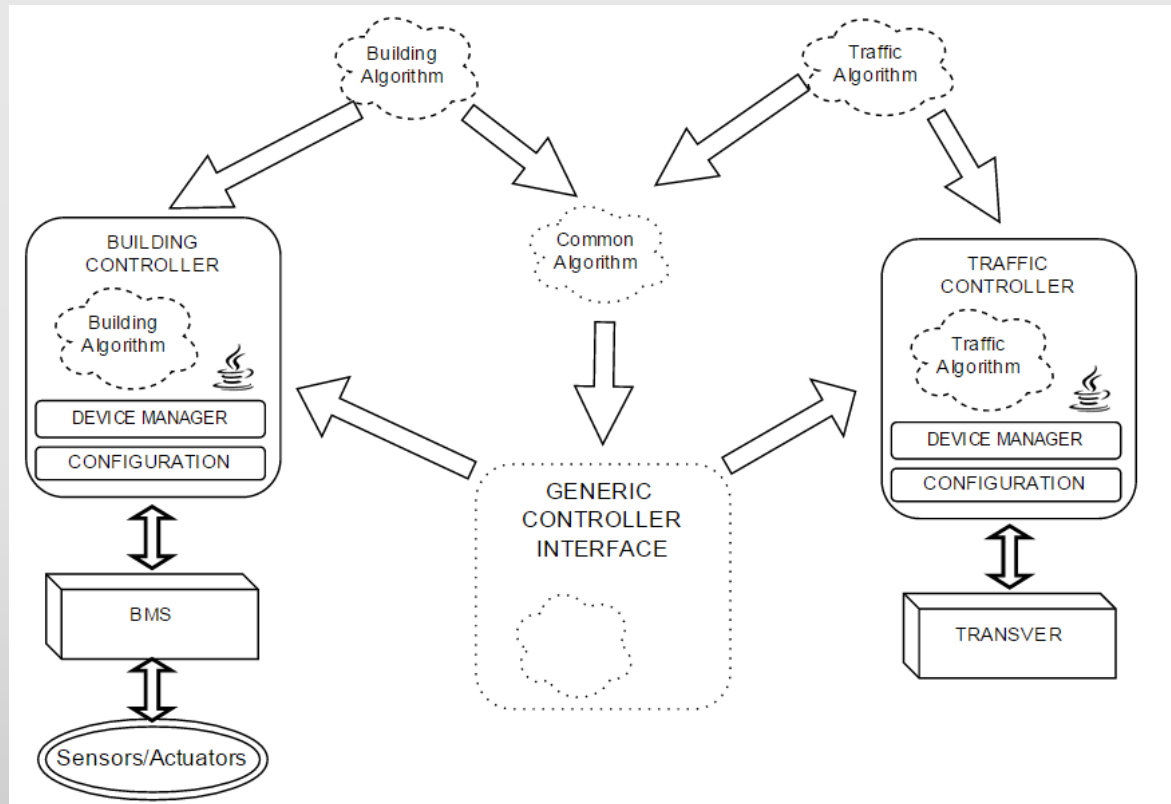
- Custom algorithms for each of the use cases
- Implement the algorithms in the existing simulation/control environments
- Already in progress

Algorithm integration plan



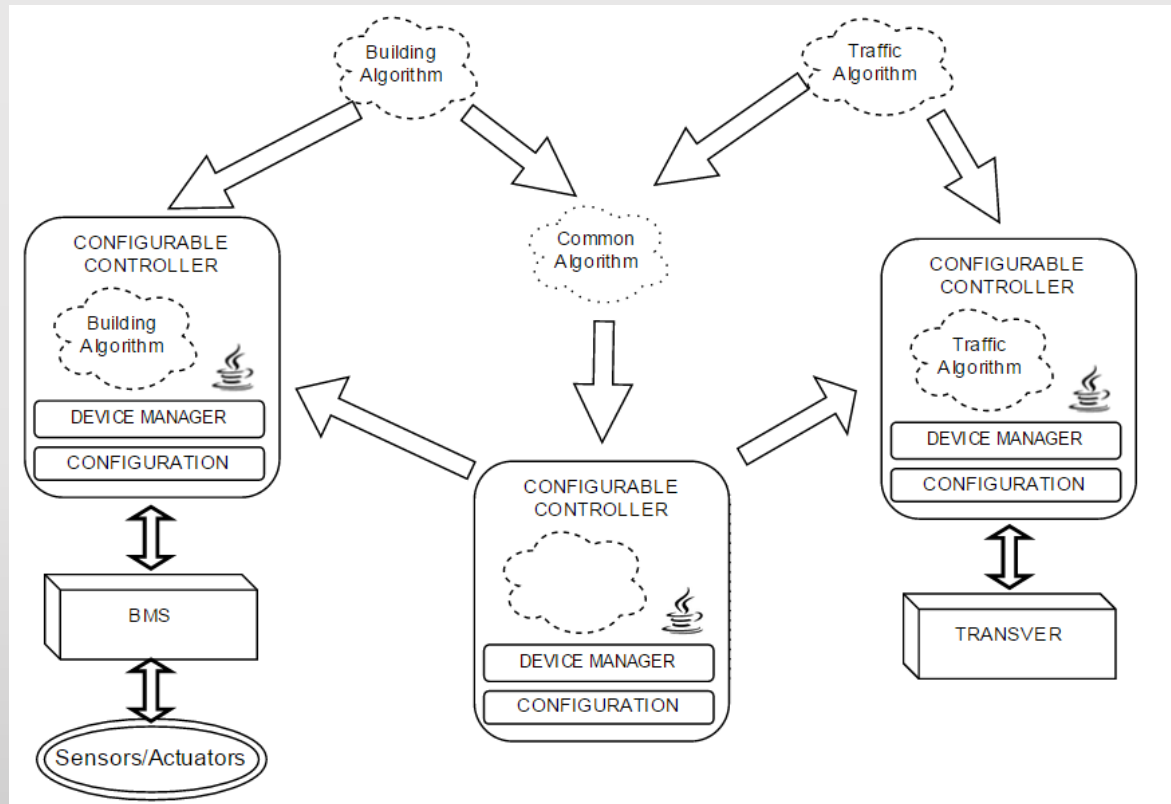
- Find common attributes of both algorithms
- Generic controller interface (a set of defined functions)

Algorithm integration plan



- 2 different custom controller for each of the use cases, using the same interface
- Controllers would replace existing the control system

Algorithm integration plan



- Generic configurable controller
- The same controller in both use cases, each one with it's specific algorithms configured

Resulting products

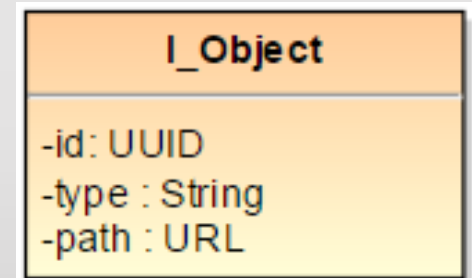
1. System specific libraries to be used inside control/simulation environment.
 - Validates the L2C/C4L algorithm/mechanism
 - Tested in each use case environment.
 - Configurable for similar use cases.
2. System specific Java controller aimed to replace/support existing control/simulation environment.
 - Integrate the libraries from product 1 into a Java Object
 - Common interface to different use cases
 - Deployable in any system able to run a Java objects(JVM)
3. Generic Java controller configured to replace/support existing control/simulation environment.
 - Most complex product.
 - Highly adaptable and configurable to different use cases

Common Interfaces

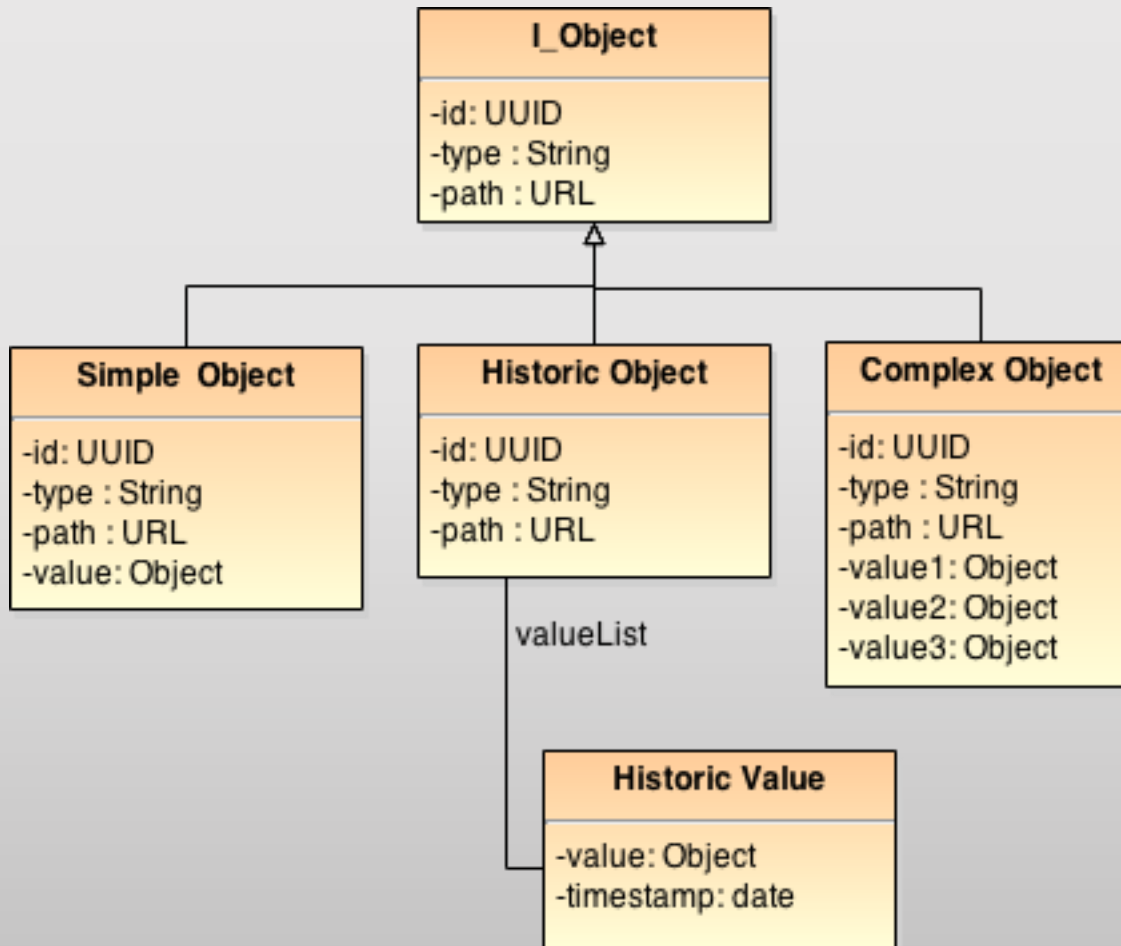
- Reason: The Local4Global System has been conceived as a generic product, able to adapt to different frameworks.
- Interfaces provide a good environment to implement the different algorithms by defining a set of functionalities independent of their final implementation:
 - Data access (Local)
 - Interaction with other controllers (Global)
 - Control & configuration management
- **As simple as possible**

Common Interfaces – Objects

- Objects define any configurable ‘thing’ in a controller: control variables, data sources, file paths...
- The simplest way to define any object in the system is via this four attributes:
 - **Id:** Unique identifier.
 - **Type:** This attribute will help in both describing, accessing and parsing the object and its value.
 - **Path:** Used along with *type* to access the object in both reads and writes
- Additional attributes can be defined in specific implementations

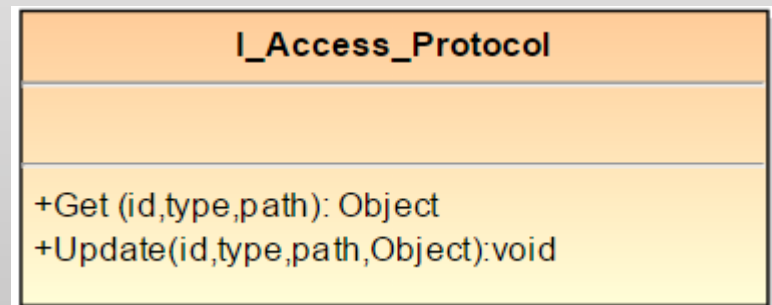
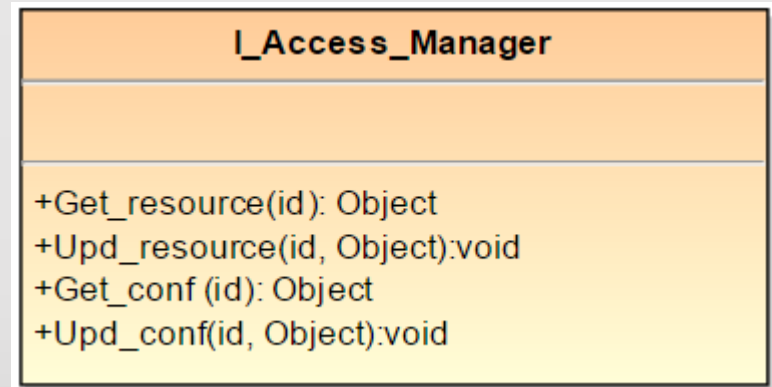


Common Interfaces – Objects (examples)



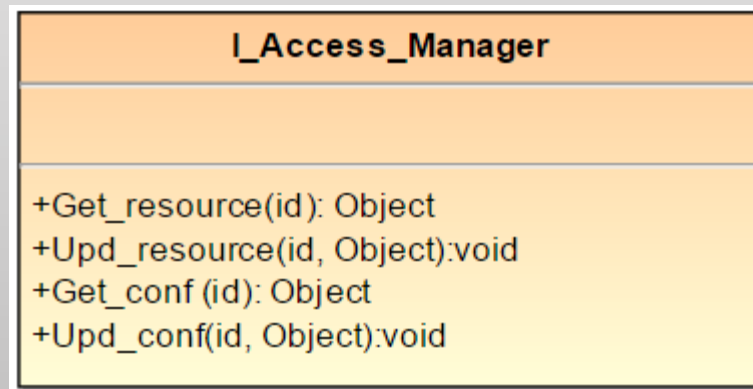
Common Interfaces – Access Manager

- Access Interface will be used by the controller **get** or **set** objects.
- Two different scopes have been defined:
 - **Resources** refer to data that the controllers consume and/or produce
 - **Configuration** refers to the controllers' internal variables.
- Implementations will use the **type** and **path** attributes of the object to choose the specific access protocol.
- Controllers will **not define** new objects. All the existing objects are pre-defined and configured.



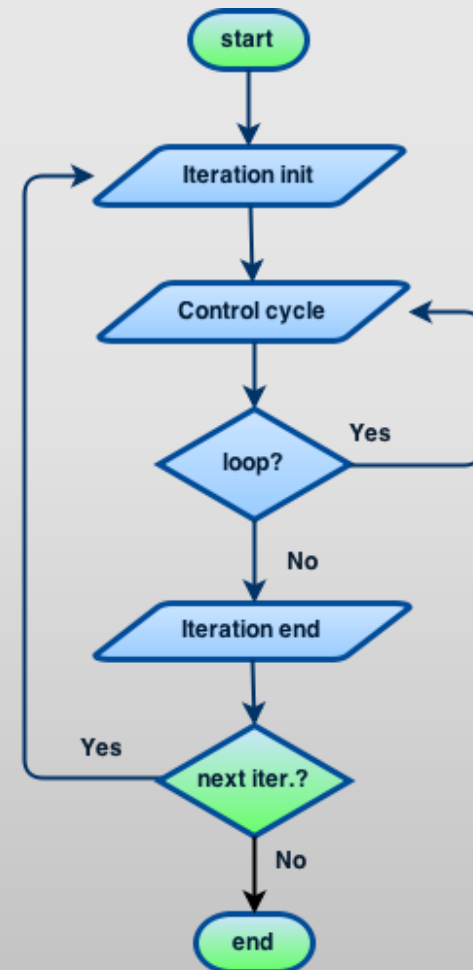
Common Interfaces – External Communications

- Controller will interact with other controllers and services
- The basic functions are the same as the access manager
- Depending the implementation, external agents can access data and/or configure controller variables. The petitions should go through a **security layer**.
- Additional functions can be defined to offer additional functionalities.



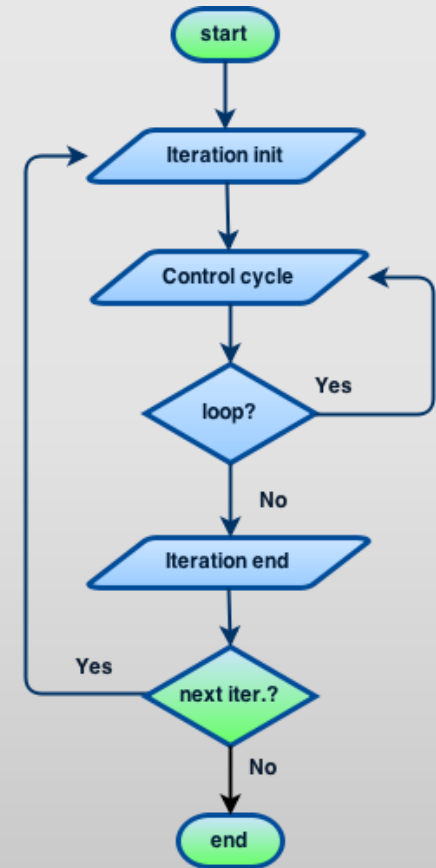
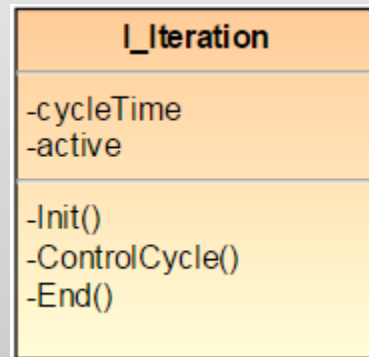
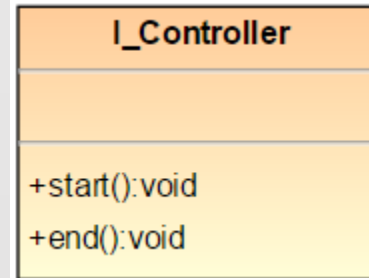
Common Interfaces – L4G Algorithm

- **Inputs and outputs:** Resources and configuration variables, provided by the access interface.
- The algorithm repeats a set of task for each iteration :
 - Init:
 - The algorithm initializes the different algorithm and control variables
 - Control Cycle:
 - Context information is gathered
 - A set of actions are generated and executed based on the control logic.
 - Conditions for next cycle are assessed
 - End:
 - At the end of the iteration, global knowledge is used to ready the algorithm variables for the next iteration

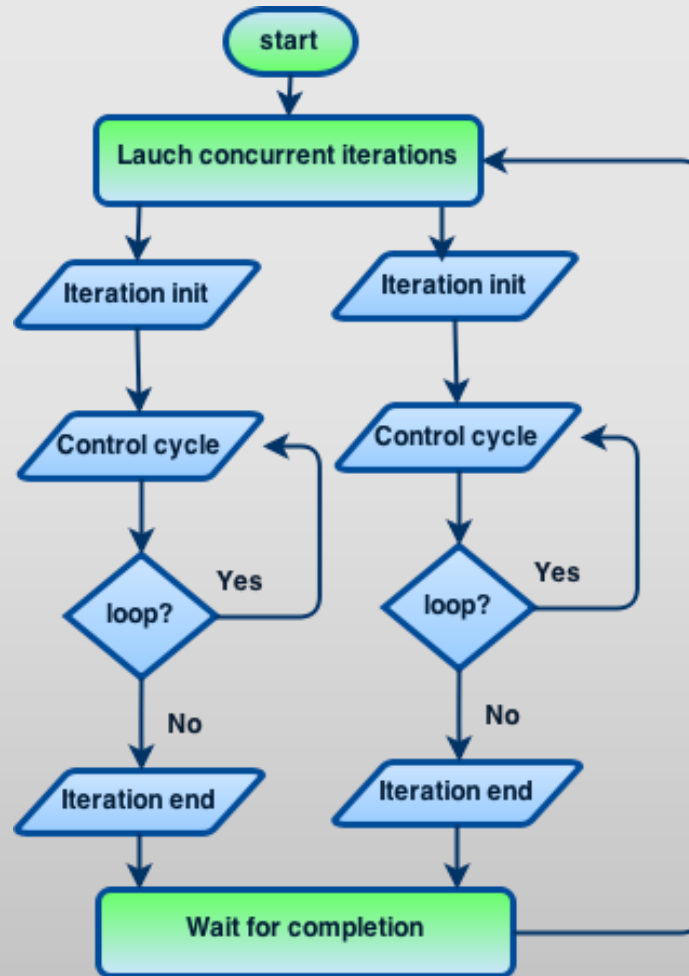


Common Interfaces – L4G Algorithm

- Two interfaces have been defined:
- Controller:
 - Acts as the main class
 - Will manage the different iterations
- Iteration:
 - Will implement different iterations, each one with a initialization, control cycle and end.



Common Interfaces – L4G Algorithm



Common Interfaces – Global repository

- In this approach every controller knows where access the data it is looking for, using the **path** attribute.
 - Different protocol implementations will allow to access sources like other controllers, local files, web services...
- The global repository is not necessary: If the proper communication protocols are implemented, any controller can be accessed
- But could be useful for:
 - Security: Simplified the security layer in the controller.
 - Protocol simplicity: Each controller would implement only one
 - Management: Global level queries could be addressed.
 - Technical limitations: Some controllers may not be able to keep persistent data.

Developing Java Controllers

- First algorithm approaches concerns Windows native libraries and programs (DLL and EXE)
- Can be directly used by control and simulation environments
- Can be directly called from a Java object, but this requires the object to be deployed in a Windows environment
- Alternatives
 - Recompile code (if possible)
 - Rewrite libraries to pure Java (also if possible)
 - Others?...

Testing Java Controllers

- Replacing the L4G controllers by the existing control environments is not always possible
- *Approach*: Feed the controllers the same data used for the control/simulation environment test scenarios and compare the responses.
- Should check:
 - Resulting control actions
 - Response time
 - Technical requirement differences
 - Power consumption
- Could also be deployed as an “advisor” instead of a controller, producing advised course actions instead of direct control actions

Work to be done

- Continue with simulations.
 - Simulation scenarios' used data and obtained results should be recorded for further software testing and validation
- Analyze library integration alternatives
- Component development
 - First interface implementations
 - Component integration
- Define testing and use case deployments
- Support and Feedback

Thank you

Questions?