Assignment Needs to Objects

Table 1: Needs <-> Objects

Objects	User Needs	Facility Manager Needs	Nonfunctional Needs
Floor	8, 9, 11	6, 7, 8, 9, 10, 11	7, 8
SunDet	11		7, 8
Hallway	17, 18, 19, 20	1, 2, 4	1, 3, 5, 7, 8
Office, CompLab, PRoom, MRoom	1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 14	1, 3, 5	1, 2, 7, 8
HWDoor	18		7, 8
HWOcc	3, 4, 11		4, 7, 8
Desc	2, 3, 4	3, 4	7, 8
RoomOcc	3, 4, 11		4, 7, 8

TaskAssignment

1 Object: Desk

1.1 Task: control desk illuminance

1.1 Strategy:

- return the state of task light on request
- switch light on/off on request from office

1.2 Task: administration of the state of the task light

1.3 Task: computation of the energy consuption

Direction: Office -> Desk

- setTaskLight (Boolean)
- setEnergyDiffTime (Duration)

Direction: Office <- Desk

- newTaskLight (Boolean)
- newEnergy (Joule, Duration)

2 Object: Hallway

2.1 Task: If hallway is occupied, supply enough light (100lx)

2.1 Strategy:

- 2 states: occupied, not occupied from hwOcc
- input: room illuminance near wall to hallway
- control ceiling lights appropiately

2.2 Task: U19

2.2 Strategy:

- since the wall switches directly control the physical light, the control systems only deactivates the occupancy control for 5 time units after a wall switch has been pushed

2.3 Task: FM4

2.3 Strategy: panel

2.4 Task: Remote control

2.4 Strategy: external settings provided by the floor panel must be realized immediately

2.5 Task: Energy Consumption

2.5 Strategy: compute the energy consumption depending on the duration the light is on and the times the light is switched on and off

2.6 Task: Behavior in Case of Malfunction

2.6 Strategy: transmit a reported malfunction from the object HwOcc to the object floor

Direction: Hallway -> HwOcc

- getHwOcc

Direction: Hallway -> Rooms

- getIlluminance

Direction: Hallway -> HwLight

- setHwLight2

Direction: Hallway -> Floor

- newHwEnergyConsumption (Joule, Duration), Duration to report
- newHwMalFunction (Errortype)

Direction: Hallway <- Floor

- setHwEnergyDiffTime (Duration)
- setHwLight (Boolean)
- setHwPbDelay (Duration)

Direction: Hallway <- HwOcc

- newHwOcc (Boolean)
- newOccMal (ErrorType)

Direction: Hallway <- Rooms

- newIlluminance (RoomIdType, Lux)

Direction: Hallway <- HwLight

- newSll (Binary)

Direction: Hallway -> Floor

- getHwPBDelay

3 Object: HWOcc

3.1 Task: Determine occupancy by motion detection

3.1 Strategy:

- if imd3 sends a true, set occupancy to true
- if one of the inside imds turns to true, set the occupancy to true.

3.2 Task: Determine occupancy by hallway door (HWDoor)

3.2 Strategy:

- if HWDoor reports the entrance of a person into the hallway set occupancy to true

3.3 Task: determine unoccupation by motion detectors

- if both of the inside motion detectors and the imd3 detect no motion for t sec, set occupancy to false.

3.4 Task: Determine occupancy by room doors

3.4 Strategy:

- if room door turns to open, set the occupancy to true.

3.5 Task: Malfunction of one of the sensors

3.5 Strategy: if there exists a malfunction, set the occupancy to true, send a message to the FM.

- if a Hallwaydoor sends an entrance, and the imd3 does not sens an motion, there is a malfunction.
- if all doors are closed, and the state changes to unoccupied without a change of at least one door, there is a malfunction after a delaytime of 1 hour.

3.6 Task: Return status by request

3.6 Strategy: store the actual status and return it by request

Direction: HwOcc -> Hallway

- newHwOcc (Boolean)
- newOccMal (ErrorType)

Direction: HwOcc <- MotDet

- newMotDet (Binary)
- newMotDetMal (ErrorType)

Direction: HwOcc <- Door

- newEntrance

Direction: HwOcc <- HwDoor

- newEntrance

Direction: HwOcc <- Hallway

- getHwOcc

Direction: HwOcc <- Floor

- setHWOccDelay (Duration)

Direction: HwOcc -> Floor

- getHwOccDelay

4 Object: RoomOcc

4.1 Task: Determine occupancy by motion detection

4.1 Strategy:

- if imd sends a "1", set occupancy to true

4.2 Task: determine unoccupation by motion detectors

4.2 Strategy:

- the room must be occupied and a newDoorsOpen (i.e., all doors were closed, at least one door is opened) must be received from a room, before the state can switch to false.
- if the imd detect no motion for 60 sec, set occupancy to false, in regard to the previous condition.

4.3 Task: Propagate room state on request or spontaneous change

4.4 Task: Determine occupancy by room doors

4.4 Strategy:

- if the room is unoccupied and all doors are closed, at least one door has to be opened before the state can switch to true.

4.5 Task: Malfunction of the imd

4.5 Strategy:

- if the imd does not work correctly, set the occupancy to true.

Direction: Rooms -> RoomOcc

- newDoorsOpen

Direction: RoomOcc -> Rooms

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: RoomOcc <- MotDet

- newMotDet (Binary)
- newMotDetMal (ErrorType)

Direction: RoomOcc -> Door

- getDoor

Direction: Door -> RoomOcc

- newDoor (Boolean)

5 Object: SunDet

5.1 Task: determine illuminance in respect of the given room

5.1 Strategy:

- determine the correct outdoor light sensor(s) and return its value(s), according table 2.
- if the illuminance value(s) of column "light sensor" is not accessible use the light sensors as given in column "aux. light sensor" otherwise return an error.

5.2 Task: detect malfunctions of individual light sensors

5.2 Strategy:

- perform an inconsistency check every hour by comparing the values of light sensors on the same side of the facade
- perform plausibility checks by considering the current day time
- if the value of a sensor does not change during one hour it is assumed to be damaged
- check whether the returned value of the light sensor is in the interval [1..100000]

5.3 Task: Report malfunctions

5.3 Strategy:

- if any malfunction occurs an error signal has to be sent to the Floor

Table 2 Assignment room number <->relevant light sensors

Roomnumber	light sensor	aux. light sensor
435, 433, 431, 429	ols1	ols5
427	(3*ols1+5*ols5) / 8	ols1 oder ols5
425, 423, 411, 413	ols5	(ols1+ols3) / 2
421	(5*ols3+1*ols5) / 6	ols3 oder ols5
419, 417, 415	ols3	ols5
426, 424	ols2	ols6
422	(ols2 + 3*ols6)/4	ols2 oder ols6
420	(ols4 + 3* ols6)/4	ols4 oder ols6
418, 416, 414	ols4	ols6
412	(3*ols4 + 3*ols6)/6	ols4 oder ols6
410	ols6	(ols2+ols4) / 2

Direction: Rooms -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> Rooms

- newSun (RoomIdType, Lux)

Direction: SunDet -> OutDoorLight

- getOls()

Direction: OutDoorLight -> SunDet

- newOls(Lux)

Direction: SunDet -> Floor

- newSunDetMal (ErrorType)

6 Object: Light

6.1 Task: Switch the lamp on or off

6.1 Strategy:

- If a pulse arrives the status of the lamp has to be changed from “on” to “off” respectively from “off” to “on”.

6.2 Task: Return lamp status by request

6.2 Strategy:

- Return the status of the lamp (“on” or “off”) on the sll by request.

6.3 Task: Set the dimmer to the given value

6.3 Strategy:

- Set the dimmer to the requested value [in percent].

6.4 Task: Monitor the control system

6.4 Strategy:

- If not every t_5 a control pulse on the CIA-input is detected, set the dimm value to 100% and switch on the light.

Direction: Rooms -> Light

- setLight (Boolean)
- setDimLight(Percent)

- setLight (Boolean)
- setCIA
- getLight

Direction: Light -> Rooms

- newLight (Boolean)

Direction: Light -> Switch

- getSwitch

Direction: Switch ->Light

- newSwitch(Binary)

Direction: Light -> Dimmer

- setDV(Percent)

Direction: Light -> PulsGen

- setPG(Binary)
-

7 Object: HWLight

7.1 Task: Switch the lamp on or off

7.1 Strategy:

- If a pulse arrives the status of the lamp has to be changed from “on” to “off” respectively from “off” to “on”.

7.2 Task: Propagate lamp status

7.2 Strategy:

- Propagate the status of the lamp (“on” or “off”) if it changes.

7.3 Task: Compute Energy Consumption

7.3 Strategy: compute the energy consumption depending on the duration the light is on and the times the light is switched on and off

Direction: HwLight -> Hallway

- newSll (Binary)

Direction: HwLight <- Hallway

- setHwLight2

Direction: Hallway -> HwLight

- setEnergyDiffTime(Duration)

8 Object: Office

8.1 Task: control ceiling lights with minimal energy consumption (500lx)

8.1 Strategy:

- A light scene contains the desired room/desk light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups and the task light. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

8.2 Task: control and distribution of light scenes

8.2 Strategy: provide a control panel to set the task light, the ceiling lights and the actual ambient light level (light scene)

8.3 Task: compute illuminance near wall to hallway

8.3 Strategy: inputs: outdoor illuminance, current light level of the task and ceiling lights, state of door to hallway

8.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

8.4 Strategy: use the panel of the Floor

8.5 Task: inform the user and the floor in case of malfunctions

8.5 Strategy: use the room monitor

8.6 Task: inform the lights about the state of the control system

8.6 Strategy: as long as the control system is active it sends periodically a signal to the ceiling lights

8.7 Task: Propagate door changes

8.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

8.8 Task: Compute the enrgy consumption

Direction: Office -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> Office

- newLight (Boolean)

- newEnergy(Joule)

Direction: Office -> Desk

- setTaskLight (Boolean)

Direction: Desk -> Office

- newTaskLight (Boolean)

Direction: Office -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> Office

- newSun (Lux)

Direction: RoomOcc -> Office

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: Office -> RoomOcc

- newDoorsOpen

Direction: Office -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> Office

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> Office

- getIlluminance

Direction: Office -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: Office -> HwOcc

- newEntrance

Direction: Door -> Office

- newDoor (Boolean)

9 Object: CompLab

9.1 Task: control ceiling lights with minimal energy consumption (500lx)

9.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

9.2 Task: control and distribution of light scenes

9.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

9.3 Task: compute illuminace near wall to hallway

9.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

9.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

9.4 Strategy: use the panel of the Floor

9.5 Task: inform the user and the floor in case of malfunctions

9.5 Strategy: use the room monitor

9.6 Task: inform the lights about the state of the control system

9.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

9.7 Task: Propagate door changes

9.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is

9.8 Task: Compute the enrgy consumption

Direction: CompLab -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> CompLab

- newLight (Boolean)
- newEnergy(Joule)

Direction: CompLab -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> CompLab

- newSun (Lux)

Direction: RoomOcc -> CompLab

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: CompLab -> RoomOcc

- newDoorsOpen

Direction: CompLab -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> CompLab

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> CompLab

- getIlluminance

Direction: CompLab -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: CompLab -> HwOcc

- newEntrance

Direction: Door -> CompLab

- newDoor (Boolean)

10 Object: PRoom

10.1 Task: control ceiling lights with minimal energy consumption (500lx)

10.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

10.2 Task: control and distribution of light scenes

10.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

10.3 Task: compute illuminace near wall to hallway

10.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

10.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

10.4 Strategy: use the panel of the Floor

10.5 Task: inform the user and the floor in case of malfunctions

10.5 Strategy: use the room monitor

10.6 Task: inform the lights about the state of the control system

10.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

10.7 Task: Propagate door changes

10.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

10.8 Task: Compute the enrgy consumption

Direction: PRoom -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> PRoom

- newLight (Boolean)
- newEnergy(Joule)

Direction: PRoom -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> PRoom

- newSun (Lux)

Direction: RoomOcc -> PRoom

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: PRoom -> RoomOcc

- newDoorsOpen

Direction: PRoom -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> PRoom

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> PRoom

- getIlluminance

Direction: PRoom -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: PRoom -> HwOcc

- newEntrance

Direction: Door -> PRoom

- newDoor (Boolean)

11 Object: MRoom

11.1 Task: control ceiling lights with minimal energy consumption (500lx)

11.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.

- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

11.2 Task: control and distribution of light scenes

11.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

11.3 Task: compute illuminace near wall to hallway

11.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

11.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

11.4 Strategy: use the panel of the Floor

11.5 Task: inform the user and the floor in case of malfunctions

11.5 Strategy: use the room monitor

11.6 Task: inform the lights about the state of the control system

11.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

11.7 Task: Propagate door changes

11.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

11.8 Task: Compute the enrgy consumption

Direction: MRoom -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> MRoom

- newLight (Boolean)
- setEnergy(Joule)

Direction: MRoom -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> MRoom

- newSun (Lux)

Direction: RoomOcc -> MRoom

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: MRoom -> RoomOcc

- newDoorsOpen

Direction: MRoom -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> MRoom

- newSunDetMal (RoomIdType, ErrorType)

- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> MRoom

- getIlluminance

Direction: MRoom -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: MRoom -> HwOcc

- newEntrance

Direction: Door -> MRoom

- newDoor (Boolean)

12 Object: Door

12.1 Task: provide the state of the door closed contact (dcc)

12.1 Strategy: propagate and convert type of the state

Direction: RoomOcc -> Door

- getDoor

Direction: Door -> Rooms

- newDoor (Boolean)

Direction: Door -> Switch

- getSwitch

Direction: Switch -> Door

- newSwitch (Binary), true iff door is closed

13 Object: HWDoor

13.1 Task: detect and propagate occupancy by entrance from adjacent hallways

13.1 Strategy: (this strategy is valid for both ends of the hallway)

- inputs: door closed contact, imds
- output: hallwayEntrance
- if the outside imd turns to true, and the door is closed, switch hallwayEntrance to true, iff the door opens.
- if the outside imd turns to true, and the door is opened, switch hallwayEntrance to true, iff the inside imd turns to true.

Direction: HwDoor -> Switch

- getSwitch

Direction: HwDoor -> HwOcc

- newEntrance

Direction: HwDoor <- MotDet

- newMotDet (Binary)

Direction: HwDoor <- Switch

- newSwitch (Binary) = 1 iff door is closed

14 Object: Section

14.1 Task: no task found

15 Object: Floor

15.1 Task: Provide a possibility to set several values: default light scene, default ambient light level, values T1, T3 for each room

15.1 Strategy: use panel and monitor

15.2 Task: all lights can be controlled with the panel

15.2 Strategy:

- all lights in all rooms and hallway sections can be switched on or off

15.3 Task: administration of malfunctions and energy consumption reports

15.3 Strategy:

- energy consumption is received every T5 minutes from all rooms and all hallway sections
- propagate the value of T5 to all rooms and all hallway sections by broadcasting
- malfunctions are received immediately and stored
- on request malfunctions or energy consumption are reported

15.4 Task: display malfunctions and energy consumption on request

15.4 Strategy: use the monitor

Direction: Floor -> Rooms

- setDefaultLightScene(RoomIdType, LightSceneType)
- setRoomDelay (RoomIdType, Duration)
- setReoccRoomDelay (RoomIdType, Duration)
- setRoomEnergyDiffTime(Duration)
- newSunDetMal(RoomIdType, ErrorType)

Direction: Rooms -> Floor

- newRoomMalfunction(RoomIdType, ErrorType)
- newRoomEnergyConsumption(RoomIdType, Joule, Duration)

Direction: Floor -> Hallway

- newHwPbDelay(Duration)
- setHwLight(Boolean)
- setHwEnergyDiffTime(Duration)

Direction: Hallway -> Floor

- getHwPbDelay

- newHwEnergyConsumption(Joule, Duration)
- newHwMalfunction(ErrorType)

Direction: HwOcc <- Floor

- setHWOccDelay (Duration)

Direction: HwOcc -> Floor

- getHWOccDelay

Direction: SunDet -> Floor

- newSunDetMal (ErrorType)

Assignment Needs to Objects

Table 1: Needs <-> Objects

Objects	User Needs	Facility Manager Needs	Nonfunctional Needs
Floor	8, 9, 11	6, 7, 8, 9, 10, 11	7, 8
SunDet	11		7, 8
Hallway	17, 18, 19, 20	1, 2, 4	1, 3, 5, 7, 8
Office, CompLab, PRoom, MRoom	1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 14	1, 3, 5	1, 2, 7, 8
HWDoor	18		7, 8
HWOcc	3, 4, 11		4, 7, 8
Desc	2, 3, 4	3, 4	7, 8
RoomOcc	3, 4, 11		4, 7, 8

TaskAssignment

1 Object: Desk

1.1 Task: control desk illuminance

1.1 Strategy:

- return the state of task light on request
- switch light on/off on request from office

1.2 Task: administration of the state of the task light

1.3 Task: computation of the energy consuption

Direction: Office -> Desk

- setTaskLight (Boolean)
- setEnergyDiffTime (Duration)

Direction: Office <- Desk

- newTaskLight (Boolean)
- newEnergy (Joule, Duration)

2 Object: Hallway

2.1 Task: If hallway is occupied, supply enough light (100lx)

2.1 Strategy:

- 2 states: occupied, not occupied from hwOcc
- input: room illuminance near wall to hallway
- control ceiling lights appropiately

2.2 Task: U19

2.2 Strategy:

- since the wall switches directly control the physical light, the control systems only deactivates the occupancy control for 5 time units after a wall switch has been pushed

2.3 Task: FM4

2.3 Strategy: panel

2.4 Task: Remote control

2.4 Strategy: external settings provided by the floor panel must be realized immediately

2.5 Task: Energy Consumption

2.5 Strategy: compute the energy consumption depending on the duration the light is on and the times the light is switched on and off

2.6 Task: Behavior in Case of Malfunction

2.6 Strategy: transmit a reported malfunction from the object HwOcc to the object floor

Direction: Hallway -> HwOcc

- getHwOcc

Direction: Hallway -> Rooms

- getIlluminance

Direction: Hallway -> HwLight

- setHwLight2

Direction: Hallway -> Floor

- newHwEnergyConsumption (Joule, Duration), Duration to report
- newHwMalFunction (Errortype)

Direction: Hallway <- Floor

- setHwEnergyDiffTime (Duration)
- setHwLight (Boolean)
- setHwPbDelay (Duration)

Direction: Hallway <- HwOcc

- newHwOcc (Boolean)
- newOccMal (ErrorType)

Direction: Hallway <- Rooms

- newIlluminance (RoomIdType, Lux)

Direction: Hallway <- HwLight

- newSll (Binary)

Direction: Hallway -> Floor

- getHwPBDelay

3 Object: HWOcc

3.1 Task: Determine occupancy by motion detection

3.1 Strategy:

- if imd3 sends a true, set occupancy to true
- if one of the inside imds turns to true, set the occupancy to true.

3.2 Task: Determine occupancy by hallway door (HWDoor)

3.2 Strategy:

- if HWDoor reports the entrance of a person into the hallway set occupancy to true

3.3 Task: determine unoccupation by motion detectors

- if both of the inside motion detectors and the imd3 detect no motion for t sec, set occupancy to false.

3.4 Task: Determine occupancy by room doors

3.4 Strategy:

- if room door turns to open, set the occupancy to true.

3.5 Task: Malfunction of one of the sensors

3.5 Strategy: if there exists a malfunction, set the occupancy to true, send a message to the FM.

- if a Hallwaydoor sends an entrance, and the imd3 does not sens an motion, there is a malfunction.
- if all doors are closed, and the state changes to unoccupied without a change of at least one door, there is a malfunction after a delaytime of 1 hour.

3.6 Task: Return status by request

3.6 Strategy: store the actual status and return it by request

Direction: HwOcc -> Hallway

- newHwOcc (Boolean)
- newOccMal (ErrorType)

Direction: HwOcc <- MotDet

- newMotDet (Binary)
- newMotDetMal (ErrorType)

Direction: HwOcc <- Door

- newEntrance

Direction: HwOcc <- HwDoor

- newEntrance

Direction: HwOcc <- Hallway

- getHwOcc

Direction: HwOcc <- Floor

- setHWOccDelay (Duration)

Direction: HwOcc -> Floor

- getHwOccDelay

4 Object: RoomOcc

4.1 Task: Determine occupancy by motion detection

4.1 Strategy:

- if imd sends a "1", set occupancy to true

4.2 Task: determine unoccupation by motion detectors

4.2 Strategy:

- the room must be occupied and a newDoorsOpen (i.e., all doors were closed, at least one door is opened) must be received from a room, before the state can switch to false.
- if the imd detect no motion for 60 sec, set occupancy to false, in regard to the previous condition.

4.3 Task: Propagate room state on request or spontaneous change

4.4 Task: Determine occupancy by room doors

4.4 Strategy:

- if the room is unoccupied and all doors are closed, at least one door has to be opened before the state can switch to true.

4.5 Task: Malfunction of the imd

4.5 Strategy:

- if the imd does not work correctly, set the occupancy to true.

Direction: Rooms -> RoomOcc

- newDoorsOpen

Direction: RoomOcc -> Rooms

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: RoomOcc <- MotDet

- newMotDet (Binary)
- newMotDetMal (ErrorType)

Direction: RoomOcc -> Door

- getDoor

Direction: Door -> RoomOcc

- newDoor (Boolean)

5 Object: SunDet

5.1 Task: determine illuminance in respect of the given room

5.1 Strategy:

- determine the correct outdoor light sensor(s) and return its value(s), according table 2.
- if the illuminance value(s) of column “light sensor” is not accessible use the light sensors as given in column “aux. light sensor” otherwise return an error.

5.2 Task: detect malfunctions of individual light sensors

5.2 Strategy:

- perform an inconsistency check every hour by comparing the values of light sensors on the same side of the facade
- perform plausibility checks by considering the current day time
- if the value of a sensor does not change during one hour it is assumed to be damaged
- check whether the returned value of the light sensor is in the interval [1..100000]

5.3 Task: Report malfunctions

5.3 Strategy:

- if any malfunction occurs an error signal has to be sent to the Floor

Table 2 Assignment room number <->relevant light sensors

Roomnumber	light sensor	aux. light sensor
435, 433, 431, 429	ols1	ols5
427	(3*ols1+5*ols5) / 8	ols1 oder ols5
425, 423, 411, 413	ols5	(ols1+ols3) / 2
421	(5*ols3+1*ols5) / 6	ols3 oder ols5
419, 417, 415	ols3	ols5
426, 424	ols2	ols6
422	(ols2 + 3*ols6)/4	ols2 oder ols6
420	(ols4 + 3* ols6)/4	ols4 oder ols6
418, 416, 414	ols4	ols6
412	(3*ols4 + 3*ols6)/6	ols4 oder ols6
410	ols6	(ols2+ols4) / 2

Direction: Rooms -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> Rooms

- newSun (RoomIdType, Lux)

Direction: SunDet -> OutDoorLight

- getOls()

Direction: OutDoorLight -> SunDet

- newOls(Lux)

Direction: SunDet -> Floor

- newSunDetMal (ErrorType)

6 Object: Light

6.1 Task: Switch the lamp on or off

6.1 Strategy:

- If a pulse arrives the status of the lamp has to be changed from “on” to “off” respectively from “off” to “on”.

6.2 Task: Return lamp status by request

6.2 Strategy:

- Return the status of the lamp (“on” or “off”) on the sll by request.

6.3 Task: Set the dimmer to the given value

6.3 Strategy:

- Set the dimmer to the requested value [in percent].

6.4 Task: Monitor the control system

6.4 Strategy:

- If not every t_5 a control pulse on the CIA-input is detected, set the dimm value to 100% and switch on the light.

Direction: Rooms -> Light

- setLight (Boolean)
- setDimLight(Percent)

- setLight (Boolean)
- setCIA
- getLight

Direction: Light -> Rooms

- newLight (Boolean)

Direction: Light -> Switch

- getSwitch

Direction: Switch ->Light

- newSwitch(Binary)

Direction: Light -> Dimmer

- setDV(Percent)

Direction: Light -> PulsGen

- setPG(Binary)
-

7 Object: HWLight

7.1 Task: Switch the lamp on or off

7.1 Strategy:

- If a pulse arrives the status of the lamp has to be changed from “on” to “off” respectively from “off” to “on”.

7.2 Task: Propagate lamp status

7.2 Strategy:

- Propagate the status of the lamp (“on” or “off”) if it changes.

7.3 Task: Compute Energy Consumption

7.3 Strategy: compute the energy consumption depending on the duration the light is on and the times the light is switched on and off

Direction: HwLight -> Hallway

- newSll (Binary)

Direction: HwLight <- Hallway

- setHwLight2

Direction: Hallway -> HwLight

- setEnergyDiffTime(Duration)

8 Object: Office

8.1 Task: control ceiling lights with minimal energy consumption (500lx)

8.1 Strategy:

- A light scene contains the desired room/desk light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups and the task light. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

8.2 Task: control and distribution of light scenes

8.2 Strategy: provide a control panel to set the task light, the ceiling lights and the actual ambient light level (light scene)

8.3 Task: compute illuminance near wall to hallway

8.3 Strategy: inputs: outdoor illuminance, current light level of the task and ceiling lights, state of door to hallway

8.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

8.4 Strategy: use the panel of the Floor

8.5 Task: inform the user and the floor in case of malfunctions

8.5 Strategy: use the room monitor

8.6 Task: inform the lights about the state of the control system

8.6 Strategy: as long as the control system is active it sends periodically a signal to the ceiling lights

8.7 Task: Propagate door changes

8.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

8.8 Task: Compute the enrgy consumption

Direction: Office -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> Office

- newLight (Boolean)

- newEnergy(Joule)

Direction: Office -> Desk

- setTaskLight (Boolean)

Direction: Desk -> Office

- newTaskLight (Boolean)

Direction: Office -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> Office

- newSun (Lux)

Direction: RoomOcc -> Office

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: Office -> RoomOcc

- newDoorsOpen

Direction: Office -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> Office

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> Office

- getIlluminance

Direction: Office -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: Office -> HwOcc

- newEntrance

Direction: Door -> Office

- newDoor (Boolean)

9 Object: CompLab

9.1 Task: control ceiling lights with minimal energy consumption (500lx)

9.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

9.2 Task: control and distribution of light scenes

9.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

9.3 Task: compute illuminace near wall to hallway

9.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

9.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

9.4 Strategy: use the panel of the Floor

9.5 Task: inform the user and the floor in case of malfunctions

9.5 Strategy: use the room monitor

9.6 Task: inform the lights about the state of the control system

9.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

9.7 Task: Propagate door changes

9.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is

9.8 Task: Compute the enrgy consumption

Direction: CompLab -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> CompLab

- newLight (Boolean)
- newEnergy(Joule)

Direction: CompLab -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> CompLab

- newSun (Lux)

Direction: RoomOcc -> CompLab

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: CompLab -> RoomOcc

- newDoorsOpen

Direction: CompLab -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> CompLab

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> CompLab

- getIlluminance

Direction: CompLab -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: CompLab -> HwOcc

- newEntrance

Direction: Door -> CompLab

- newDoor (Boolean)

10 Object: PRoom

10.1 Task: control ceiling lights with minimal energy consumption (500lx)

10.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.
- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

10.2 Task: control and distribution of light scenes

10.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

10.3 Task: compute illuminace near wall to hallway

10.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

10.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

10.4 Strategy: use the panel of the Floor

10.5 Task: inform the user and the floor in case of malfunctions

10.5 Strategy: use the room monitor

10.6 Task: inform the lights about the state of the control system

10.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

10.7 Task: Propagate door changes

10.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

10.8 Task: Compute the enrgy consumption

Direction: PRoom -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> PRoom

- newLight (Boolean)
- newEnergy(Joule)

Direction: PRoom -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> PRoom

- newSun (Lux)

Direction: RoomOcc -> PRoom

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: PRoom -> RoomOcc

- newDoorsOpen

Direction: PRoom -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> PRoom

- newSunDetMal (RoomIdType, ErrorType)
- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> PRoom

- getIlluminance

Direction: PRoom -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: PRoom -> HwOcc

- newEntrance

Direction: Door -> PRoom

- newDoor (Boolean)

11 Object: MRoom

11.1 Task: control ceiling lights with minimal energy consumption (500lx)

11.1 Strategy:

- A light scene contains the desired room light levels and a priority list defining the order of lights (L1, L2, TaskLight) to achieve these light levels.
- if the room is reoccupied within T1 minutes, reestablish the last state of the lights.

- if the room is occupied now and was unoccupied for more than T1 minutes, then the default light scene has to be reestablished
- compute illuminance w.r.t. a specific point in the room depending on the chosen light scene
- compute periodically depending on the computed illuminance, the value requested from the sun light detector, and the priority given by the chosen light scene the settings of the ceiling light groups. The settings are given in percent.
- propagate the computed settings
- if the room is unoccupied for more than T3 minutes and the light is on, the light must be switched off.
- external settings provided by the floor panel must be realized immediately
- assume darkness in case of SunDet malfunction

11.2 Task: control and distribution of light scenes

11.2 Strategy: provide a control panel to set the ceiling lights and the actual ambient light level (light scene)

11.3 Task: compute illuminace near wall to hallway

11.3 Strategy: inputs: outdoor illuminance, current light level of the ceiling lights, state of door to hallway

11.4 Task: provide a possibility to set values T1, T3, the default light scene, and the default ambient light level

11.4 Strategy: use the panel of the Floor

11.5 Task: inform the user and the floor in case of malfunctions

11.5 Strategy: use the room monitor

11.6 Task: inform the lights about the state of the control system

11.6 Strategy: as long as the control system is active it sends periodically a signal to the lights

11.7 Task: Propagate door changes

11.7 Strategy: send a newDoorsOpen to RoomOcc, iff all doors were closed and at least one door is opened

11.8 Task: Compute the enrgy consumption

Direction: MRoom -> Light

- setDimLight (Percent)
- setLight (Boolean)
- setCIA
- getLight
- setEnergyDiffTime (Duration)

Direction: Light -> MRoom

- newLight (Boolean)
- setEnergy(Joule)

Direction: MRoom -> SunDet

- getSun(RoomIdType)

Direction: SunDet -> MRoom

- newSun (Lux)

Direction: RoomOcc -> MRoom

- newOcc (Boolean)
- newOccMal (ErrorType)

Direction: MRoom -> RoomOcc

- newDoorsOpen

Direction: MRoom -> Floor

- newRoomMalfunction (RoomIdType, ErrorType)
- newRoomEnergyConsumption (RoomIdType, Joule, Duration)

Direction: Floor -> MRoom

- newSunDetMal (RoomIdType, ErrorType)

- setRoomDelay (RoomIdType, Duration)
- setRoomReoccDelay (RoomIdType, Duration)
- setDefaultLightScene (RoomIdType, LightSceneType)
- setDefaultAmbientLightLevel (RoomIdType, LightLevel)
- setEnergyDiffTime (Duration)

Direction: Hallway -> MRoom

- getIlluminance

Direction: MRoom -> Hallway

- newIlluminance (RoomIdType, Lux)

Direction: MRoom -> HwOcc

- newEntrance

Direction: Door -> MRoom

- newDoor (Boolean)

12 Object: Door

12.1 Task: provide the state of the door closed contact (dcc)

12.1 Strategy: propagate and convert type of the state

Direction: RoomOcc -> Door

- getDoor

Direction: Door -> Rooms

- newDoor (Boolean)

Direction: Door -> Switch

- getSwitch

Direction: Switch -> Door

- newSwitch (Binary), true iff door is closed

13 Object: HWDoor

13.1 Task: detect and propagate occupancy by entrance from adjacent hallways

13.1 Strategy: (this strategy is valid for both ends of the hallway)

- inputs: door closed contact, imds
- output: hallwayEntrance
- if the outside imd turns to true, and the door is closed, switch hallwayEntrance to true, iff the door opens.
- if the outside imd turns to true, and the door is opened, switch hallwayEntrance to true, iff the inside imd turns to true.

Direction: HwDoor -> Switch

- getSwitch

Direction: HwDoor -> HwOcc

- newEntrance

Direction: HwDoor <- MotDet

- newMotDet (Binary)

Direction: HwDoor <- Switch

- newSwitch (Binary) = 1 iff door is closed

14 Object: Section

14.1 Task: no task found

15 Object: Floor

15.1 Task: Provide a possibility to set several values: default light scene, default ambient light level, values T1, T3 for each room

15.1 Strategy: use panel and monitor

15.2 Task: all lights can be controlled with the panel

15.2 Strategy:

- all lights in all rooms and hallway sections can be switched on or off

15.3 Task: administration of malfunctions and energy consumption reports

15.3 Strategy:

- energy consumption is received every T5 minutes from all rooms and all hallway sections
- propagate the value of T5 to all rooms and all hallway sections by broadcasting
- malfunctions are received immediately and stored
- on request malfunctions or energy consumption are reported

15.4 Task: display malfunctions and energy consumption on request

15.4 Strategy: use the monitor

Direction: Floor -> Rooms

- setDefaultLightScene(RoomIdType, LightSceneType)
- setRoomDelay (RoomIdType, Duration)
- setReoccRoomDelay (RoomIdType, Duration)
- setRoomEnergyDiffTime(Duration)
- newSunDetMal(RoomIdType, ErrorType)

Direction: Rooms -> Floor

- newRoomMalfunction(RoomIdType, ErrorType)
- newRoomEnergyConsumption(RoomIdType, Joule, Duration)

Direction: Floor -> Hallway

- newHwPbDelay(Duration)
- setHwLight(Boolean)
- setHwEnergyDiffTime(Duration)

Direction: Hallway -> Floor

- getHwPbDelay

- newHwEnergyConsumption(Joule, Duration)
- newHwMalfunction(ErrorType)

Direction: HwOcc <- Floor

- setHWOccDelay (Duration)

Direction: HwOcc -> Floor

- getHWOccDelay

Direction: SunDet -> Floor

- newSunDetMal (ErrorType)