**3GPP TSG-SA5 Meeting #156S5-243537**

**19 - 23 August 2024, Maastricht, Netherlands**

**Source: Nokia**

**Title: Rel-19 pCR TR28.867 Align CCL conflict management**

**Document for: Approval**

**Agenda Item: 6.19.4**

# 1 Decision/action requested

***The group is asked to discuss and agree on the proposal.***

# 2 References

[1] 3GPP TR 28.867: “Closed control loop management” v0.3.0

# 3 Rationale

Use cases 5.6 and 5.7 in this TR are overlapping as they both are concerned with CCL conflicts management. This pCR aggregates the two into a single use case with a single solution approach and a single baseline solution on CCL coordination.

# 4 Detailed proposal

|  |
| --- |
| **First Change** |

5.6 Use case 6: CCL conflicts management

5.6.1 Description

5.6.1.1 CCL Conflicts

Multiple CCLs could co-exist and concurrently act within the same environment. The CCLs can affect one another, in the worst cases leading to conflicts. The different kinds of conflicts are summarized by Table 5.6.1-1.

5.6.1-1: Types of potential conflicts among CCL instances for goals g1, g2 and g3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conflict Type | Description | CCL-A | CCL-B | Comments |
| Target Conflict | For CCLs C1 and C2, when same at least 1 target of a goal is present in both CCL asking for different outcomes on that target on same controlled entity (ME1). | Control Scope: ME1  Goal targets:   * Load > 90% (to maximize resource utilization) * latency < 10ms | Control Scope: ME1  Goal target:   * Load < 90% (to avoid congestion) | Conflict among the targets within the goals - due to different required target outcomes |
| Action Conflict | For CCLs C1 and C2, when both C1 and C2 is trying to configure the same characteristics of same target entity (gNB-g1) in contradiction. | **Example 1** | | Conflict due to configuration actions at execution step because both CCL want different contradicting value for a particular characteristic of gNB-g1.  Effect: even when executed at different times, the value may ping-pong continuously. |
| Goals target:   * Throughput > 10gbps   Actions:   * Target Entity: gNB-g1 * Target Change: scale-out virtual resource | Goals target:   * EC is < 10KVA   Actions:   * Target Entity: gNB-g1 * Target Change: scale-in virtual resource |
| **Example 2** | |
| Goal target:   * HO failure is < 2%   Actions:   * Target Entity: gNB-g1 * Target Change: set CIO to a small **positive** value{to guarantee HOs with low chances of HO failure} | Goal target:   * Load < 80%   Actions:   * Target Entity: gNB-g1 * Target Change: set CIO to a small negative value [to advance HOs and move load to other cells] |
| Indirect target conflict | For CCLs C1 and C2, when C1 [optimize handover] and C2 [minimize interference] have different goals but the actions of C1 affect the goals of C2 | Goal target:   * HO failure is < 2%   Actions:   * Target Entity: gNB-g1 * Target Change: reduce CIO {to reduce chances of HO failure} | Goal target:   * SINR > 10dB   Actions:   * Target Entity: gNB-g1 * Target Change: lower antenna tilt | By reducing antenna tilt to minimize interference C2 affect the HO goal target of C1 |
| Action Execution Time Conflict | For CCLs C1 and C2, when both C1 and C2 are trying to configure the same characteristics of same target entity (gNB-g1) in contradiction. | Goals:   * Throughput > 10gbps   Actions:   * Target Entity: gNB-g1 * Target Change: scale-out * Target Time: 04:00 | Goals:   * EC is < 10KVA   Actions:   * Target Entity: gNB-g1 * Target Change: scale-in * Target Time: 04:00 | Conflict due to the time of executing the configuration actions at the execution step |
| Scope conflict | For CCLs C1 and C2, C1 and C2 have different goals and actions but their scopes are overlapping – e.g. C1’s control scope (i.e. the controlled entities in the network) is part of C2’s measurement scope (i.e. the measured entities in the network) | Measurement scope: cells g1  Control Scope: g1  Goal targets:   * EC/bit is < 1WA   Actions:   * Target Entity: gNB-g2 * Target Change: switch off g2 | Measurement scope: cells g1, g2, g3, g4  Control Scope: g2  Goals:   * Load < 80%   Actions:   * Target Entity: gNB-g2 * Target Change: change CIO | By switching off g2, C1 affects the scope which C2 reads for its load distribution measurements |

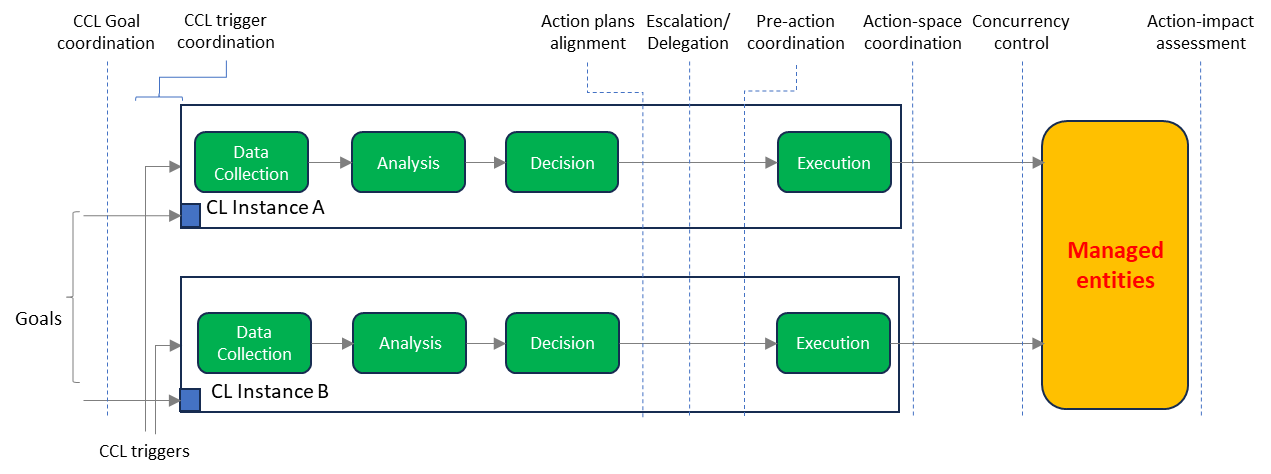
The CCL may detect or observe events that identify the possibility of any one of the above conflicts. The conflict can be avoided using some information or the policies (e.g., priority) provided by the consumer. If the conflict actually occurs, the CCL MnS producer should support services to inform MnS consumers the confirmed detected conflicts. This may also include informing MnS consumer about the potential conflict.

5.6.1.2. CCL conflicts management and coordination interactions

The coordination of CCLs includes the management services needed to detect, resolve, or avoid conflicts among goals and their targets, , control scopes or actions of the CCLs. To address the different conflict situations, coordination capabilities could be required for the following scenarios:

* Capabilities to identify different interaction types between CCLs such as cooperation (positive interaction), conflict (negative interaction) or dependency (neutral interaction).
* Capabilities to align targets among the goals of individual CCLs sharing a given scope.
* Capabilities to identify different types of conflicts between CCLs such as parameters conflict, metrics conflict, or any others.
* Capabilities to address the different interactions between CCLs with adequate mechanisms, such as conflict resolution mechanisms.
* Capabilities to identify before the execution of a proposed action of CCL that such an action could cause undesired effects to other CCLs or to managed entities (e.g., pre-execution and post-execution coordination, concurrency coordination, etc.).
* Capabilities to evaluate the impact and effectiveness of CCLs actions after their execution (e.g., impact assessment).

The coordination of CCLs could be required at different execution points of the CCL translating into different CCL coordination use cases with corresponding CCL coordination services required at those points as illustrated by example Figure 5.6.1.2-1. The coordination of CCLs could be achieved via direct interaction among the CCLs or via a third-party entity, say called the CCLs coordination Function (or simply CCL Coordinator).



**Figure 5.6.1.2-1: Exemplary Closed Control Loop Coordination interaction points.**

Note: the terms at the top indicate general naming of the groupings of coordination interactions at the different execution points during the execution of the CCL. Action-space coordination implies coordinating the sets of actions that the different CCL can apply. Concurrency control implies coordinating the times at which different CCLs can execute actions. Action-impact assessment indicates interactions and processes on the evaluation of the impacts of the different CCLs.

5.6.2 Potential Requirements

REQ-CCL-CONFLICT-1: The MnS Producer for CCL management should support a capability to detect a potential or actual conflict.

Note: A potential conflict is where some events are observed that indicate that there may be a conflict, but the CCL MnS Producer cannot conclude that it is a conflict. So, the CCL can indicate this so that some other entity e.g. the MnS consumer takes responsibility to confirm the conflict.

REQ-CCL-CONFLICT-2: The MnS Producer for CCL management should support a capability to inform an authorized MnS consumer about a potential conflict that has been detected.

REQ-CCL-CONFLICT-3: The MnS Producer for CCL management should support a capability to confirm a detected potential goal, action, indirect target, action execution time, scope conflict.

REQ-CCL-CONFLICT-4: The MnS Producer for CCL management should support a capability to resolve a goal, action, indirect target, action execution time, scope conflict that has been detected.

REQ-CCL-CONFLICT-5: The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to avoid the conflict.

REQ-CCL-CONFLICT-6: The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to resolve the conflict.

5.6.3 Potential Solutions

5.6.3.1 Alternative CCL coordination Approaches

The coordination of CCLs could be accomplished via one of three approaches illustrated by Figure 5.6.3.1-1:

* distributed coordination with distributed execution (Figure 5.6.3.1-1 a), where the CCLs directly coordinate with one another, and each manages execution of its decisions.
* Hierarchical coordination with distributed execution (Figure 5.6.3.1-1 b), where the CCLs coordinate through a separate coordination layer, say via a coordination CCL, but each manages execution of its coordinated decisions.
* Hierarchical coordination and execution (Figure 5.6.3.1-1 c), where the CCLs coordinate through a separate coordination layer, say via a coordination CCL that besides coordination also manages execution of the coordinated decisions.

A diagram of a computer system

Description automatically generated

**Figure** 5.6.3.1**-1: Closed Control Loop Coordination approaches**

Distributed coordination can lead to too many exchanges between the CCLs which may unnecessarily clog the system. On the other hand, “Hierarchical coordination and execution” implies that too much responsibility is concentrated in a single CCL. A desired behavior is that the individual CCLs are responsible for their own decision execution, so it is recommended that to follow the "hierarchical coordination with distributed execution” approach. In this approach, the CCLs are responsible for making their decisions and executing actions but they coordinate with the CCL coordinator before, during or after execution.

The CoordinationCCL supports interactions with different CCLs for the following CCL conflicts

* Goal target Conflicts
* Scope conflicts,
* Direct-action conflicts,
* Indirect target conflict,
* Action execution-time conflict.

For each conflict, the CoordinationCCL supports interactions to

* Detect potential conflicts
* Avoid potential conflicts
* Detect actual conflicts
* resolve actual conflicts

Accordingly, as illustrated by Figure 5.6.3.1-2, an IOC could be introduced for the coordination CCL with child IOCs for the specific capabilities for each conflict type.

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class ManagedEntity <<ProxyClass>>

class CoordinationCCL <<InformationObjectClass>>

class CoordinationProfile <<dataType>>

ManagedEntity "1" \*-- "1" CoordinationCCL: <<names>>

CoordinationCCL "1" -- "\*" CoordinationProfile

note left of ManagedEntity

Represents the following IOCs:

Subnetwork or

ManagedFunction

end note

note right of CoordinationProfile

Represents the following capabilities:

GoalTargetCoordination

ScopeAssignmentsCoordination

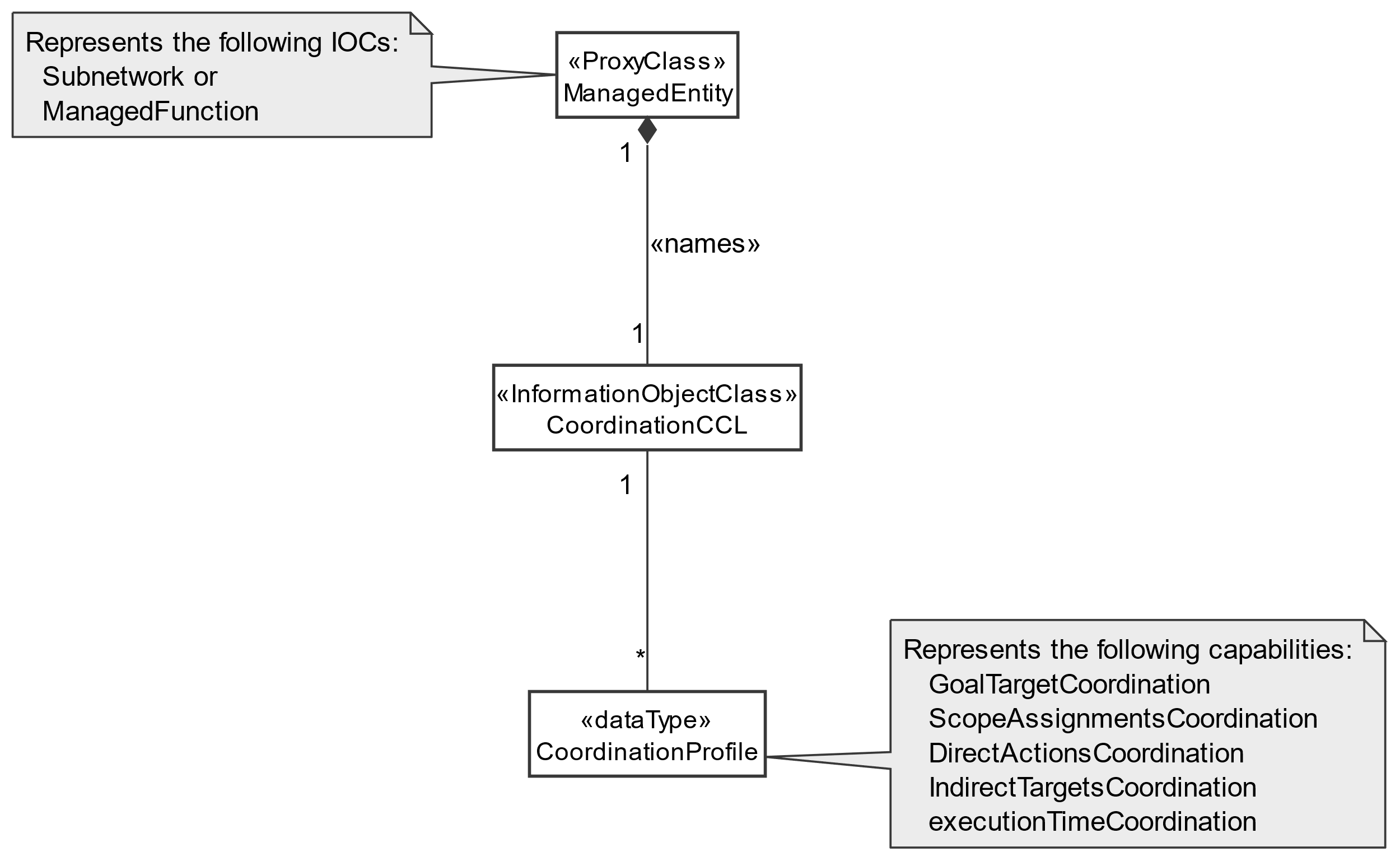
DirectActionsCoordination

IndirectTargetsCoordination

executionTimeCoordination

end note

@enduml



; **Figure 5.6.3.1-2: Closed Control Loop Coordination NRM fragment**

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

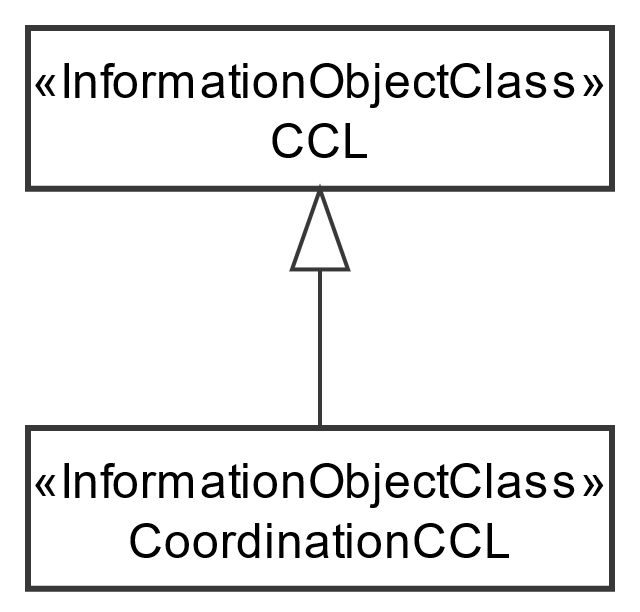
class CCL <<ProxyClass>>

class CCL <<InformationObjectClass>>

class CoordinationCCL <<InformationObjectClass>>

CCL <|-- CoordinationCCL

@enduml



**Figure 5.6.3.1-2: Closed Control Loop Coordination Inheritance Relations**

5.8.4 Evaluation of solutions

5.8.4.1 Alternative CCL coordination Approaches

Distributed coordination can lead to too many exchanges between the CCLs which may unnecessarily clog the system. On the other hand, “Hierarchical coordination and execution” implies that too much responsibility is concentrated in a single CCL. It is better that the individual CCLs are responsible for their own decision execution, so it is recommended that to follow the "hierarchical coordination with distributed execution” approach.

|  |
| --- |
| **Next Change** |