Grading protocol for IRM-SA controlled experiment

Compare each architecture design to the exemplar (see next page), and subtract points from a total of 115 in the following way:

1. Subtract 1 point for every missing knowledge
2. Subtract 2 points for every missing role
3. Subtract 5 points for every missing process
4. Subtract 1 point for every wrong process input or output field
5. Subtract 2 points for every wrong process period
6. Subtract 2 points for every wrong process condition
7. Subtract 8 points for every missing ensemble
8. Subtract 3 points for every wrong ensemble membership condition
9. Subtract 3 points for every wrong ensemble knowledge exchange
10. Subtract 2 points for every wrong ensemble period
11. Subtract 2 points for every wrong ensemble condition

Then normalize the results in the [0-100] scale to obtain the final grades.

Follow these rules to remove any ambiguity in the grading process:

- If a process or an ensemble is entirely missing, subtracting points just from columns D and H, do not penalize the design more
- For the process periods, only penalize for mistakes in the temperature-monitoring processes and the risk-assessment process (because the timing requirements for other processes were not specified explicitly)
- For the roles, only check their existence (and their interpretation in the design), but not the knowledge fields they contained
- Do not penalize (neither reward) extra processes that were meaningful, e.g. monitoring processes (e.g. "check GPS availability", "check acceleration")
- Do not penalize (neither reward) extra ensembles that were meaningful, such as splitting the "data propagation" into two ones, one for propagating the data fast, and one for propagating the data in a slower rate
- Do not penalize (neither reward) extra knowledge fields, e.g. some put fields related to the control logic (e.g. "quickMode", "currentTime").
MonitorAcceleration
Function: getAcceleration()
Input: --
Output: acceleration
Period: 10 sec
Condition: --

MonitorPositionPS
Function: getPositionPS()
Input: --
Output: position
Period: 10 sec
Condition: GM indoors

MonitorPositionGPS
Function: getPositionGPS()
Input: --
Output: position
Period: 10 sec
Condition: GM outdoors

MonitorTemperatureClosely
Function: getTemperature()
Input: --
Output: temperature
Period: 10 sec
Condition: acceleration = 0 in last 10 sec

MonitorTemperatureScarcely
Function: getTemperature()
Input: --
Output: temperature
Period: 60 sec
Condition: acceleration > 0 in last 20 secs

SensorDataAggregator
groupId
positionsMap
temperaturesMap
accelerationsMap

SensorDataProvider
groupId
position
temperature
acceleration

MembersAtRiskProvider
groupId
membersAtRisk

MembersAtRiskListener
groupId
membersAtRisk

GroupLeader
+ groupId
+ positionsMap
+ temperaturesMap
+ accelerationsMap
+ membersAtRisk

AssessGroupMembersCondition
Function: findMembersAtRisk()
Input: positionsMap, temperaturesMap, accelerationsMap
Output: membersAtRisk
Period: 20 sec
Condition: --

PropagationSensorData
Membership:
  member.groupId == coord.groupId
Knowledge exchange:
  coord.positionsMap.put((member.id, member.position))
  coord.temperaturesMap.put((member.id, member.temperature))
  coord.accelerationsMap.put((member.id, member.acceleration))
Period: 10 sec
Condition: --

ExchangeAvailabilityWhenFarFromPOI
Membership:
  member.groupId == coord.groupId
Knowledge exchange:
  member.membersAtRisk = coord.membersAtRisk
Period: 20 sec
Condition: coord.membersAtRisk.size() > 0