Contents

Composed Problem Specification 1.0

Iteration Group Formalization: 2

Iteration Decomposition/Composition: 1

Project: Light_Heating_32_4

Authors: Kronenburg, Peper

Status: During Composition

Date: November 23, 1998

Referenced Document: BD : team3.2-BuildingDescription

Referenced Document: PD : team3.2-ProblemDescription

Referenced Document: DIC : team3.2-Dictionary

Comment: Einleitung überarbeiten!

Introduction:

This document contains the formalization of several needs for a new light and heating control system for the fourth floor of building 32 of the University of Kaiserslautern. The needs are given in the document Problemdescription. Needs which could up to now not be formalized are presented in their natural language version. A more informal description of the environment in which the control system should be embedded is given in the document team3_inst_arch.v[n].fm. A more informal explanation of the terms and entities used in this requirements are given in the document dictionary. The formal definitions of the terms used in this document are given in the document domain knowledge.

The main motivation for the development of a new light control system are the disadvantages of the currently existing system. Since all lights are controlled manually, electrical energy is wasted by lighting rooms which are not occupied and by little possibilities to adjust light sources relative to need and daylight.

Additionally this document contains a formalization of the environment.

Comment: Die folgenden Fragen müssen noch an den entsprechenden Stellen als Customer Requests einsortiert werden.

CustomerReq: Sollen die Peripherieraume gesteuert werden oder nicht?

CustomerReq: Ist es richtig, daß das Task Light nicht durch die Maschine gesteuert werden kann?

CustomerReq: Wie sieht das Verhalten der Pushbuttons aus? Was passiert, wenn man auf eine Pushbutton drückt: Licht aus/an? Signal an die Maschine? oder beides?

CustomerReq: Wozu sind in der BD "wall switches" bei den Sensoren angegeben, sie werden nirgendwo verwendet. (allgemein sollten wir mal überprüfen, ob alle Sachen, die in der PD+BD vorkommen, benötigt werden, d.h. bei uns referenziert werden; dafür müßte man die PD+BD in eindeutige Einheiten zerlegen, die man auch referenzieren kann.)

CustomerReq: Existiert das ControlPanel bereits (siehe BD) oder nicht (siehe PD)?

CustomerReq: Was ist mit "upper sash"; wird nicht bei den Aktuatoren erwähnt.

CustomerReq: Wie sehen bei den Aktuatoren die "bekannten Positionen" aus und woher weiß man, daß man in einer solchen ist? Gibt es da noch Sensoren, die bisher nicht erwähnt wurden? (Siehe z.B. "completely closed" bei BD/Actuators.)

CustomerReq: Was ist von den Bemerkungen in der BD zu halten, die angeben, daß Sensoren (ceiling light sensor) oder Aktuatoren (dimmer) nicht verwendet oder verfügbar sind?

CustomerReq: Welche Fehlfunktionen können manuell eingegeben werden und wie hat sich das System dann zu verhalten?

Comment: Inhaltsverzeichnis und wenn es geht auch einen Index erzeugen lassen.

Global Signature

Domain $TEMP = \{-50 \ldots +70\}$

NL	: temperature values
Intention	: The set of possible temperature values for selections, etc. in degrees celsius
References	: PD/UH1, PD/FMH3
DeveloperReq	: Is there another domain for temp values in the real world?
CustomerReq	: Is the definition o.k.? Resolution?

Domain $BINARY = \{0, 1\} < NAT$

Intention	: Domain modelling the binary range 0,1.
References	: BD/Sensors
Comment	: Bessere Intention?
DeveloperReq	: Brauchen wir diese Domäne noch?

Domain $PERCENT = \{0 \ldots 100\} < NAT$

Intention	: Domain modelling percent.
References	: BD/Sensors
Comment	: Bessere Intention?

Domain $DT1\_RANGE < TIME$

NL	: range of DT1
Intention	: set of allowed values for the time DT1
References	: PD/Par7
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $DT2\_RANGE < TIME$

NL	: range of DT2
Intention	: set of allowed values for the time DT2
References	: PD/Par7
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $DT3\_RANGE < TIME$

NL	: range of DT3
Intention	: set of allowed values for the time DT3
References	: PD/Par7
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $COMFORT\_TEMP < TEMP$

NL	: range of comfort temperatures
Intention	: set of allowed comfort temperatures
References	: PD/UH1, PD/FMH3
Comment	: Dieser Wert kann unterschiedlich für jeden Typ von Raum sein. Dies kann evtl. über Vererbung bzw. Aggregation ausgedrückt werden.
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $WORKING\_TIME\_RANGE$

NL	: range of working times
Intention	: set of allowed working times
References	: PD/FMH3, PD/UH4
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $HOLIDAY\_TIME\_RANGE$

NL	: range of holiday times
Intention	: set of allowed holiday times
References	: PD/FMH3, PD/UH5
CustomerReq	: Wie sieht der erlaubte Wertebereich aus?

Domain $CALENDER\_TIME$

NL	: calender time
Intention	: Domain of the possible calender times that can be used for expressing a date.
References	: PD/FMH4, PD/FMH5
Comment	: Dieser Wertebereich verändert sich von Tag zu Tag, weil der Bereich stets vom aktuellen Tag bis ein Jahr später sein muß.
DeveloperReq	: Ist das ausreichend hier stets die Eingabe auf ein Jahr im voraus zu beschränken oder gibt es Ausnahmen?
DeveloperReas	: halber mal nachgefragt.

Domain $THREE\_POINT\_ACTUATOR\_RANGE = \{FORWARD, BACKWARD, STOP\}$

Intention	: Set of the possible values that can be assigned to a three point actuator.
References	: BD/Actuators

Domain $STRING$

Intention	: set of strings that cam be entered
Comment	: Hier stellt sich wieder die Frage, wo kann überhaupt etwas eingegeben werden?

Timed Function $now() \to TIME$

NL	: current time
Intention	: This function represents the current time, i.e. the current day, month, year, hour, minute, second, millisecond, ...
Scope	: eh
Comment	: Geeignete Umwandlungsfunktionen sollten bereit gestellt werden, wasaber mehr die Aufgabe der Implementierung ist.
CustomerReq	: Kann man auf so eine Größe zugreifen? Sie wird ja offensichtlich benötigt. Deshalb ist sie zunächst als eh eingestuft.
CustomerReas	: Ergänzung zur PD

Timed Predicate $daytime$

NL	:	=1: daytime =0: nighttime
Intention	:	true iff it is daytime
Scope	:	eh
References	:	PD/NFH4
CustomerReq	:	Wie kann die Maschine das feststellen? In PD/NFH4 ist von daytime und nighttime die Rede. Es wird davon ausgegangen, daß der eine Zustand genau dann vorliegt, wenn der andere nicht vorliegt, es also keine weiteren Zustände mehr gibt und daß immer genau einer vorliegt. Ist diese Annahme korrekt?

Domain *LIGHT_SCENES*

NL	:	values of light scenes
Intention	:	set of possible values that can be chosen for light scenes. The first component determines the value of the task light, the second component the value of the window ceiling light, the third component the value of the wall ceiling light, and the fourth component the ambient light level. In a room that is not an office the first component is not relevant.
Comment	:	Das muß noch überarbeitet werden. Eigene Klasse einbauen.

Description Class EnteredValue

Local Paramters:
Domain ENTERED_DOM
Domain TRANSFERED_DOM

Intention: Class providing all entities and properties to describe the entering and transfer of a value. It is assumed that the entered value can not be seen by the machine, i.e. there exists up to now no possibility to enter the value; the development of such a device is part of the overall development of the system. In our case this is the control panel, which still has to be developed. Therefore the timed function enteredValue has the scope eh. The first parameter is the domain of the value that can be entered and the second parameter is the domain of the value that is transfered to the machine.

Since it is a general class, not in all cases references are given here.

Comment: Genauere Beschreibung angeben. Insgesamt nochmal überarbeiten.

Comment: Überprüfen, ob die angegebenen Referenzen noch sinnvol sind. Ich vermute: Nein. (MK)

Comment: Da es die Control Panels noch nicht gibt, also noch keine Möglichkeit vorhanden ist, Daten einzugeben, ist enteredValue als eh deklariert. Wir sollten eine weitere Klasse in unseren Reuse-Vorrat aufnehmen, die die Situation modelliert, daß die Eingabemöglichkeit bereits vorhanden ist. Dann müßte folgendes bei der Intention stehen: ** It is assumed that the entered value can be seen by the machine. ** Therefore the timed function enteredValue has the scope ev. Möglich ist auch eine gemeinsame Klasse einzuführen, die den Scope als Parameter besitzt. Analoge Klassen sollten für die Sensoren und AKtuatoren eingeführt werden, d.h. eine Unterschiedung, ob ein Sensor/Aktuator bereits existiert oder nicht.

Comment: Wir müssen irgendwie noch ausdrücken, daß eine solche eingegebene Größe auf keine andere Art und Weise beeinflußt werden kann. Zumindest als Bemerkung mitaufnehmen.

Signature

Timed Function $enteredValue() \rightarrow ENTERED\_DOM$

NL	:	entered value
Intention	:	This function represents the last value entered by a user.
Scope	:	eh
References	:	PD/UH6, FMH3

Timed Function $transferedValue() \rightarrow TRANSFERED\_DOM$

NL	: transfered value
Intention	: This function represents the value decided by the machine to use.
Scope	: mh
References	: PD/UH6, FMH3
Comment	: Diese Größe wird vom Kunden erwähnt.

Timed Function $nearestValue : (ENTERED\_DOM) \rightarrow TRANSFERED\_DOM$

Intention	: This function returns the value which is the "nearest" one in the domain TRANSFERED_DOM w.r.t. the first argument.
Scope	: mh
References	: PD/FMH3
Comment	: Diese Größe wird explizit vom Kunden erwähnt. Beachte, daß die angegebenen Bereiche sich verändern können, deshalb ist dies eine Timed Function und keine Function.

Timed Function $transTime : () \rightarrow Time$

NL	: time for transference
Intention	: Time needed to check and transfer a value entered by a user to the value transfered to the machine.
Scope	: eh
Comment	: Als Zeitfunktion und nicht als Zeitkonstante eingeführt, damit die Sache generischer ist. Die Zeiten für alle Eingabegrößen kann wohl vereinheitlicht werden.
CustomerReq	: Welche Zeit ist hier angemessen?
CustomerReas	: Ergänzung zur PD

ReqSpec

Property P1

Formal	: $\Box(enteredValue \rhd^{nearestValue}_{\leq transTime} transferedValue)$
NL	: The transfered value is equal to the entered value w.r.t. the function nearestValue and the time delay transTime.
PatternInst	: DelayedCopy
Comment	: Hier brauchen wir wohl noch eine bessere und verständlichere Übersetzung.
CustomerReq	: Wie lange behält dieser Wert seine Gültigkeit? Durch welche Ereignisse kann er zurückgesetzt werden, z.B. wenn der Raum "lange" genug leer war?
CustomerReas	: Unklarheit in der PD

Description Class ControlledInput

Local Paramters:
Domain dom

Intention: This class contains all the entities and properties that are necessary to model the *controlled* input of an entity E by a user. *Controlled* here means that the facility manager prescribes an interval in which the value entered by the user has to be. Furthermore, the facility manager prescribes a default value which is valid as long as the user has not entered a value for E. As soon as the user has entered a value this is the actual value for E unless the user changes it by entering a new value. There is no other possibility to change the value for E.

The parameter of this class is the domain all values considered here have to be element of. These are the bounderies of the interval the facility manager can prescribe, the default value given by the facility manager and the value entered by the user.

Signature

Object $allowedRange : DynamicInterval(STRING, dom)$

Intention	: The range of possible user entries, preselected by the facility mangaer.
Comment	: vermutl. bel. Domaene - nicht nur Interval (CP) Ich würde es erstmal als Intervall lasse, (MK) Es müssen auch hier default-Werte für die Intervallgrenzen vorgegeben werden. So etwas kann man bei Objekten dieser Klasse immer verlangen!!

Object $default : EnteredValue(STRING, dom)$

Intention	: The default value given by the facility mangager. Note that by this declaration it is only guaranteed that the default value is an element of the domain dom. It is not guaranteed that it is a value of the interval specified by the facility manager. This is expressed by one of the properties of this class (see below).

Object $userValue : EnteredValue(STRING, dom)$

Intention	: The value entered by the user. Note that by this declaration it is only guaranteed that the entered value is an element of the domain dom. It is not guaranteed that it is a value of the interval specified by the facility manager. This is expressed by one of the properties of this class (see below).

Timed Function $value() \rightarrow dom$

Intention	: The currently valid value used by the system.
Scope	: mh

Timed Predicate $userValueEntered$

Intention	: This predicate is true iff the user has entered a value.
Scope	: ev

ReqSpec

Initialization

Property

Formal	: $\neg userValueEntered$
NL	: Initially the user has not entered a value.

Property I2

Formal	: $value = default.transfered$
NL	: The initial value is equal to the initial f.m. default value.
CustomerReq	: O.k. ?

End of Initialization

Property

Formal	: $\Box(userValueEntered \rightarrow \Box userValueEntered)$
NL	: If the user has entered a value once, from this time point it will be valid that he has entered a value.
References	: CustomerReview
Comment	: Da müssen wir die Referenz noch genauer angeben.

Property

Formal	: $\Box(userValueEntered \rightarrow (allowedRange.leq(userValue.transferedValue, allowedRange.start.transferedValue) \rightarrow value = allowedRange.start.transferedValue) \wedge (allowedRange.leq(allowedRange.end.transferedValue, userValue.transferedValue) \rightarrow value = allowedRange.start.transferedValue) \wedge ((\neg allowedRange.leq(userValue.transferedValue, allowedRange.start.transferedValue) \wedge \neg allowedRange.leq(allowedRange.end.transferedValue, userValue.transferedValue)) \rightarrow value = userValue.transferedValue))$
NL	: If the user has entered a value which is smaller than the lower bound of the allowed interval specified by the facility manager the current value is equal to the lower bound of the interval specified by the facility manager. If the user has entered a value which is greater than the upper bound of the allowed interval specified by the facility manager the current value is equal to the upper bound of the interval specified by the facility manager. Otherwise, if the entered value is within the interval specified by the facility manager, the current value is equal to the value entered by the user.
Comment	: Solch eine Fallunterscheidung ist mit Sicherheit auch ein Muster, das wir angeben sollten, insbesonder im Hinblick auf den Katalog. Bei dieser Anforderung ahndelt es sich sogar um eine bedingte Fallunterscheidung. Von vielen Mustern wird es wohl eine unbedingte und eine bedingte Variante geben. Was wiederum ein Muster auf einer höheren Ebene ist!

Property

Formal	: $\Box(allowedRange.leq(allowedRange.start.transferedValue, default.transferedValue) \wedge allowedRange.leq(default.transferedValue, allowedRange.end.transferedValue))$
NL	: The default value given by the facility manager has always to be within the allowed interval specified by the facility manager.
Comment	: Im Gegensatz zu den Eingaben des Benutzers hat hier der Kunde nicht genauer angegeben, wie sich das System bei einer Verletzung dieser Eigenschaft verhalten soll.

Description Class StaticInterval

Local Paramters:
Domain INT_DOM

Intention: Class providing all entities and properties to describe a static interval. "Static" means that the values of the start and end point of the interval can not change over time. The values are therefore constants. (See in contrast to this the Description Class DynamicInterval.)

The parameter is the domain the interval, i.e. the values of the start and the end point, belongs to.

Take the scope of the start and end point into account. It is assumed that they are mh.

Comment: Since it is a general class, no references are given here.

DeveloperReq: Sollen wir solche Verweise auf andere Klassen machen können?

Signature

Function $start : () \rightarrow INT\_DOM$

Intention	: starting point of the interval
Scope	: mh
Comment	: eingeführt, um die Darstellung übersichtlicher und kompakter zu machen

Function $end : () \rightarrow INT\_DOM$

Intention	: end point of the interval
Scope	: mh
Comment	: eingeführt, um die Darstellung übersichtlicher und kompakter zu machen

Predicate $leq : INT\_DOM, INT\_DOM$

NL	: $1 is smaller than or equal 2$
Intention	: Is true if the first argument is "smaller" than or equal the second argument.
Scope	: mh
Comment	: eingeführt, um die Darstellung übersichtlicher und kompakter zu machen
DeveloperReq	: Ist dies mh? Wenn ja, muß es ausdrücklich vom Kunden gewünscht sein. Oder gehen wir schon zu sehr in die Tiefe? Siehe auch Properties P1 und P2.

ReqSpec

Property P1

Formal	: $\Box(leq(start, end))$
NL	: The starting point of the interval is always smaller than the end point of the interval or both values are equal.
Comment	: eingeführt, um die Darstellung übersichtlicher und kompakter zu machen

Description Class DynamicInterval

Local Paramters:
Domain INT_DOM

Intention: Class providing all entities and properties to describe a dynamic interval. "Dynamic" means that the values of the start and end point of the interval can change over time. The values can be entered by a user. Therefore the start and end point are objects of the type EnteredValue. (See in contrast to this the Description Class StaticInterval.)

The parameter is the domain the interval, i.e. the values of the start and the end point, belongs to.

Comment: Since it is a general class, no references are given here.

DeveloperReq: Sollen wir solche Verweise auf andere Klassen machen können?

Signature

Object $start : EnteredValue(STRING, INT\_DOM)$

Intention	:	starting point of the interval
Scope	:	eh
References	:	PD/FMH4
Comment	:	eingeführt, um die Darstellung übersichtlicher und kompakter zu machen
DeveloperReq	:	Welchen Scope hat dieses Objekt?
CustomerReq	:	Wie wird dieser Wert dem System bekannt gemacht? Daher ist dieser Wert noch als eh eingestuft.

Object $end : EnteredValue(STRING, INT\_DOM)$

Intention	:	end point of the interval
Scope	:	eh
References	:	PD/FMH4
Comment	:	eingeführt, um die Darstellung übersichtlicher und kompakter zu machen
DeveloperReq	:	Welchen Scope hat dieses Objekt?
CustomerReq	:	Wie wird dieser Wert dem System bekannt gemacht? Daher ist dieser Wert noch als eh eingestuft.

Predicate $leq : INT\_DOM, INT\_DOM$

NL	:	$1 is smaller than or equal 2$
Intention	:	Is true if the first argument is "smaller" than or equal the second argument.
Scope	:	mh
Comment	:	eingeführt, um die Darstellung übersichtlicher und kompakter zu machen
DeveloperReq	:	Ist dies mh? Wenn ja, muß es ausdrücklich vom Kunden gewünscht sein. Oder gehen wir schon zu sehr in die Tiefe? Siehe auch Properties P1 und P2.

ReqSpec

Property P1

Formal	:	$\Box(leq(start.transferedValue, end.transferedValue))$
NL	:	The starting point of the interval is always smaller than the end point of the interval or both values are equal.
Comment	:	eingeführt, um die Darstellung übersichtlicher und kompakter zu machen
CustomerReq	:	Ist das eine zulässige Ergänzung?
CustomerReas	:	Ergänzung zur PD

Description Class CalenderInterval

Derived from the following classes:
Description Class DynamicInterval
Instantiated Parameters:
INT_DOM = CALENDER_TIME

Intention: Class providing all entities and properties to describe an interval of calender times. The start and end point of the interval can change over time. Additionally a timed predicate indicates whether the current date is in this interval or not.

Comment: Since it is a general class, no references are given here.

Comment: Diese Klasse wird eingeführt, um die Beschreibungen übersichtlicher und kompakter gestalten zu können.

Signature

Timed Predicate *actual*

NL	: =1 : in calender interval =0 : not in calender interval
Intention	: indicates whether the current date is in the calender interval given by the start and end value of the inherited interval.
Scope	: mh
References	: PD/Par3, PD/FMH4
Comment	: Solche Größen werden vom Kunden erwähnt.
CustomerReq	: Was für Auswirkungen hat der jeweilige Zustand?
CustomerReas	: Unklarheit in der PD

ReqSpec

Property P1

Formal	: $\square(actual \leftrightarrow smaller(start.transferedValue, now) \wedge smaller(now, end.transferedValue))$
NL	: actual is true if and only if the starting point of the calender interval is smaller than the current calender time and the current calender time is smaller than the end point of the calender interval.
DeveloperReq	: Ist das die intendierte Definition?
DeveloperReas	: zur PD

Description Class TempInterval

Derived from the following classes:
Description Class DynamicInterval
Instantiated Parameters:
INT_DOM = TEMP

Intention: Class providing all entities and properties to describe an interval of temperatures. The start and end point of the interval can change over time.

Description Class LightScene

Intention: This Class contains all the entities concerning a light scene for an arbitrary room. These are w.r.t. the DIC document:

- a name
- a value of the illuminance of the ambient light of a room
- An ordered list containing the installed illuminaires. The list defines the order in which the control system has to use the illuminaires to reach the given ambient light value. Since this list always consists of two entries we model this list by two functions firstAmbientIlluminaire and secondAmbientIlluminaire.

Aspects concerning a light scene of an office are presented in the derived description class LightSceneOffice.

CustomerReq: Im Dictionary wird nicht zwischen verschiedenen light scenes unterschieden. Eine solche Unterscheidung liegt aber doch aufgrund der unterschiedlichen Ausstattung der Räume doch vor, oder?

Signature

Domain $ILLUMINAIRES > \{WINDOW, WALL\}$

Intention	: Domain representing the possible illuminaires that can be used in the ordered list of installed illuminaires. $WINDOW$ stands for the window ceiling light and $WALL$ for the wall ceiling light.
References	: DIC/light scene

Timed Function $() \rightarrow STRING$

Intention	: Name of the light scene.
References	: DIC/light scene

Timed Function $ambientIlluminace$

Intention	: Value in the light scene defining the desired illuminance of the ambient light of a room.
References	: DIC/light scene

Timed Function $firstAmbientIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the first illuminaire that should be used to reach the desired ambient light of a room.
References	: DIC/light scene

Timed Function $secondAmbientIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the second illuminaire that should be used to reach the desired ambient light of a room.
References	: DIC/light scene

ReqSpec

Property

Formal	: $\Box(\neg firstAmbientIlluminaire = secondAmbientIlluminaire)$
CustomerReq	: Ist das eine sinnvolle Forderung? Wenn ja entsprechende für die abgeleitete Klasse einführen.

Description Class LightSceneOffice

Derived from the following classes:
Description Class LightScene

Intention: This Class contains all the entities concerning a light scene for an office. In addition to the entities belonging to the basic class LightScene this class also contains the following entities.

- a value of the illuminance for the desk
- An ordered list containing the installed illuminaires. The list defines the order in which the control system has to use the illuminaires to reach the given illuminance value for the desk.
- An extension of the derived ordered list defining the illuminaires to reach the desired ambient light of a room. Therefore the following extensions are introduced: the set ILLUMINAIRES of possible values is extended by the value *TASK* and the function thirdAmbientIlluminaire is introduced.

Signature

Value $TASK : ILLUMINAIRES$

Intention	: Extension of the domain representing the possible illuminaires that can be used in the ordered list of installed illuminaires. *TASK* stands for the task light.
References	: BD/Offices

Timed Function $thirdAmbientIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the third illuminaire that should be used to reach the desired ambient light of a room.
References	: DIC/light scene

Timed Function $firstDeskIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the first illuminaire that should be used to reach the desired desk light of a room.
References	: DIC/light scene

Timed Function $secondDeskIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the second illuminaire that should be used to reach the desired desk light of a room.
References	: DIC/light scene

Timed Function $thirdDeskIlluminaire() \rightarrow ILLUMINAIRES$

Intention	: This function represents the third illuminaire that should be used to reach the desired desk light of a room.
References	: DIC/light scene

Description Class NoMalfunctionSensor

Local Paramters:
Domain NO_MAL_ENV_DOMAIN
Domain NO_MAL_MEASURED_DOMAIN
Domain NO_MAL_CONVERTED_DOMAIN

Intention: Class providing all entities and properties to describe the behavior of a *sensor* if there is *no malfunction*.

Comment: Since it is a general class, no explicit references are given here.

DeveloperReq: Zur Zeit haben wir keine Werteparameter bei Klassenbeschreibungen, daher müssen die Werte für die Zeiten bei der Aggregation eines Objektes dieser Klasse noch explizit als Eigenschaft angegeben werden, aber das dürfte kein Problem sein. Später kann man entsprechende Abkürzungen einführen.

Signature

Timed Function $envEntity() \rightarrow NO\_MAL\_ENV\_DOMAIN$

Intention	: Value of the measured entity in the real world. This function represents the situation as it is in the real world and not how it is seen by the machine.
Scope	: eh

Timed Function $measuredEntity() \rightarrow NO\_MAL\_MEASURED\_DOMAIN$

Intention	: Analog measured value of the entity of the environment.
Scope	: eh

Timed Function $convertedEntity() \rightarrow NO\_MAL\_CONVERTED\_DOMAIN$

Intention	: Value of measuredEntity but converted to a digital value.
Scope	: ev

Time Constant *reactionTime*

Intention	: Reaction time of the sensor, i.e. the time between a change of the value of envEntity and the time point measuredEntity has reached 90 percent of the difference.
Scope	: eh
CustomerReq	: Ist das die korrekte Erläuterung der Reaktionszeit bei analogen Sensoren?
CustomerReas	: Entwickler kennt sich nicht aus.

Time Constant *conversionTime*

Intention	: Time needed to convert the analog value measuredEntity to the digital value convertedEntity.

Timed Function $modifyReaction(NO\_MAL\_ENV\_DOMAIN) \rightarrow NO\_MAL\_MEASURED\_DOMAIN$

Intention	: Function representing the way in which the value envEntity is modified during the reaction of the sensor.
CustomerReq	: Ist das eine Timed Function oder eine Function?

Timed Function $modifyConversion(NO\_MAL\_MEASURED\_DOMAIN) \rightarrow NO\_MAL\_CONVERTED\_DOMAIN$

Intention	: Function representing the way in which the value measuredEntity is modified during the conversion into a digital value.
CustomerReq	: Ist das eine Timed Function oder eine Function?

DomKnow

Property

Formal	: $\Box(envEntity \rhd^{modifyReaction}_{\leq reactionTime} measuredEntity)$
PatternInst	: DelayedCopy
Comment	: Hier fehlt noch die NL.

Property

Formal	: $\Box(measuredEntity \rhd^{modifyConversion}_{\leq conversionTime} convertedDigitalEntity)$
PatternInst	: DelayedCopy
Comment	: Hier fehlt noch die NL.

Description Class MalfunctionSensor

Local Paramters:
Domain MAL_CONVERTED_DOMAIN

Intention: Class providing all entities and properties to describe the behavior of a *sensor* if there is a *malfunction*. A sensor is a sensor that has not a binary set of measured values.

Comment: Since it is a general class, no explicit references are given here.

DeveloperReq: Zur Zeit haben wir keine Werteparameter bei Klassenbeschreibungen, daher müssen die Werte für die Zeiten bei der Aggregation eines Objektes dieser Klasse noch explizit als Eigenschaft angegeben werden, aber das dürfte kein Problem sein. Später kann man entsprechende Abkürzungen einführen.

Signature

Object $manualEnteredVal : EnteredValue(MAL\_CONVERTED\_DOMAIN)$

Intention	: Value for this sensor entered manually by a user in the case of a malfunction of this sensor.
Scope	: eh
Comment	: Es muß evtl. noch berücksichtigt werden, daß die Eingabe nur im Fehlerfall möglich ist.
CustomerReq	: Wann kann diese Eingabe erfolgen? Nur im Fehlerfall?

Timed Predicate $stateManualEnteredValue$

Intention	: This predicate is true iff a value has been entered manually.
Scope	: mh
Comment	: Ist vom Kunden eingeführt worden.

Timed Function $automaticDetermValue() \rightarrow MAL\_CONVERTED\_DOMAIN$

Intention	: Function specifying in which way the value of the sensor is determined automatically by the machine in the case of a malfunction of the sensor.
Scope	: mh

Timed Function $valueMal() \rightarrow MAL\_CONVERTED\_DOMAIN$

Intention	: Represents the value of the sensor used by the machine in the case of a malfunction of the sensor.
Scope	: mh

ReqSpec

Initialization

Property

Formal	: $\neg stateManualEnteredValue$
NL	: Initially no value for the case of a malfuntion is entered manually.
CustomerReq	: Ist ja wohl ok so, oder?

End of Initialization

Property

Formal	: $\Box(stateManualEnteredValue \rightarrow (valueMal = manualEnteredVal.transfered) \wedge (\neg stateManualEnteredValue \rightarrow (valueMal = automaticDetermValue)))$
NL	: If a value for the status of the sensor in the case of a malfunction has been entered manually this is the value used by the machine in the case of a malfunction of the sensor. Otherwise, the value used by the machine in the case of a malfunction of the sensor is the value determined automatically in the case of a malfunction of the sensor.
References	: CustomerReviewResults?

Description Class Sensor

Local Paramters:

Domain ENV_DOMAIN
Domain MEASURED_DOMAIN
Domain CONVERTED_DOMAIN

Derived from the following classes:

Description Class NoMalfunctionSensor
Instantiated Parameters:
NO_MAL_ENV_DOMAIN = ENV_DOMAIN
NO_MAL_MEASURED_DOMAIN = MEASURED_DOMAIN
NO_MAL_CONVERTED_DOMAIN = CONVERTED_DOMAIN
Description Class MalfunctionSensor
Instantiated Parameters:
MAL_CONVERTED_DOMAIN = CONVERTED_DOMAIN

Intention: Class providing all entities and properties to describe a sensor.

Comment: Since it is a general class, no explicit references are given here.

DeveloperReq: Zur Zeit haben wir keine Werteparameter bei Klassenbeschreibungen, daher müssen die Werte für die Zeiten bei der Aggregation eines Objektes dieser Klasse noch explizit als Eigenschaft angegeben werden, aber das dürfte kein Problem sein. Später kann man entsprechende Abkürzungen einführen.

Signature

Timed Predicate *malDetEnv*

Intention	: true iff the sensor existing in the real world has a malfunction
Scope	: eh
Comment	: Diese Größe kann verwendet werden, um Anforderungen zu formulieren.

Timed Predicate *malDetMach*

Intention	: true iff the sensor has a malfunction. This predicate models the situation concerning the malfunction status of the sensor as it seen and used in the machine.
Scope	: mh
Comment	: Dies steht zwar nicht explizit in der PD, der Kunde wird aber sicherlich davon überzeugt werden können, daß er über diese Größe auch reden möchte, sonst macht das manuelle Eingeben keinen Sinn.

Timed Predicate *stateManualEnteredMal*

Intention	: This predicate is true iff it has been manually entered whether there is a malfunction of the sensor or whether there is no malfunction of it.
Scope	: mh
References	: CustomerReviewResults, PD?

Object $manualEnteredMal : EnteredValue(ENTERED\_DOM = BINARY, TRANSFERED\_DOM = BINARY)$

Intention	: Value entered manually by a user concerning a malfunction of the sensor. 1 stands for a malfunction, 0 stands for no malfunction.
Scope	: eh

Timed Predicate $automaticDetMal$

Intention	: This predicate is true iff a malfunction of the sensor has been detected automatically, i.e. not by entering it manually.
Scope	: mh
CustomerReq	: Hier wird noch eine geeignete Definition vom Domänenexperten benötigt, und zwar für jeden Sensor!

Timed Predicate $malfunction$

Intention	: This predicate is true iff the machine assumes that there is a malfunction of the sensors.
Scope	: mh
Comment	: Diese Größe ist zur abkürzenden Schreibweise und zur Strukturierung eingeführt worden. Sie kann gemäß der Definition unten expandiert werden.

Timed Function $usedValue() \rightarrow CONVERTED\_DOMAIN$

Intention	: This function models the value used by the machine. This depends on whether there is a malfunction of the sensor or not.
Scope	: mh
Comment	: Diese Größe wird explizit vom Kunden eingeführt.

Timed Predicate $FMInformed$

Intention	: This predicate is true iff the facility manager is informed about a malfunction of this sensor. This has to be established by sending an email.
Scope	: eh
References	: CustomerReviewResults, PD/U11
DeveloperReq	: Hier bräuchten wir eigentlich auch wieder so etwas wie eine FIFO, die aufgetretenen Fehler mit Informiertwerden koppelt. Oder wir müßten so etwas einführen, daß ein Signal losgeschickt wurde. Es wurde versucht das unten anssatzweise zu modellieren.
CustomerReq	: Soll der FM auch darüber informiert werden, wenn der Fehler wieder behoben wurde?

Timed Predicate $userInformed$

Intention	: This predicate is true iff the user is informed about a malfunction of this sensor. This has to be established by sending an email.
Scope	: eh
References	: CustomerReviewResults, PD/U11
DeveloperReq	: Hier bräuchten wir eigentlich auch wieder so etwas wie eine FIFO, die aufgetretenen Fehler mit Informiertwerden koppelt. Oder wir müßten so etwas einführen, daß ein Signal losgeschickt wurde. Es wurde versucht das unten anssatzweise zu modellieren.
CustomerReq	: Soll der Benutzer über Fehler bei allen Sensoren informiert werden. In der PD ist nur von motDet und outlight die Rede? Soll der Benutzer auch darüber informiert werden, wenn der Fehler wieder behoben wurde?

Time Constant $T_{infDelay}$

Intention	: maximal delay between an event concerning this sensor (e.g. a malfunction or the repair of a malfunction) a user or the facility manager has to be informed about and the time point the user or the facility manager is informed.

Time Constant T_{info}

Intention	: Maximal time span a user or the facility manager is informed.
DeveloperReq	: Mal nochmal überlegen, ob das die geschickteste Modellierung ist.

ReqSpec

Initialization

Property

Formal	: $\neg stateManualEnteredMal$
NL	: Initially it has not been entered manually whether there is a malfunction of the sensor or whether there is no malfunction of it.
CustomerReq	: Das ist ja wohl ok so!?

Property

Formal	: $\neg userInformed$
NL	: Initially the user is not informed about a malfunction of this sensor.
CustomerReq	: Ist ja wohl ok so?

Property

Formal	: $\neg FMInformed$
NL	: Initially the facility manager is not informed about a malfunction of this sensor.
CustomerReq	: Ist ja wohl ok so?

End of Initialization

Property

Formal	:	$\Box(malfunction \leftrightarrow (stateManualEnteredMal \wedge manualEnteredMal) \vee (\neg stateManualEnteredMal \wedge automaticDetMal))$
NL	:	The machine assumes a malfunction of the sensor iff one of the follwoing conditions is fulfilled: • a malfunction has been entered manually • it has not been entered manually that there is no malfunction and a malfunction has been detected automatically
CustomerReq	:	Ist das die korrekte Priorissierung: Manuell über automatisch?

Property

Formal	:	$\Box([userInformed] \rightarrow \Diamond_{=T_{info}} \neg userInformed)$
NL	:	Whenever a user is informed the predicate userInformed becomes false again within T_info time units. By this requirement it is expressed, that the user is not permanently informed, but that is a kind of signal, e.g. an email.

Property

Formal	:	$\Box([FMInformed] \rightarrow \Diamond_{=T_{info}} \neg FMInformed)$
NL	:	Whenever the facility manager is informed the predicate FMInformed becomes false again within T_info time units. By this requirement it is expressed, that the user is not permanently informed, but that is a kind of signal, e.g. an email.

Property

Formal	:	$\Box(\neg malfunction \rightarrow usedValue = convertedEntity.transferedValue)$
NL	:	Whenever the machine assumes that the sensor is working correctly the value used by the machine corresponds with the converted version of the measured value.
DeveloperReq	:	Hier lösen wir einen Konflikt auf, ohne ihn explizit anzugeben, nämlich den Konflikt zwischen Normalfall (= NichtFehlerfall) und Fehlerfall. In dieser Anforderung stecken zwei Sachen: - Wie sich machUsedValue im Normalfall verhalten soll. - daß der Fehlerfall Priorität hat. Dafür sollten wir uns was überlegen, zumindest hier explizit angeben. Diese Anforderung ist im wesentlichen vom motion detector übernommen wurden. Dort wurde sie aber noch mit eh-Größen beschrieben und es wurden Zeitverzögerungen betrachtet. Diese Zeitverzögerungen stecken meiner Meinung nach jetzt in measuredEntity, convertedEntity und in enteredFault. Ist das so o.k.?

Property

Formal	:	$\Box([\neg malfunction] \rightarrow \Diamond_{\leq T_{infDelay}} userInformed)$
NL	:	Whenever the sensor is again working correctly after it has had a malfunction, the user is informed about this.
CustomerReq	:	Ist das ok so?

Property

Formal	:	$\Box([\neg malfunction] \rightarrow \Diamond_{\leq T_{infDelay}} FMInformed)$
NL	:	Whenever the sensor is again working correctly after it has had a malfunction, the facility manager is informed about this.
CustomerReq	:	Ist das ok so?

Property

Formal	:	$\Box(malfunction \rightarrow usedValue = valueMal)$
NL	:	Whenever the machine assumes that the sensor is not working correctly the value used by the machine corresponds with the value computed for this case.

Property

Formal	:	$\Box([malfunction] \rightarrow \Diamond_{\leq T_{infDelay}} userInformed)$
NL	:	Whenever a malfunction occurs, the user is informed about this malfunction.

Property

Formal	:	$\Box([malfunction] \rightarrow \Diamond_{\leq T_{infDelay}} FMInformed)$
NL	:	Whenever a malfunction occurs, the user is informed about this malfunction.

Description Class RoomTempSensor

Derived from the following classes:

Description Class Sensor

Instantiated Parameters:

ENV_DOMAIN = REAL

MEASURED_DOMAIN = ???

CONVERTED_DOMAIN = ROOM_TEMP

DeveloperReq: What is the 'measuredDomain'?

DeveloperResp: Ich würde mal ROOM_TEMP einsetzen; ist aber eine Frage an den Domänenexperten.(MK)

Signature

Domain $ROOM\_TEMP = \{-40 \ldots 70\} < INTEGER$

Intention	:	Range of possible values transfered by a room temperature sensor.
References	:	BD/Sensor:room temperature sensor
DeveloperReq	:	Es ist noch die Frage, ob wir diese Domäne nicht eigentlich global definieren müßten.

DomKnow

Property D1

Formal	:	$\Box(conversionTime = 160000)$
NL	:	The conversion time of a room temperature sensor is 160 seconds.
References	:	BD/Sensor:room temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.
CustomerReq	:	Wieso ist die Conversion Time so groß?

Property D2

Formal	:	$\Box(reactionTime = 10000)$
NL	:	The reaction time of a room temperature sensor is 10 seconds.
References	:	BD/Sensor:room temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.

Description Class OutdoorTempSensor

Derived from the following classes:

Description Class Sensor

Instantiated Parameters:

ENV_DOMAIN = REAL

MEASURED_DOMAIN = ???

CONVERTED_DOMAIN = OUT_TEMP

DeveloperReq: What is the 'measuredDomain'? Wo wird ein Objekt dieses Typs verwendet?

DeveloperResp: Ich würde mal OUT_TEMP einsetzen; ist aber eine Frage an den Domänenexperten.(MK)

Signature

Domain $OUT\_TEMP = \{-40 \ldots 60\} < NAT$

Intention	:	Range of possible values transfered by an outdoor temperature sensor.
References	:	BD/Sensor:outdoor temperature sensor
DeveloperReq	:	Es ist noch die Frage, ob wir diese Domäne nicht eigentlich global definieren müßten.

DomKnow

Property D1

Formal	:	$\Box(conversionTime = 160000)$
NL	:	The conversion time of an outdoor temperature sensor is 160 seconds.
References	:	BD/Sensor:outdoor temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.

Property D2

Formal	:	$\Box(reactionTime = 120000)$
NL	:	The reaction time of an outdoor temperature sensor is 120 seconds.
References	:	BD/Sensor:outdoor temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.

Description Class WaterTempSensor

Derived from the following classes:

Description Class Sensor
Instantiated Parameters:
ENV_DOMAIN = REAL
MEASURED_DOMAIN = ???
CONVERTED_DOMAIN = WATER_TEMP

DeveloperReq: What is the 'measuredDomain'?

DeveloperResp: Ich würde mal WATER_TEMP einsetzen; ist aber eine Frage an den Domänenexperten.

Signature

Domain $WATER\_TEMP = \{0 \ldots 100\} < NAT$

Intention	:	Range of possible values transfered by a water temperature sensor.
References	:	BD/Sensor:water temperature sensor
DeveloperReq	:	Es ist noch die Frage, ob wir diese Domäne nicht eigentlich global definieren müßten.

DomKnow

Property D1

Formal	:	$\Box(conversionTime = 160000)$
NL	:	The conversion time of an outdoor temperature sensor is 160 seconds.
References	:	BD/Sensor:water temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.

Property D2

Formal	:	$\Box(reactionTime = 100000)$
NL	:	The reaction time of an outdoor temperature sensor is 120 seconds.
References	:	BD/Sensor:water temperature sensor
Comment	:	Beachte, die Basiseinheit ist ms.
DeveloperReq	:	Sollen wir bei PatternInst hier [](p) angeben? Dies könnte für Analyse-Zwecke und für einen "Katalog" interessant sein.

Description Class OutdoorLightSensor

Derived from the following classes:

Description Class Sensor
Instantiated Parameters:
ENV_DOMAIN = REAL
MEASURED_DOMAIN = ???
CONVERTED_DOMAIN = OUT_LIGHT_VALUES

DeveloperReq: What is the 'measuredDomain'? Ist der Zugriff auf das lokale OUT_LIGHT_VALUES erlaubt?

DeveloperResp: Sollten wir uns mal drüber unterhalten. Zur Zeit tendiere ich mit etwas Widerwillen zu Nein. Dasselbe Problem haben wir auch oben.(MK)

Signature

Domain $OUT\_LIGHT\_VALUES = \{1 \ldots 10000\}$

NL	:	outdoor light values
Intention	:	set of possible values reported by the outdoor light sensor
References	:	BD/Sensors:outdoor light sensor

Timed Function $lastCorrectOutLight() \rightarrow OUT\_LIGHT\_VALUES$

NL	:	last correctly measured outdoor light
Intention	:	last corectly measured value of the outdoor illuminance
Scope	:	eh
References	:	PD/NF2
Comment	:	Wenn diese Größe eh ist, brauchen wir noch eine die mh ist und wir brauchen eine Vorschrift, die besagt, wie diese dann zu bestimmen ist. Oder ist das schon Implementiereung? Ich (MK) bin der Meinung, daß es sich hierbei um Spezifikation handelt, nicht um Implementierung. Wir brauchen hier auf jeden Fall den Domänenexperten.

ReqSpec

Property

Formal	:	$\Box(automaticDetermValue = lastCorrectOutLight)$
NL	:	The value of the sensor determined automatically in the case of a malfunction of the outdoor light sensor corresponds with the last correctly measured value.
References	:	PD/NF2
CustomerReq	:	Dies ist eine Anforderung; wie lautet die entsprechende Spezifikation?

Description Class BinarySensor

Derived from the following classes:

Description Class Sensor

Instantiated Parameters:

ENV_DOMAIN = BINARY

MEASURED_DOMAIN = BINARY

CONVERTED_DOMAIN = BINARY

Intention: Class providing all entities and properties to describe a binary sensor. A binary sensor is a sensor that can return a 0 or a 1.
Note: This class can be used to describe a binary sensor that *already exists* in the environment. This is expressed as follows: At first the value measured by the sensor, namely measuredEntity, has the scope "ev". If the sensor does not exist, i.e. the sensor itself has to be developed, the scope of measuredEntity has to be "eh". Secondly, the property D1 given in this class is characterized as domain knowledge (DomKnow). Otherwise, if the sensor does not exist, it has to be a requirement, i.e. characterized as a requirement/specification (ReqSpec).

Comment: Since it is a general class, no references are given here.

DeveloperReq: Sollen wir diese Klasse nicht besser ExistBinarySensor nennen, um deutlich zu machen, daß es diese Art von Sensoren bereits gibt?

DomKnow

Property

Formal	: $\Box(conversionTime = 0)$
NL	: The time needed to convert the analog value measuredEntity to the digital value convertedEntity is 0 in the case of a binary sensor.
CustomerReq	: Stimmt das?

Property

Formal	: $\Box(\forall b \in BINARY\ modifyReaction(b) = b)$
NL	: For a binary sensor the function representing the way in which the value envEntity is modified during the reaction of the sensor is the identity.
CustomerReq	: Kann man davon ausgehen?

Property

Formal	: $\Box(\forall b \in BINARY\ modifyConversion(b) = b)$
NL	: For a binary sensor the function representing the way in which the value measuredEntity is modified during the conversion into a digital value is the identity. Note that no conversion is needed.
CustomerReq	: Kann man davon ausgehen?

Description Class Light

Derived from the following classes:

Description Class BinarySensor

Intention: Class representing the different kinds of lights. This class models the status line mentioned in the BD.

CustomerReq: Dieser Sensor wird in der BD nicht genauer beschrieben.

DomKnow

Property

Formal	: $\Box(reactionTime = 10)$
NL	: The reaction time of a status line is 10 ms.
References	: BD/Sensors:dimmable light(sll)

Description Class DoorContact

Derived from the following classes:

Description Class BinarySensor

Intention: Class representing the different kinds of door contacts.

DomKnow

Property

Formal	:	$\Box(reactionTime = 10)$
NL	:	The reaction time of a door contact is 10 ms.
References	:	BD/Sensors:door closed contact, BD/Sensors:door open contact, BD/Sensors:door lock contact

Description Class MotDet

Derived from the following classes:

Description Class BinarySensor

ReqSpec

Property

Formal	:	$\Box(automaticDetermValue = 1)$
NL	:	Whenever the machine assumes that the motion detector is not working correctly the machine behaves as if the motion detector would deliver a 1.

Description Class BinaryActuator

Intention: Class providing all entities and properties to describe a binary actuator. A binary actuator is an actuator that is controlled using a 0 or a 1. This is modelled here by using timed predicates. Note: This class can be used to describe a binary actuator that already exists in the environment. This is expressed as follows: At first the value adjusted by the actuator, namely adjustedEntity, has the scope "ev". If the actuator does not exist, i.e. the actuator itself has to be developed, the scope of adjustedEntity has to be "eh". Secondly, the property D1 given in this class is characterized as domain knowledge (DomKnow). Otherwise, if the actuator does not exist, it has to be a requirement, i.e. characterized as a requirement/specification (ReqSpec).

Comment: Since it is a general class, no explicit references are given here.

Comment: Auch hier wird nur der Nicht-Fehlerfall betrachtet.

Signature

Timed Predicate *envEntity*

Intention	:	value of the entity of the environment that is adjusted
Scope	:	eh

Timed Predicate *adjustedEntity*

Intention	:	value that should be adjusted for the entity of the environment
Scope	:	mv
Comment	:	Bessere Formulierung der Intention wünschenswert.

Time Constant *reactionTime*

Intention	:	reaction time of the actuator, i.e. the time between a change of the value of adjustedEntity and the time point envEntity changes its value.
Scope	:	eh
CustomerReq	:	Ist das die korrekte Erläuterung der Reaktionszeit bei binären Aktuatoren?
CustomerReas	:	Entwickler kennt sich nicht aus.

DomKnow

Property D1

Formal	:	$\Box(adjustedEntity \Leftrightarrow_{\leq reactionTime} envEntity)$
PatternInst	:	DelayEquivalence
Comment	:	Hier fehlt noch die NL.

Description Class ControlSystemActive

Intention: Class representing the signal "control system is active".

DomKnow

Property

Formal	:	$\Box(reactionTime = 10)$
NL	:	The reaction time of a signal "control system is active" is 10 ms.
References	:	BD/Sensors:dimmable light(cia)
Comment	:	Steht irrtümmlicherweise bei den Sensoren und ist dort auch noch als "status line" gekennzeichnet.
CustomerReq	:	Ist das vielleicht auch ein ToggleActuator? Oder kann man auch permanent eine 1 schicken?

Description Class ToggleActuator

Intention: Class providing all entities and properties to describe a to toggle actuator. A toggle actuator is an actuator that is controlled using a 0 or a 1. This is modelled here by using timed predicates. Note: This class can be used to describe a toggle actuator that already exists in the environment. This is expressed as follows: At first the value toggled by the actuator, namely toggledEntity, has the scope "ev". If the actuator does not exist, i.e. the actuator itself has to be developed, the scope of toggledEntity has to be "eh". Secondly, the properties D1 and D2 given in this class are characterized as domain knowledge (DomKnow). Otherwise, if the actuator does not exist, they have to be requirements, i.e. characterized as requirement/specification (ReqSpec).

Comment: Since it is a general class, no explicit references are given here.

Comment: Auch hier wird nur der Nicht-Fehlerfall betrachtet.

DeveloperReq: Ist die Ausdrucksweise mit "toggle" als Verb sinnvoll?

CustomerReq: Diese Aktuatorklasse wird eingeführt, um "pulse" aus der BD modellieren zu können. Dieser ist nirgendwo genauer erläutert. Ist seine Modellierung hier ok?

Signature

Timed Predicate *envEntity*

Intention	: value of the entity of the environment that is controlled
Scope	: eh
Comment	: Bessere Formulierung der Intention wünschenswert.

Timed Predicate *toggledEntity*

Intention	: value that should be toggled to control the entity of the environment
Scope	: mv
Comment	: Bessere Formulierung der Intention wünschenswert.

Time Constant *reactionTime*

Intention	: reaction time of the actuator, i.e. the time between a change of the value of adjustedEntity and the time point envEntity changes its value.
Scope	: eh
CustomerReq	: Ist das die korrekte Erläuterung der Reaktionszeit bei binären Aktuatoren? Gibt es so etwas auch bei dieser Art von Aktuatoren?
CustomerReas	: Entwickler kennt sich nicht aus.

DomKnow

Property D1

Formal	: $\Box(envEntity \wedge [toggledEntity] \rightarrow \Diamond_{\leq reactionTime} \neg envEntity)$
NL	: If the envEntity is set, i.e. equal to 1, and a toggle signal arrives, the envEntity is set to 0.
PatternInst	: Bounded Response
Comment	: Hier fehlt noch die NL.
DeveloperReq	: Gibt es noch andere Möglichkeiten, diesen Wert zu beeinflussen? Das sollten wir auch ausdrücken? (Damit sind wir bei dem FrameProblem.)
CustomerReq	: Wie schnell hinter einander kann so ein Ereignis [toggledEntity] kommen?

Property D2

Formal	: $\Box(\neg envEntity \wedge [toggledEntity] \rightarrow \Diamond_{\leq reactionTime} envEntity)$
NL	: If the envEntity is not set, i.e. equal to 0, and a toggle signal arrives, the envEntity is set to 1.
PatternInst	: Bounded Response
Comment	: Hier fehlt noch die NL.
DeveloperReq	: Gibt es noch andere Möglichkeiten, diesen Wert zu beeinflussen? Das sollten wir auch ausdrücken? (Damit sind wir bei dem FrameProblem.)
CustomerReq	: Wie schnell hinter einander kann so ein Ereignis [toggledEntity] kommen?

Description Class Pulse

Derived from the following classes:
Description Class ToggleActuator

Intention: Class representing the pulse line of a dimmable light.

DomKnow

Property

Formal	:	$\square(reactionTime = 10)$
NL	:	The reaction time of a pulse line/actuator is 10 ms.
References	:	BD/Sensors:relais
Comment	:	Domänenexperte hat angegeben, daß es sich bei dem als relais gekennzeichneten Sensor um den hier gemeinten Aktuator handelt!

Description Class RangeActuator

Local Paramters:
Domain ADJUSTED_DOMAIN
Domain ENVDOMAIN

Intention: Class providing all entities and properties to describe a range actuator. The first parameter represents the domain of the value that is set by the machine. The second parameter represents the domain of the corresponding entity of the environment. Note: This class can be used to describe a range actuator that already exists in the environment. This is expressed as follows: At first the value adjusted by the actuator, namely adjustedEntity, has the scope "mv". If the actuator does not exist, i.e. the actuator itself has to be developed, the scope of adjustedEntity has to be "eh". Secondly, the property D1 given in this class is characterized as domain knowledge (DomKnow). Otherwise, if the actuator does not exist, it has to be a requirement, i.e. characterized as a requirement/specification (ReqSpec).

Comment: Since it is a general class, no explicit references are given here.

Comment: Auch hier wird nur der Nicht-Fehlerfall betrachtet.

Signature

Timed Function $envEntity() \rightarrow ENV\_DOMAIN$

Intention	:	value of the entity of the environment that is adjusted
Scope	:	eh

Timed Function $adjustedEntity() \rightarrow ADJUSTED\_DOMAIN$

Intention	:	value that should be adjusted for the entity of the environment
Scope	:	mv
Comment	:	Bessere Formulierung der Intention wünschenswert.

Time Constant $reactionTime$

Intention	:	reaction time of the actuator, i.e. the time between a change of the value of adjustedEntity and the time point envEntity changes its value.
Scope	:	eh
CustomerReq	:	Ist das die korrekte Erläuterung der Reaktionszeit bei Bereichsaktuatoren?
CustomerReas	:	Entwickler kennt sich nicht aus.

Timed Function $modifyReaction(ADJUSTED\_DOMAIN) \rightarrow ENV\_DOMAIN$

Intention	:	Function representing the way in which the value adjustedEntity is modified during the reaction of the actuator.
CustomerReq	:	Ist das eine Timed Function oder eine Function?

DomKnow

Property

Formal	: $\Box(adjustedEntity \rhd^{modifyReaction}_{\leq reactionTime} envEntity)$
PatternInst	: DelayCopy
Comment	: Hier fehlt noch die NL.

Description Class Dimmer

Derived from the following classes:

Description Class RangeActuator
Instantiated Parameters:
ADJUSTED_DOMAIN = PERCENT
ENVDOMAIN = PERCENT

DomKnow

Property

Formal	: $\Box(reactionTime = 10)$
NL	: The reaction time of a signal "control system is active" is 10 ms.
References	: BD/Sensors:dimmable light(cia)
Comment	: Steht irrtümmlicherweise bei den Sensoren.

Description Class ThreePointActuator

Local Paramters:
Domain ADJUSTED_DOMAIN

Intention: Class providing all entities and properties to describe a three point actuator. A three point actuator is an actuator which can be in three control states: forward, backward, stop.

Comment: Since it is a general class, not always references are given here.

DeveloperReq: Es kann nicht die Aufgabe eines solchen Aktuators sein, die aktuelle Position zu protokollieren, dies muß intern in der Maschine erfolgen. Angegeben werden müßte noch, wie sich der Zustand verändert, wenn sich der Aktuator eine bestimmte Zeit in einem Zustand befindet. Das steht aber nirgends, also muß das der Kunde auch noch sagen, oder?

CustomerReq: Kann man einen allgemeinen Initialzustand festlegen, z.B. den Zustand "stop"?

Signature

Domain $CONTROL\_STATES = \{forward, backward, stop\}$

Intention	: The three possible control states of a three point actuator.
References	: BD/Actuators

Timed Function $envStatus() \rightarrow CONTROL\_STATES$

Intention	: the current state of the environment entity
Scope	: eh

Timed Function $adjustedStatus() \rightarrow CONTROL\_STATES$

Intention	: the current state of the entity controlled by the machine
Scope	: mv

Time Constant $reactionTime$

Intention	: reaction time of the actuator, i.e. the time between a change of the value of adjustedStatus and the time point envStatus changes its value.
Scope	: eh
CustomerReq	: Ist das die korrekte Erläuterung der Reaktionszeit bei Dreipunkt-Aktuatoren?
CustomerReas	: Entwickler kennt sich nicht aus.

Timed Function $modifyReaction(CONTROL\_STATES) \rightarrow CONTROL\_STATES$

Intention	: Function representing the way in which the value adjustedStatus is modified during the reaction of the actuator.
CustomerReq	: Ist das eine Timed Function oder eine Function? In diesem Fall dürfte es wohl immer die Identität sein!?

DomKnow

Property

Formal	: $\Box(adjustedStatus \rhd^{modifyReaction}_{\leq reactionTime} envStatus)$
PatternInst	: DelayCopy
Comment	: Hier fehlt noch die NL.

Property

Formal	: $\Box(\forall s \in CONTROL\_STATES : modifyReaction(s) = s)$
NL	: The modification function of a three point actuator is the identity. This measn that the value determined by the machine does not change during its transfer to the environment value.
CustomerReq	: Ist das tatsächlich so?
CustomerReas	: Vervollständigung des Domänenwissens.

Description Class UpperSashOpener

Derived from the following classes:
Description Class ThreePointActuator

CustomerReq: Dieser Aktuator ist nirgends beschrieben. Handelt es sich überhaupt um eine ThreePointActuator? Wie lauten die Angaben?

Description Class SunBlind

Derived from the following classes:
Description Class ThreePointActuator
Instantiated Parameters:
ADJUSTED_DOMAIN = PERCENT

Intention: Class describing a sun blind lifter and rotator. In addition to a the basic class of a three point actuator this class also contains two sensors recognizing whether the upper or lower end has been reached.

An actuator of this class can move up and down the blind as well can rotate the blind blades. To model this behavior several states are introduced.

Comment: Das Verhalten von dem Teil hier muß noch beschrieben werden!

- sowohl das die Sensoren die Blinds automaisch stoppen, wenn sie 1 werden
- was passiert, wenn man ganz unten (oben) ist und weiter in diese Richtung das Teil bewegt?
- wie sich das mit dem Rotieren verhält!!!!

DeveloperReq: Bisher haben wir hier keine Funktion, die den aktuellen Zustand des blind lifters repräsentiert. Brauchen wir so etwas? Hierfür wird weder eine Domäne in der BD eingeführt, noch wird darauf Bezug genommen.

DeveloperReq: Sollen wir Funktionen angeben, die genau beschreiben, wie sich der Zustand beim Liften bzw. Rotieren ändert in Abhängigkeit von der Zeit, in der auf den Knopf gedrückt wird? Das wird bisher nirgendwo erwähnt, wäre aber vielleicht ganz hilfreich, oder?

CustomerReq: In der Aktuatorenbeschreibung wird sowohl von forward/backward als auch von up/down als mögliche Zustände gesprochen. Wie ist denn der Zusammenhang zwischen diesen Größen?

Signature

Object *upperEndContact* : *BinarySensor*

Intention	:	End contact detecting whether the blind is lifted completely up or not.
Scope	:	eh
References	:	BD/Actuators
Comment	:	Wenn die beiden Endkontakte vom selben Typ sind, sollte hierfür eine eigene Klasse in der Datei BinarySensors angelegt werden; wenn nicht, werden eben zwei Klassen angelegt.
CustomerReq	:	Wie ist die Reaktionszeit?

Object $lowerEndContact : BinarySensor$

Intention	: End contact detecting whether the blind is lifted completely down or not.
Scope	: eh
References	: BD/Actuators
CustomerReq	: Wie ist die Reaktionszeit?

Timed Predicate $moveUp$

Intention	: This predicate is true iff the blind is moving up.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Timed Predicate $moveDown$

Intention	: This predicate is true iff the blind is moving down.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Timed Predicate $moveStop$

Intention	: This predicate is true iff the blind stops, i.e. it is neither moving down nor up.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Domain $DEGREE = \{0 \ldots 180\} < NAT$

Intention	: This domain models the possible values of the blind rotator, namely 0 to 180 degrees.
References	: BD/Actuators:blind lifter/rotator

Timed Function $rotateStatus() \rightarrow DEGREE$

Intention	: This function represents the current state of the blind rotator, i.e. the current position of the blind blades.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Timed Predicate $rotateIncrease$

Intention	: This predicate is true iff the blind blades are rotated in the direction of 0 degree.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Timed Predicate $rotateDecrease$

Intention	: This predicate is true iff the blind blades are rotated in the direction of 180 degree.
Scope	: eh
References	: BD/Actuators:blind lifter/rotator

Timed Predicate $rotateStop$

Intention	: This predicate is true iff the blind blades do not rotate in any direction. Note that this does not implies that the rotating angle of the blades is 0 or 180 degrees.

DomKnow

Property

Formal	: $\Box(moveUp \leftrightarrow (\neg moveDown \wedge \neg moveStop) \wedge moveDown \leftrightarrow (\neg moveUp \wedge \neg moveStop) \wedge moveStop \leftrightarrow (\neg moveUp \wedge \neg moveDown))$
NL	: The blind lifter is always in exactly one of the three possible modes moveUp, moveDown, moveStop.
CustomerReq	: Dies steht nirgendwo explizit. Gilt aber, oder?
CustomerReas	: Ergänzung Domänenwissen.

Property

Formal	: $\Box(rotateIncrease \leftrightarrow (\neg rotateDecrease \wedge \neg rotateStop) \wedge rotateDecrease \leftrightarrow (\neg rotateIncrease \wedge \neg rotateStop) \wedge rotateStop \leftrightarrow (\neg rotateIncrease \wedge \neg rotateDecrease))$
NL	: The blind rotator is always in exactly one of the three possible modes rotateIncrease, rotateDecrease, rotateStop.
CustomerReq	: Dies steht nirgendwo explizit. Gilt aber, oder?
CustomerReas	: Ergänzung Domänenwissen.

Property

Formal	: $\Box(moveDown \rightarrow rotateState = 0)$
NL	: If the blinds are moved down the rotating state, i.e. the angle of the blades have to be 0 degree.
References	: BD/Actuators:blind lifter/rotator

Property

Formal	: $\Box(moveUp \rightarrow rotateState = 180)$
NL	: If the blinds are moved up the rotating state, i.e. the angle of the blades have to be 180 degrees.
References	: BD/Actuators:blind lifter/rotator

Property

Formal	: $\Box(envStatus = stop \rightarrow moveStop \wedge rotateStop)$
NL	: Whenever the status of the sun blind is set to stop the blind is neither lifted in any direction nor are the blades rotated in any direction.
CustomerReq	: Steht zwar nirgends explizit, gilt aber, oder?
CustomerReas	: Ergänzung des Domänenwissens

Property

Formal	: $\Box(((envStatus = forward) \wedge (rotateStatus = 180) \wedge \neg upperEndContact) \rightarrow (moveUp \wedge rotateStop))$
NL	: Whenever the status of the sun blind is set to forward and the rotating state of the blades, i.e. the angle, is 180 degrees and the blind is not at the upper end, the blind is lifted up and the blades are not rotating.
References	: BD/Actuators:blind lifter/rotator

Property

Formal : $\Box(((envStatus = backward) \wedge (rotateStatus = 0) \wedge \neg lowerEndContact) \rightarrow (moveDown \wedge rotateStop))$

NL : Whenever the status of the sun blind is set to backward and the rotating state of the blades, i.e. the angle, is 0 degrees and the blind is not at the lower end, the blind is lifted down and the blades are not rotating.

References : BD/Actuators:blind lifter/rotator

Property

Formal : $\Box(((envStatus = forward) \wedge \neg(rotateStatus = 180)) \rightarrow (moveStop \wedge rotateIncrease))$

NL : Whenever the status of the sun blind is set to forward and the rotating state of the blades, i.e. the angle, is not 180 degrees the blind is not lifted up or down in any direction but rotated in the direction of 180 degrees.

References : BD/Actuators:blind lifter/rotator

Property

Formal : $\Box(((envStatus = backward) \wedge \neg(rotateStatus = 0)) \rightarrow (moveStop \wedge rotateDecrease))$

NL : Whenever the status of the sun blind is set to backward and the rotating state of the blades, i.e. the angle, is not 0 degree the blind is not lifted up or down in any direction but rotated in the direction of 0 degrees.

References : BD/Actuators:blind lifter/rotator

Property

NonFormal : If the rotating angle of the blind blades is x degrees light can enter from an angle of x degrees from vertical.

References : BD/Actuators:blind lifter/rotator

DeveloperReq : Kann das formalisiert werden bzw. bringt das Formalisieren etwas?

ReqSpec

Property

Formal : $\Box(\blacksquare_{\geq T}(adjustedStatus = forward \wedge lowerEndContact.measuredEntity) \rightarrow malfunction)$

NL : Whenever the machine has set the actuator to forward, i.e. to mowe up, for at least T time units, and the lowerEndContact constantly signals that the blind is still at the lower end, something is wrong.

Comment : Falls das so akzeptiert wird müssen die Größen noch eingeführt werden; insbesondere muß malfunction noch aufgeschlüsselt werden in die beiden möglichen Fehlerursachen und entsprechende Nachrichten müssen an den FM geschickt werden.

CustomerReq : Könnte so ein Plausibilitätscheck für diesen Aktuator aussehen? Als Unterstützung könnte das System in diesem Fall sagen, daß entweder der Aktuator oder der untere Sensor wahrscheinlich kaputt ist.

CustomerReas : Ergänzung der PD, Konkretisierung von PD/FMH7.

Description Class ValveAct

Derived from the following classes:

Description Class ThreePointActuator

Instantiated Parameters:

ADJUSTED_DOMAIN = PERCENT

Signature

Timed Function $settings() \rightarrow PERCENT$

Intention	: the current valve settings
References	: BD
DeveloperReq	: Ist es die Aufgabe eines Aktuators, Buch zu führen, an welcher Position er sich befindet? Ich würde sagen: Nein. (MK)

Description Class FMRoomTemperatureSettings

Intention: This class contains the default values concerning temperature the facility manager can make for the different kinds of rooms. It is assumed that the values can change over time. Therefore they are represented as objects of the class EnteredValue.

Comment: Ähnliche Vorgaben kann der Benutzer nicht machen (oder doch?); deshalb gibt es das hier nur für den FM.

Signature

Object $DT1Default : EnteredValue(STRING, DT1RANGE)$

NL	: default value of time DT1
Intention	: Default value of time DT1 set by the facility manager.
Scope	: eh
References	: PD/Par7
Comment	: Zeit DT1 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	: Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?
CustomerReas	: Unklarheit in der PD

Object $DT2Default : EnteredValue(STRING, DT2RANGE)$

NL	: default value of time DT2
Intention	: Default value of time DT2 set by the facility manager.
Scope	: eh
References	: PD/Par7
Comment	: Zeit DT2 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	: Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?
CustomerReas	: Unklarheit in der PD

Object $DT3Default : EnteredValue(STRING, DT3RANGE)$

NL	: default value of time DT3
Intention	: Default value of time DT3 set by the facility manager.
Scope	: eh
References	: PD/Par7
Comment	: Zeit DT3 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	: Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?
CustomerReas	: Unklarheit in der PD

Object *comfortTempDefault* : *EnteredValue*(*STRING*, *COMFORTTEMPRANGE*)

NL	:	default value of comfort temperature
Intention	:	Default value of comfort temperature set by the facility manager.
Scope	:	eh
References	:	PD/FMH3
Comment	:	Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	:	Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?

Object *nonFreezeTemperature* : *EnteredValue*(*STRING*, *TEMP*)

NL	:	non freeze temperature
Intention	:	temperature set by the facility manager that should be applied if the non freeze auto operating mode is active.
Scope	:	eh
Comment	:	Wie wird dieser Wert eingegeben. Daher noch eh als scope.
CustomerReq	:	Ist diese Temperatur variabel oder eine Konstante (in PD/FMH5 steht "preselected"). Wenn es mehrere Intervalle gitb, kann dann für jedes Intervall ein solcher Wert angegeben werden? Kann es für einzelne Räume andere Einstellungen dieses Wertes geben, bisher gehen wir davon aus, daß es einen für alle Räume gibt.
CustomerReas	:	Unklarheit in der PD

Object *workingTimeDefault* : *EnteredValue*(*STRING*, *WORKINGTIMERANGE*)

NL	:	default value of working time
Intention	:	Default value of working time set by the facility manager.
Scope	:	eh
References	:	PD/FMH3, PD/UH4
Comment	:	Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	:	Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?

Object *holidayTimeDefault* : *EnteredValue*(*STRING*, *HOLIDAYTIMERANGE*)

NL	:	default value of holiday time
Intention	:	Default value of holiday time set by the facility manager.
Scope	:	eh
References	:	PD/FMH3, PD/UH5
Comment	:	Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	:	Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant? Daher ist dieser Wert noch als eh eingestuft.

Object *tempComfortPre* : *TempInterval*

NL	:	temperature preselection interval
Intention	:	The preselected interval in which a user may enter the room comfort temperature
References	:	PD/UH1

CustomerReq: Sind alle möglichen Werte, die der FM und der Benutzer angeben können erfaßt?

ReqSpec

Property D1

Formal	:	$DT1Default.transferedValue = 60000$
NL	:	The initial default value of time DT1 is 1 minute.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD!

Property D2

Formal	:	$DT2Default.transferedValue = 600000$
NL	:	The default value of time DT2 is 10 minutes.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD!

Property D3

Formal	:	$DT3Default.transferedValue = 1800000$
NL	:	The default value of time DT3 is 30 minutes.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD!

Description Class FMRoomLightSettings

Intention: This class contains the default values concerning light the facility manager can make for the different kinds of rooms. It is assumed that the values can change over time. Therefore they are represented as objects of the class EnteredValue.

Comment: Noch zu ergänzen.

Signature

Object *lightOff* : *EnteredValue*

NL	:	facility manager light off
Intention	:	This object models whether the facility manager has turned all lights off in a certain room.
Scope	:	eh
References	:	PD/FM6
CustomerReq	:	Wielange bleibt das aktiv? Wie kann er das eingeben? Kann er alle Lampen auch anmachen? Wie ist es mit dem TaskLight, kann das überhaupt ausgeschaltet werden?

Description Class FacMan

DeveloperReq: Einige der als NonFormal angegebenen Eigenschaften werden wohl auch noch in anderen Klassen auftauchen.

Signature

Object *OfficeTempSetting* : *FMRoomTemperatureSettings*

Intention	:	Settings of the temperature relevant entities for all offices.

Object *CompLabTempSetting* : *FMRoomTemperatureSettings*

Intention	:	Settings of the temperature relevant entities for all computer labs.

Object *PeriRoomTempSetting* : *FMRoomTemperatureSettings*

Intention	:	Settings of the temperature relevant entities for all peripheral rooms.

Object *MeetRoomTempSetting* : *FMRoomTemperatureSettings*

Intention	:	Settings of the temperature relevant entities for all meating rooms.

Object *heatingPeriod* : *CalenderInterval*

Intention	:	representing the information of the heating period, i.e. start and end point and whether the actual date is within the heating period.
Scope	:	eh
References	:	PD/FMH4
CustomerReq	:	Wie werden die Werte dem System bekannt gemacht? Wie kann der Facility Manager sie eingeben? Kann er die Werte ändern? Daher ist dieses Objekt noch als eh eingestuft.

Object *nonHeatingPeriod* : *CalenderInterval*

Intention	:	representing the information of the non heating period, i.e. start and end point and whether the actual date is within the non heating period.
Scope	:	eh
References	:	PD/FMH4
CustomerReq	:	Wie werden die Werte dem System bekannt gemacht? Wie kann der Facility Manager sie eingeben? Kann er die Werte ändern? Daher ist dieses Objekt noch als eh eingestuft.

Object *buildingNonFreezePeriod* : *CalenderInterval*

Intention : representing the information of the period within which the non freeze mode is set for the whole building.
Scope : eh
References : PD/FMH5
CustomerReq : In PD/FMH5 wird der Plural verwendet. Ist es wirklich gewünscht, daß mehrere solche Intervalle eingegeben werden können. Wenn ja, wieviele kann es denn maximal geben, oder unbegrenzt? Wie kann der Facility Manager die Werte eingeben? Daher ist dieses Objekt noch als eh eingestuft.

Timed Predicate *adaptiveDelayTimes*
NL : =1 : delay times are adapted =0 : delay times are not adapted
Intention : indicates whether the facility manager has decided that the delay times for detecting occupancy have to be adapted by learning the typical user behavior or not.
Scope : eh
References : PD/FMH6
CustomerReq : Wie kann der FM diesen Wert einstellen? Deshalb ist er noch als eh gekennzeichnet. Kann der FM dies nur für alle Räume einheitlich vorgeben oder für jeden Raum getrennt. Zur Zeit wird von der ersten Variante ausgegangen.
CustomerReas : Unklarheit in der PD

Time Constant : *informDelay*
NL : information delay
Intention : delay between the time an event occurs about the facility manager should be informed and the time the facility manager is informed.
Scope : eh
References : PD/FMH8
Comment : neu eingeführt
CustomerReq : Wieviel Zeit darf verstreichen?
CustomerReas : Ergänzung zur PD

Timed Predicate *unreasonableBehavior*
NL : unreasonable behavior
Intention : is true if there is a repeated unreasonable user behavior
Scope : eh
References : PD/FMH8
CustomerReq : Was heißt das? Wie kann man das feststellen?
CustomerReas : Unklarheit in der PD

Timed Predicate *informed*
NL : facility manager informed
Intention : facility manager is informed
Scope : eh
References : PD/FM7, PD/FMH8
CustomerReq : Wie lange bleibt er das? Wie wird der FM informiert? Gibt es einen Unterschied zwischen verschiedenen Arten des Informiertwerdens, z.B. bzgl. Licht oder Temperatur?
CustomerReas : Unklarheit in der PD

ReqSpec

Property P1
NonFormal : Important informations should be sent to the facility manager. The facility manager is supported by the system in checking the reason before he goes to the location or take other actions.
References : PD/FMH7
Comment : Fragen über Fragen!
CustomerReq : Was sind "important informations"? Wie sollen sie verschickt werden? Per Hauspost? Per Rauchsignale? Wobei soll der FM unterstützt werden? Gründe von was???
CustomerReas : Unklarheit in der PD

Property P2
Formal : $\Box(unreasonableBehavior \Rightarrow_{\leq informDelay} informed)$
NL : The facility manager is informed about repeated unreasonable user behaviour, which contradicts aspects of energy saving.
References : PD/FMH8
PatternInst : delImp(unreasonableBehavior, informed, informDelay)
CustomerReq : Auf welche Art und Weise wird der FM informiert? Kann die Vorbedingung noch genauer spezifiziert werden, d.h. was ist unvernünftig, wie oft muß dies vorgefallen sein (was heißt also "repeated)? Soll das nach Benutzern oder Räumen unterschieden werden? Wenn ja, wie können verschiedene Benutzer erkannt werden? Wie lange soll der FM informiert bleiben?
CustomerReas : Unklarheit in der PD

Property P3
NonFormal : The control system provides basic energy saving procedures automatically for control comfort. If the facility manager deviates from such settings, the system provides a report on expected results.
References : PD/FMH9
Comment : Nix kapiert.
CustomerReq : Was ist hiermit gemeint?
CustomerReas : Unklarheit in der PD

Property P4

NonFormal	: The system warns against unreasonable inputs (input consistency).
References	: PD/FMH10
CustomerReq	: Auf welche Weise kann/soll die Maschine Warnungen veranlassen? Die unvernünftigen Eingaben werden doch bereits alle abgefangen und entsprechend korrigiert (siehe PD/FMH3). Bedeutet dies, daß in einem Korrekturfalle eine Warnungen gegeben werden soll? Bezieht sich das nur auf die Temperatursteuerung?
CustomerReas	: Unklarheit in der PD

Property P5

NonFormal	: The system provides reports on current and past energy consumption (cost reporting).
References	: PD/FMH11
CustomerReq	: Auf welche Art und Weise sollen die Berichte zur Verfügung gestellt werden? Was soll hier alles genau rein? Hier brauchen wir den Domänenexperten. Bezieht sich das nur auf die Temperatursteuerung?
CustomerReas	: Unklarheit in der PD

Property P6

NonFormal	: All malfunctions, alarms and unusual conditions are stored and reported on request.
References	: PD/FMH12
CustomerReq	: Was sind "malfunctions, alarms and unusual conditions"? Wie kann die Maschine sie feststellen? Wo werden sie gespeichert? Auf welche Art und Weise kann ein request erfolgen? Wo erfolgt dann die Ausgabe? Bezieht sich das nur auf die Temperatursteuerung?
CustomerReas	: Unklarheit in der PD

Property P7

NonFormal	: Malfunction of components that the system cannot detect, can be entered manually.
References	: PD/FMH13
CustomerReq	: Um welche malfunctions geht es konkret hier? Welche "components" werden betrachtet, Sensoren, Aktuatoren, Leitungen, User, ...? Was sind die gewünschten Reaktionen? Wie können die Fehlfunktionen wieder zurückgesetzt werden? Bezieht sich das nur auf die Temperatursteuerung?
CustomerReas	: Unklarheit in der PD

Description Class LightCtrlPanelFM

Intention: This class contains all the values concerning the control of the light in a room the facility manager can enter using the control panel. Here we consider the values given for *one single* room, not for all or a set of rooms.
Note that such a control panel does not exist up to now and has to be developed, too.

Signature

Object *lightOff* : *EnteredValue*(*BINARY*, *BINARY*)

Intention	: This object models whether the facility manager has turned all lights on or off in a certain room.
Scope	: eh
References	: PD/FM6, CustomerReviewResults
Comment	: Referenz genauer angeben. In der Einleitung auch CustomerReviewResults als referenziertes Dokument mit aufnehmen.
CustomerReq	:

1. Wie lautet hier der Initial-Wert. Es wäre günstig, wenn der für alle Räume gleich wäre.

2. Es wird bisher davon ausgegangen, daß der FM stets für jeden Raum einen Default-Wert (an,aus) vorgeben muß. Ist das in Ordnung so?

Description Class LightCtrlPanelRoom

Intention: This class contains all the values concerning the control of the light in a room that can be entered using the control panel that should be installed in a room. Entities, like domains, needed to define these values are specified in this class, too.
Note that such a control panel does not exist up to now and has to be developed, too.

Signature

Domain $CL\_SWITCH\_STATUS = \{0, 100, ambient\}$

NL	: values of the ceiling light switch of the control panel
Intention	: Set of possible values that can be set for the ceiling lights by using the control panel.
References	: PD/U14, PD/U16

Domain $AMB\_LIGHT\_LEV = \{0, 10 \ldots 100\}$

NL	: values of the setting of the ambient light level of the control panel
Intention	: Set of possible values that can be chosen for the ambient light level by using the control panel.
References	: PD/U14, PD/U16
CustomerReq	: Als mögliche Werte wurden hier die Werte angenommen, mit denen der Dimmer eingestellt werden kann. Ist dies ok?

Object $chosenLightScene : EnteredValues(LightScenes, LightScenes)$

NL	: control panel light scene
Intention	: light scene set by using the control panel
Scope	: eh
References	: PD/U6

Object $ambLightLev : EnteredValues(AMB\_LIGHT\_LEV, AMB\_LIGHT\_LEV)$

NL	: ambient light level of the control panel
Intention	: the value defined for the ambient light level by using the control panel
Scope	: eh
References	: PD/U14, PD/U16, PD/U7 ("The ambient light level setting is contained in the control panel.")

Object $winCeilLightSwitch : EnteredValue(CL\_SWITCH\_STATUS, CL\_SWITCH\_STATUS)$

NL	: window ceiling light switch of the control panel
Intention	: the value defined for the window ceiling light by using the control panel
Scope	: ev
References	: PD/U14, PD/U16
DeveloperReq	: Hier vielliecht das Suffix "Switch" streichen?

Object $wallCeilLightSwitch : EnteredValue(CL\_SWITCH\_STATUS, CL\_SWITCH\_STATUS)$

NL	: wall ceiling light switch of the control panel
Intention	: the value defined for the wall ceiling light by using the control panel
Scope	: ev
References	: PD/U14, PD/U16
DeveloperReq	: Hier vielliecht das Suffix "Switch" streichen?

Description Class LightCtrlPanel

Derived from the following classes:

Description Class LightCtrlPanelFM
Description Class LightCtrlPanelRoom

Intention: This class combines the two control classes LightCtrlPanelFM and LightCtrlPanelRoom resulting in a class that contains all entities concerning the control of the lights in a room that can be entered using a control panel. This combination is just a collection of the entities of the two mentioned classes.

Description Class TemperatureCtrlPanelFM

Intention: This class contains all the values concerning the control of the temperature in a room the facility manager can enter using the control panel and which can not be changed by a user entering a value at the control panel in a room. This kind of values is considered in the description class TemperatureCtrlPanel.
Here we consider the values given for *one single* room, not for all or a set of rooms.
Note that such a control panel does not exist up to now and has to be developed, too.

CustomerReq: Sind alle möglichen Werte, die der FM angeben kann und nicht vom Benutzer überschrieben werden können, erfaßt?

Signature

Object *nonFreezePeriod* : *CalenderInterval*

Intention	: representing the information of the period within which the non freeze mode is set for a room
Scope	: eh
References	: PD/FMH5
Comment	: Es können mehrere Intervalle angegeben werden. Das müssen wir noch berücksichtigen oder den Kunden davon überzeugen, daß es ausreichend ist, immer die nächste anzugeben.
CustomerReq	: In PD/FMH5 wird der Plural verwendet. Ist es wirklich gewünscht, daß mehrere solche Intervalle eingegeben werden können. Wenn ja, wieviele kann es denn maximal geben, oder unbegrenzt? Wie kann der Facility Manager die Werte eingeben? Daher ist dieses Objekt noch als eh eingestuft.

Timed Predicate *roomNonFreezeDefined*

NL	: a non-freeze period is defined
Intention	: The facility manager has defined a non-freeze period for a certain room.
References	: PD/FMH5

Object *nonFreezeTemperature* : *EnteredValue*(*STRING*, *TEMP*)

NL	: non freeze temperature
Intention	: temperature set by the facility manager that should be applied if the non freeze auto operating mode is active.
Scope	: eh
CustomerReq	: Ist diese Temperatur variabel oder eine Konstante (in PD/FMH5 steht "preselected"). Wenn es mehrere Intervalle gitb, kann dann für jedes Intervall ein solcher Wert angegeben werden? Kann es für einzelne Räume andere Einstellungen dieses Wertes geben, bisher gehen wir davon aus, daß es einen für alle Räume gibt.
CustomerReas	: Unklarheit in der PD

Object *standbyTemperature* : *EnteredValue*(*STRING*, *TEMP*)

NL	:	standby temperature
Intention	:	temperature set by the facility manager that should be applied if the standby operation mode is active.
Scope	:	eh
CustomerReq	:	War es gedacht, fuer den standby operation mode eine eigene Solltemperatur definieren zu koennen? Pro Raum?
CustomerReas	:	Information fehlt in der PD

Object *offTemperature* : *EnteredValue*(*STRING*, *TEMP*)

NL	:	off temperature
Intention	:	temperature set by the facility manager that should be applied if the off operating mode is active.
Scope	:	eh
CustomerReq	:	War es gedacht, fuer den off operation mode eine eigene Solltemperatur definieren zu koennen? Pro Raum?
CustomerReas	:	Information fehlt in der PD

Object *adaptive* : *EnteredValue*(*BINARY*, *BINARY*)

Intention	:	modus set by the facility manager whether the delay times used for determining occupancy are adaptive (i.e. the systems *learns* a typical user behavior) or not.
Scope	:	eh
References	:	PD/Par. 7, PD/FMH16
Comment	:	verschoben nach NoMalfunctionOccupancy

Description Class TemperatureCtrlPanelRoom

Intention: This class contains all the values concerning the control of the temperature in a room a user can enter using the control panel and for which the facility manager can not enter a defualt value. Values for which the facility manager specifies a default value that can be changed by a user are considered in the description class TemperatureCtrlPanel.
Here we consider the values given for *one single* room, not for all or a set of rooms.
Note that such a control panel does not exist up to now and has to be developed, too.

CustomerReq: Gibt es Werte, die der Benutzer bzgl. der Temperatursteuerung eingeben kann, für die es keinen Default-Wert vom FM gibt?

DeveloperReq: Falls es keine solchen Werte gibt, sollten wir die Klasse wieder löschen.

Signature

Timed Predicate *nonFreezeOverride*

NL	:	non freeze user override
Intention	:	The user has enterd an override over the non-freeze operation mode set by the facility manager.
References	:	PD/UH3

Description Class TemperatureCtrlPanel

Derived from the following classes:

Description Class TemperatureCtrlPanelFM
Description Class TemperatureCtrlPanelRoom

Intention: This class combines the two classes TemperatureCtrlPanelFM and TemperatureCtrlPanelRoom resulting in a class that contains all entities concerning the control of the temperature in a room that can be entered using a control panel. Besides the collection of the entities of the two mentioned classes in this class also those entities are introduced which represent values for which the facility manager can specify a default value that can be changed by a user.

Comment: Dies ist die passende Stelle, um die verschiedenen Modi genauer zu erläutern! Dies müssen wir noch machen.

Signature

Object *expectedUsageTimes* : *OccDelayTimes*

Intention	: This object contains for a room the set of occupancy delay times dt1, dt2, dt3 in the case that a usage of the room is expected.

Object *notExpectedUsageTimes* : *OccDelayTimes*

Intention	: This object contains for a room the set of occupancy delay times dt1, dt2, dt3 in the case that a usage of the room is not expected.

Object *comfortTemperature* : *ControlledInput*(*COMFORT_TEMP*)

NL	: user and facility manager comfort temperature settings
Intention	: This object contains the values relevant for determining the current comfort temperature of a room. It depends on the values entered by the facility manager and the user of a room.
Scope	: eh
References	: PD/FMH3
Comment	: Der Default-Wert des FM kann unterschiedlich für jeden Typ von Raum sein. Bei den Referenzen müssen auch noch die entsprechenden User-Needs angegeben werden.

Description Class CtrlPanel

Local Paramters:

NAT numSunBlinds

Derived from the following classes:

Description Class LightCtrlPanel
Description Class TemperatureCtrlPanel

Intention: This class contains all the values that can be entered for a room concerning the control of temperature, light, and sunblinds using a control panel. It is a collection of the values that can be entered by a user or the facility manager to control the lights and the temperature in a room. In addition to these entities in this class also the control of the sunblind using a control panel is considered.
It does not contain the values that can be entered concerning the malfunction of a sensor or actuator. They are introduced in the description classes dealing with the malfunctions of sensors and actuators.

The parameter stands for the number of sunblinds that belong to a room and which can be controlled using the control panel.

Signature

Object *sunBlindSwitch*[*numSunBlinds*] : *EnteredValue*(*THREE_POINT_ACTUATOR_RANGE*, *THREE_POINT_ACTUATOR_RANGE*)

NL	: the user setting of each sunblind
Intention	: Each object contains the user setting of a sun blind.
References	: PD/UH6
CustomerReq	: Der initiale Wert dürfte wohl stets STOP sein, oder?

ReqSpec

Property P_1

NonFormal	: The control panel should be easy and intuitive to use.
References	: PD/NF9

Description Class OfficeCtrlPanel

Derived from the following classes:
Description Class CtrlPanel

Intention: In addition to the values that can be entered for each kind of room using a control panel, this class also contains those entities which are specific for the room type *office*. This addition is a possibility to enter a value for the task light.

CustomerReq: Macht das noch Sinn, da das Task-Light nicht vom Kontrollsystem gesteuert werden kann?

Signature

Object *taskLightSwitch* : *EnteredValue*(*BINARY* , *BINARY*)

NL	: task light switch of the control panel
Intention	: The value defined for the task light by using the control panel.
Scope	: ev
References	: PD/U14

Description Class OccDelayTimes

Intention: Defines the values DT1, DT2, DT3 given by the user and the facility manager for a certain expected *usage profile* of a room.
DT1, DT2, and DT3 are ***D****elay* ***T****imes* that are used to determine whether a room is defined to be *occupied* or not. Note that *occupied* is not equal to the fact that a room is *not empty*, i.e. that a person is actually in a room. The entity *not empty* is measured directly using motion detectors. The entity *occupied* is a derived entity which depends on the entity *not empty* and three delay times DT1, DT2, and DT3. The precise definition is the property ???? given in description class ???. In the current description class *OccDelayTimes* only the general frame is introduced to deal with the three time delays in a common setting.

DeveloperReq: Was bedeutet hier *expected* usage profile?

Comment: Dies ist der richtige Platz, um die Zeiten dt1, dt2 und dt3 genauer zu erläutern.
Wo wird *usage profile* erläutert?

Signature

Object *dt1* : *ControlledInput*(*DT1RANGE*)

NL	: default value and range of time DT1
Intention	: Value of time delay DT1 determined w.r.t. a default value given by the facility manager and a value entered by a user.
Scope	: eh
References	: PD/Par7
Comment	: Zeit DT1 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein. Hier fehlen wohl noch einige Referenzen, nämlich die von den user-needs.

Object *dt2* : *ControlledInput*(*DT2RANGE*)

NL	: default value and range of time DT2
Intention	: Value of time delay DT2 determined w.r.t. a default value given by the facility manager and a value entered by a user.
Scope	: eh
References	: PD/Par7
Comment	: Zeit DT2 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein. Hier fehlen wohl noch einige Referenzen, nämlich die von den user-needs.

Object *dt3* : *ControlledInput*(*DT3RANGE*)

NL	:	default value and range of time DT3
Intention	:	Value of time delay DT3 determined w.r.t. a default value given by the facility manager and a value entered by a user.
Scope	:	eh
References	:	PD/Par7
Comment	:	Zeit DT3 noch genauer beschreiben (irgendwo). Dieser Wert kann unterschiedlich für jeden Typ von Raum sein. Hier fehlen wohl noch einige Referenzen, nämlich die von den user-needs.

ReqSpec

Property D1

Formal	:	$dt1.default.transferedValue = 60000$
NL	:	The initial default value of time DT1 is 1 minute.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD! Ist der Wert für alle betrachteten Benutzungs-Profile derselbe.

Property D2

Formal	:	$dt2.default.transferedValue = 600000$
NL	:	The default value of time DT2 is 10 minutes.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD! Ist der Wert für alle betrachteten Benutzungs-Profile derselbe.

Property D3

Formal	:	$dt3.default.transferedValue = 1800000$
NL	:	The default value of time DT3 is 30 minutes.
References	:	PD/Par7
Comment	:	Beachte, daß hiermit nur ein Startwert festgelegt wird! Beachte, die Basiseinheit ist ms.
CustomerReq	:	Ist das der korrekte Wert, siehe das Beispiel in der PD! Ist der Wert für alle betrachteten Benutzungs-Profile derselbe.

Description Class AdaptiveOccupancyTimes

Intention: This class provides all the entities and properties to define how the time delays are modified if the facility manager has chosen the delay times to be adaptive.

Comment: Diese Klasse ist noch nicht vollständig modelliert, weil noch einige Unklarheiten existieren.

Signature

Timed Function $dt1Adaptive$

Intention	:	Value for dt1 as it is determined using an adaptive learning function.
Scope	:	mh
References	:	PD
CustomerReq	:	Wie lautet der Initialwert hiervon?

Time Constant $T_{pastDT1}$

Intention	:	interval in the past considered for computing the value of time delay dt1
Scope	:	mv
CustomerReq	:	Wie groß ist dieser Wert? Kann er auch noch vom FM eingegeben werden, für jeden Raum?

Timed Function $dt2Adaptive$

Intention	:	Value for dt2 as it is determined using an adaptive learning function.
Scope	:	mh
References	:	PD
CustomerReq	:	Wie lautet der Initialwert hiervon?

Timed Function $dt3Adaptive$

Intention	:	Value for dt3 as it is determined using an adaptive learning function.
Scope	:	mh
References	:	PD
CustomerReq	:	Wie lautet der Initialwert hiervon?

Description Class NoMalfunctionOccupancy

Intention: This class provids all the entities and properties to define the term *occupancy* in the case that there is no mafunction.

Comment: Eine ähnliche Aufteilung zwischen Fehlerfall und nicht-Fehlerfall kann man wohl auch für die Außenhelligkeitssensoren vornehmen und später wohl als eigenständiges Muster in einer Klasse zusammenfassen. Auch im Hinblick auf einen Katalog recht interessant.

Signature

Parameter

Timed Function *dt1*

Intention	: current value used for time delay DT1 to determine occupancy.
Scope	: mh
Comment	: Dieser Wert muß entsprechend gesetzt werden, wenn die Klasse verwendet wird.

Timed Function *dt2*

Intention	: current value used for time delay DT2 to determine occupancy.
Scope	: mh
Comment	: Dieser Wert muß entsprechend gesetzt werden, wenn die Klasse verwendet wird.

Timed Function *dt3*

Intention	: current value used for time delay DT3 to determine occupancy.
Scope	: mh
Comment	: Dieser Wert muß entsprechend gesetzt werden, wenn die Klasse verwendet wird.

End of Parameter

Timed Predicate *emptyEnv*

NL	: =1 a person is in the room =0 no person is in the room
Intention	: Indicates whether a person is in a room or not. This entity represents the situation as it is in the real world and not how it is seen by the machine.
Scope	: eh
References	: PD/Paragraph7

Timed Predicate *occupiedEnv*

NL	: =1 room occupied =0 room not occupied
Intention	: Indicates whether a room is said to be occupied or not. Note that this predicate does not describe whether a person is in a room or not. See Timed Predicate emptyEnv. This entity represents the situation as it is in the real world and not how it is seen by the machine.
Scope	: eh
References	: PD/Paragraph4, PD/Paragraph7

Timed Predicate *emptyMach*

NL	: =1 a person is in the room =0 no person is in the room
Intention	: Indicates whether a person is in a room or not. This entity represents the situation as it is seen by the machine and not how it is in the real world.
Scope	: mh
References	: PD/Paragraph7
Comment	: Diese Größe muß in der übergeordneten Klasse, in der die Motion detecors aggregiert werden, noch entsprechend definiert werden.
DeveloperReq	: Ist das hier auch ein Parameter?
CustomerReq	: Ist diese Größe nur über die Motion Detectors definiert? Gehört eigentlich als Frage zu der Definition in der übergeordneten Klasse!

Timed Predicate *occupiedMach*

NL	: =1 room occupied =0 room not occupied
Intention	: Indicates whether a room is said to be occupied or not. Note that this predicate does not describe whether a person is in a room or not. See Timed Predicate emptyMach. This entity represents the situation as it is seen by the machine and not how it is in the real world.
Scope	: mh
References	: PD/Paragraph4, PD/Paragraph7

ReqSpec

Property

Formal : $\Box(occupiedEnv \leftrightarrow \blacksquare_{\leq dt1} \neg emptyEnv \vee < *[\blacksquare_{\leq dt1+dt2} \neg emptyEnv] \Rightarrow > \blacksquare \blacklozenge_{\leq dt3} \neg emptyEnv)$

NL : This is a *requirement* using the entities existing in the real world. A room is defined to be occupied if and only if one of the following conditions holds:

- during the past dt1 time units a person has continuously been in the room
- there has already been a time span of length DT1+DT2 such that a person has continuously been in the room, and since this has been the case for the last time, periods where the room has continuously been empty were not longer than or equal to DT3.

PatternInst : LazyReaction(occupiedEnv, !emptyEnv, DT1, DT2, DT3)

DeveloperReq : Ist das Domänenwissen oder eine Anforderung?

CustomerReq : Gilt diese Definition von roomOccupied nur für die Temperatursteuerung oder auch für andere Steuerungen (Licht)?

Property

Formal : $\Box(occupiedMach \leftrightarrow \blacksquare_{\leq DT1}\neg emptyMach \vee <*[\blacksquare_{\leq DT1+DT2}\neg emptyMach] \Rightarrow> \blacksquare\blacklozenge_{\leq DT3}\neg emptyMach)$

NL : This is a *specification* using only entities visible for the machine. A room is defined to be occupied if and only if one of the following conditions holds:

- during the past dt1 time units a person has continuously been in the room
- there has already been a time span of length DT1+DT2 such that a person has continuously been in the room, and since this has been the case for the last time, periods where the room has continuously been empty were not longer than or equal to DT3.

PatternInst : LazyReaction(occupiedMach, !emptyMach, DT1, DT2, DT3)

CustomerReq : Gilt diese Definition von roomOccupied nur für die Temperatursteuerung oder auch für andere Steuerungen (Licht)?

Description Class MalfunctionOccupancy

Signature

Object *manualEnteredOcc : EnteredValue(BINARY)*

Intention	: Manually entered value concerning occupancy entered by a user in the case of a malfunction of the detectors determining occupancy.
Scope	: eh
Comment	: Es muß evtl. noch berücksichtigt werden, daß die Eingabe nur im Fehlerfall möglich ist.
CustomerReq	: Wann kann diese Eingabe erfolgen? In PD/NFH4 steht, daß das nur im Fehlerfalle der Bewegungsmelder erfolgen kann.

Timed Predicate *stateManualEnteredOcc*

Intention	: This predicate is true iff occupancy has been entered manually.
Scope	: mh
References	: PD/NFH4
Comment	: Ist vom Kunden eingeführt worden. Es muß noch gemäß den Antworten des Kunden definiert werden, wie lange dieses Prädikat true ist. Z.B. ändert es seinen Wert, wenn von Tages- auf Nachtzeit gewechselt wird.

Predicate *defaultOccDaytime*

Intention	: Default value for occupancy during daytime in case of a malfunction of the sensors determining occupancy. Note that this value can not be changed.
Scope	: mh
References	: PD/NFH4
Comment	: Ist vom Kunden eingeführt wurden.
CustomerReq	: Ist das nicht immer der default value, wenn keine Anwesenheit erwartet wird?

Predicate *defaultOccNighttime*

Intention	: Default value for occupancy during nighttime in case of a malfunction of the sensors determining occupancy. Note that this value can not be changed.
Scope	: mh
References	: PD/NFH4
Comment	: Ist vom Kunden eingeführt wurden.
CustomerReq	: Ist das nicht immer der default value, wenn keine Anwesenheit erwartet wird?

Timed Predicate *occMal*

Intention	: Indicates the status of occupancy in the case of a malfunction of the sensors determining occupancy.
Scope	: mh

ReqSpec

Initialization

Property

Formal : $\neg stateManualEnteredOcc$
NL : Initially no value for occupancy in the case of a malfuntion is entered manullay.

End of Initialization

Property

Formal : $\Box(defaultOccDaytime)$
NL : During day time the default status of occupancy is *occupied*.
References : PD/NFH4

Property

Formal : $\Box(\neg defaultOccNighttime)$
NL : During night time the default status of occupancy is *not occupied*.
References : PD/NFH4

Property

Formal : $\Box(stateManualEnteredOcc \rightarrow (occMal \leftrightarrow manualEnteredOcc.transfered = 1) \wedge (\neg stateManualEnteredOcc \rightarrow (daytime \rightarrow (occMal \leftrightarrow defaultOccDaytime)) \wedge (\neg daytime \rightarrow (occMal \leftrightarrow defaultOccNighttime))))$
NL : If a value for the status of occupancy has been entered manually this is the value used for occupancy in the case of a malfunction of the sensors determining occupancy. Otherwise, the value used for occupancy in the case of a malfunction of the sensors determining occupancy depends on whether it is daytime or nighttime. In this case the corresponding default values are used.
References : PD/NFH4, CustomerReviewResults?

Description Class Occupancy

Derived from the following classes:
Description Class NonMalfunctionOccupancy
Description Class MalfunctionOccupancy

CustomerReq: Gelten die Sachen bzgl. Anwesenheit nur für die Räume oder auch für die Flure. Wenn ja, kommt das alles hier nach Area.

Signature

Timed Predicate *malOccDetEnv*
Intention : true iff the sensors determining occupancy have a malfunction Note that this predicate represents the value as it exists in the real world.
Scope : eh
CustomerReq : In PD/NFH4 ist die Rede von "serious malfunction". Gibt es eine graduelle Abstufung zwischen dem Fehlverhalten eines Sensors. Wenn ja, welche Auswirkungen hat das? Ebenso ist dort von mehreren Detektoren die Rede, sichtbar ist aber doch nur einer, oder? Bezieht sich PD/NFH4 nur auf Räume oder auch auf den Flur?

Timed Predicate *malOccDetMach*
Intention : true iff the sensors determining occupancy have a malfunction Note that this predicate represents the value as it seen by the machine.
Scope : mh
CustomerReq : In PD/NFH4 ist die Rede von "serious malfunction". Gibt es eine graduelle Abstufung zwischen dem Fehlverhalten eines Sensors. Wenn ja, welche Auswirkungen hat das? Ebenso ist dort von mehreren Detektoren die Rede, sichtbar ist aber doch nur einer, oder? Bezieht sich PD/NFH4 nur auf Räume oder auch auf den Flur?

Timed Predicate *stateManualEnteredMal*
Intention : This predicate is true iff it has been manually entered whether there is a malfunction of the sensors determining occupancy or whether there is no malfunction of them.
Scope : mh
References : CustomerReviewResults, PD?

Object $manualEnteredMal : EnteredValue(BINARY)$
Intention : Manually entered value concerning a malfunction of the sensors determining occupancy entered by a user. 1 stands for a malfunction, 0 stands for no malfunction.
Scope : eh

Timed Predicate *automaticDetMal*

Intention	: This predicate is true iff a malfunction of the sensors determining occupancy has been detected automatically, i.e. not by entering it manually.
Scope	: mh
CustomerReq	: Hier wird noch eine geeignete Definition vom Domänenexperten benötigt.

Timed Predicate *malfunction*

Intention	: This predicate is true iff the machine assumes that there is a malfunction of the detectors determining occupancy.
Scope	: mh
Comment	: Diese Größe ist zur abkürzenden Schreibweise und zur Strukturierung eingeführt worden. Sie kann gemäß der Definition unten expandiert werden.

Timed Predicate *occUsed*

Intention	: represents the value used by the machine concerning occupancy
Scope	: mh
References	: PD/NFH4
Comment	: Ist vom Kunden eingeführt worden.
DeveloperReq	: Ist das noetig? Kann man nicht immer roomOccupied verwenden und auch fuer malfunctions definieren?

ReqSpec

Initialization

Property

Formal : $\neg stateManualEnteredMal$

NL : Initially it has not been entered manually whether there is a malfunction of the sensors determining occupancy or whether there is no malfunction of them

CustomerReq : Das ist ja wohl ok so!?

End of Initialization

Property

Formal : $\Box(malfunction \leftrightarrow (stateManualEnteredMal \wedge manualEnteredMal) \vee (\neg stateManualEnteredMal \wedge automaticDetMal))$

NL : The machine assumes a malfunction of the sensors determining occupancy iff one of the follwoing conditions is fulfilled:

- a malfunction has been entered manually
- it has not been entered manually that there is no malfunction and a malfunction has been detected automatically

CustomerReq : Ist das die korrekte Priorissierung: Manuell über automatisch?

Property

Formal : $\Box(occUsed \leftrightarrow (malfunction \wedge occMal) \vee (\neg malfunction \wedge occupiedMach))$

NL : The machine assumes occupancy if and only if one of the following condition is fulfilled:

- the machine assumes a malfunction and the predicate representing the value concerning occupancy assumed in this case is true,
- the machine assumes no malfunction and the predicate representing the value concerning occupancy assumed in this case is true.

Comment : Hier findet die Komposition der beiden Modi statt!

DeveloperReq : Übersetzung ist sicherlich noch nicht so richtig toll.

Description Class ExpectedUsage

Intention: Contains the settings of the facility manager and the user concerning the expected usage of a room.

Signature

Object $workingTime : ControlledInput(WORKINGTIMERANGE)$

NL	: default value and preselection range of working time
Intention	: Default value of working time set by the facility manager.
Scope	: eh
References	: PD/FMH3, PD/UH4
Comment	: Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	: Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant?

Object $holidayTime : ControlledInput(HOLIDAYTIMERANGE)$

NL	: default value and preselection range of holiday time
Intention	: Default value of holiday time set by the facility manager.
Scope	: eh
References	: PD/FMH3, PD/UH5
Comment	: Dieser Wert kann unterschiedlich für jeden Typ von Raum sein.
CustomerReq	: Wie wird dieser Wert dem System bekannt gemacht? Wie kann der Facility Manager den Wert eingeben? Kann er den Wert ändern oder ist er konstant? Daher ist dieser Wert noch als eh eingestuft.

Timed Predicate $expectedUsage$

NL	: The expected usage
Intention	: The expected usage of a room is characterized due to working and holiday time settings of the room user and facility manager.

ReqSpec

Property D2

Formal	: $\Box(expectedUsage \leftrightarrow now \in workingTime.value \wedge now \notin holidayTime.value)$

Description Class RoomOperationMode

Intention: Defines the notion 'operation mode' for rooms

Signature

Parameter

Timed Predicate $expectedUsage$

Intention	: The expected usage of a room is characterized due to working and holiday time settings of the room user and facility manager.

Timed Predicate $roomNonFreezeDefined$

Intention	: The facility manager has defined a non-freeze period for a certain room.

Object $nonFreezePeriod : CalenderInterval$

Intention	: representing the information of the period within which the non freeze mode is set for a room
References	: TemperatureCtrlPanelFM::nonFreezePeriod

Object $buildingNonFreezePeriod : CalenderInterval$

Intention	: representing the information of the period within which the non freeze mode is set for the whole building.
References	: FacMan::buildingNonFreezePeriod

Timed Predicate $nonFreezeOverride$

Intention	: The user has enterd an override over the non-freeze operation mode set by the facility manager.
References	: TemperatureCtrlPanelRoom::nonFreezeOverride

End Parameter

Timed Predicate $nonFreezeTime$

NL	: non-freeze operation time
Intention	:

Timed Predicate $active$

NL	: this operation mode is active
Intention	: this operation mode is active
Scope	: eh
References	: PD/Paragraph4

Timed Function $desiredTemperature() \rightarrow TEMP$

NL	: desired temperature
Intention	: temperature that should be established in the room in this operation mode
Scope	: mh
References	: PD/FMH5
Comment	: Diese Größe wird explizit vom Kunden erwähnt.

ReqSpec

Property P11

Formal	:	$\Box(nonFreezeTime \leftrightarrow \neg nonFreezeOverride \wedge ((roomNonFreezeDefined \wedge nonFreezePeriod.actual) \vee (\neg roomNonFreezeDefined \wedge buildingNonFreezePeriod.actual))$
NL	:	Non-freeze operation time is set, exactly if 1) there is no user override and 2) the room specific non-freeze time is defined and currently valid or the room specific non-freeze time is not set and the building wide non-freeze time is currently valid.
References	:	PD/FMH5

Property P_15

NonFormal	:	The desired temperature shall be reached as fast as possible during the heating up state.
References	:	PD/UH2
DeveloperReq	:	What is "as good as possible"? From the user's point of view it is not interesting, when or how the room is heated up. It is only important that the room has comfort temperature during comfort mode. -¿ CustReq

Property P_16

NonFormal	:	The desired temperature ... is maintained during the hold temperature state as good as possible.
References	:	PD/UH2
DeveloperReq	:	- What is "as good as possible"? Or should the temperature stay within a certain interval around the comfort temperature. -¿ CustReq - The conflict user needs (UCs) must be formulated, but until now there is no obvious conflict ...

Description Class NonFreezeMode

Derived from the following classes:
Description Class RoomOperationMode

Intention: Defines the operation mode 'NonFreeze' for rooms

Signature

Parameter

Object $nonFreezeTemperature : EnteredValue(STRING, TEMP)$

Intention	:	temperature set by the facility manager that should be applied if the non freeze auto operating mode is active. (room specific)
References	:	TemperatureCtrlPanelFM::nonFreezeTemperature

ReqSpec

Property P11

Formal	:	$\Box(active \leftrightarrow nonFreezeTime)$
NL	:	Non-freeze mode and non-freeze time are synonyms.
References	:	PD/FMH5

Property P11

Formal	:	$\Box(desiredTemperature = nonFreezeTemperature.transfered)$
NL	:	The desired room temperature in non-freeze auto mode is the non-freeze temperature set by the facility manager.
References	:	PD/FMH5

Description Class ComfortMode

Derived from the following classes:

Description Class RoomOperationMode

Intention: Defines the operation mode 'Comfort' for rooms

Signature

Parameter

Object $comfortTemperature : EnteredValue(STRING, TEMP)$

Intention	: temperature set by the facility manager or user that should be applied if the comfort operating mode is active. (room specific)
References	: TemperatureCtrlPanel::comfortTemperature

ReqSpec

Property P

Formal	: $\Box(active \leftrightarrow roomOccupied \wedge \neg nonFreezeTime)$
NL	: The room is in comfort mode if the room is occupied ...
References	: PD/Paragraph4
DeveloperReq	: Ist das Domänenwissen oder eine Anforderung?

Property P11

Formal	: $\Box(desiredTemperature = comfortTemperature.value)$
NL	: The desired room temperature in off mode is the comfort temperature set by the facility manager or the user.
References	: PD/FMH5

Description Class StandbyMode

Derived from the following classes:

Description Class RoomOperationMode

Intention: Defines the operation mode 'Standby' for rooms

Signature

Parameter

Object $standbyTemperature : EnteredValue(STRING, TEMP)$

Intention	: temperature set by the facility manager that should be applied if the standby operating mode is active. (room specific)
References	: TemperatureCtrlPanelFM::standbyTemperature

ReqSpec

Property P2

Formal	: $\Box(active \leftrightarrow \neg nonFreezeTime \wedge usageExpected \wedge \neg roomOccupied)$
NL	: If occupancy is expected and the room is not occupied then the operating mode is standby time.
References	: PD/Paragraph4
DeveloperReq	: Ist das Domänenwissen oder eine Anforderung?

Property P11

Formal	: $\Box(desiredTemperature = standbyTemperature.value)$
NL	: The desired room temperature in standby mode is the standby mode temperature set by the facility manager.
References	: PD/FMH5

Description Class OffMode

Derived from the following classes:
Description Class RoomOperationMode

Intention: Defines the operation mode 'Off' for rooms

Signature

Parameter

Object *offTemperature* : *EnteredValue*(*STRING*, *TEMP*)

Intention	: temperature set by the facility manager that should be applied if the off operating mode is active. (room specific)
References	: TemperatureCtrlPanelFM::offTemperature

ReqSpec

Property P1

Formal	: $\square(active \leftrightarrow \neg nonFreezeTime \wedge \neg expectedUsage \wedge \neg roomOccupied)$
NL	: If no occupancy is expected and the room is not occupied then the operating mode is off time.
References	: PD/Paragraph4
DeveloperReq	: Ist das Domänenwissen oder eine Anforderung?
CustomerReq	: Was passiert wenn !roomOccupied und !expectedUsage gilt? Hier wurde entschieden, daß dann roomOccupied Priorität hat.
CustomerReas	: DomKnow-Inconsistence

Property P11

Formal	: $\square(desiredTemperature = offTemperature.value)$
NL	: The desired room temperature in off mode is the off mode temperature set by the facility manager.
References	: PD/FMH5

Description Class Radiator

Intention: This class contains all the entities belonging to a radiator.

Signature

Object *waterInTemp* : *WaterTempSensor*

Intention	: sensor to measure the input water temperature
References	: BD/Rooms

Object *waterOutTemp* : *WaterTempSensor*

Intention	: sensor to measure the output water temperature
References	: BD/Rooms

Object *meanTemp* : *WaterTempSensor*

Intention	: sensor to measure the mean radiator temperature
References	: BD/Rooms
CustomerReq	: Was bedeutet hier "mean"?
CustomerReas	: Aus Interesse. (MK)

Object *valveAct* : *ValveAct*

Intention	: This object represents the radiator control valve.
References	: BD/Rooms

Description Class WindowSection

Intention: This class contains all the entities belonging to each window section. This is a radiator and a opener for the upper sash. This is the usual window sections of rooms that are situated on the northern site of a building.

Signature

Object *rad* : *Radiator*

Intention	: an radiator of a window section
References	: BD/Rooms

Object *uso* : *UpperSashOpener*

Intention	: an upper sash opener of a window section
References	: BD/Rooms

Description Class BlindedWindowSection

Derived from the following classes:
Description Class WindowSection

Intention: In addition to a "normal" window section this class contains also a rotator for the sun blind. This is the usual window sections of rooms that are not situated on the northern site of a building.

Signature

Object *sb* : *SunBlind*

Intention	: a combined blind lifter/rotator
References	: BD/Rooms

Description Class DimmableLight

Intention: This Class contains all the entities concerning a dimmable light. These are w.r.t. the BD document: pulse, status line, control system is active, dim value.

Signature

Object *statusLine* : *Light*

Intention : represents the status line of a dimmable light
Scope : eh
References : BD/Offices, BD/Actuators:Figure, BD/Sensors:dimmable light(sll)

Object *pulse* : *Pulse*

Intention : represents the pulse line of a dimmable light
Scope : eh
References : BD/Actuators:Figure, BD/Sensors:relais
Comment : Domänenexperte hat angegeben, daß es sich bei dem als relais gekennzeichneten Sensor um den hier gemeinten Aktuator handelt!

Object *conSysActive* : *ControlSystemActive*

Intention : represents the signal "control system is active" of a dimmable light
Scope : eh
References : BD/Actuators:Figure, BD/Sensors:dimmable light(cia)

Object *dimValue* : *Dimmer*

Intention : represents the actuator dim value that can be used by the machine to set the current dim value.
Scope : eh
References : BD/Actuators:Figure, BD/Sensors:dimmable light(dimmer)

Time Constant *safeModeReactDelay*

Intention : delay between a missing signal "control system is active" and the appropriate recation, i.e. changing to fail safe mode.
Scope : sh
References : BD/Actuators:above the figure
Comment : Ergänzung zur PD

Timed Predicate *safeMode*

Intention : true iff the dimmable light is in fail safe mode
Scope : sh
References : BD/Actuators:above the figure
Comment : Diese Größe wird explizit vom Kunden eingeführt.
CustomerReq : Welche Konsequenzen sieht dieser Zustand nach sich? Sowohl wenn er gilt als auch wenn er nicht gilt.

DomKnow

Property

Formal : $\Box(\blacksquare_{>60000} \neg conSysActive \Rightarrow_{\leq safeModeReactTime} safeMode)$
References : BD/Actuators:above the figure
Comment : Beachte die Basiseinheit ist ms.
CustomerReq : In der BD steht, daß das Signal genau alle 60 Sekunden gesendet werden muß. Wir gehen davon aus, daß das nicht realisiert werden kann, sondern daß es ausreichend ist, daß innerhalb der letzten 60 Sekunden dieses Signal einmal auf eins gewesen sein muß. Wie wird der Modus denn wieder verlassen? Z.B. wenn conSysActive wieder gesendet wird?

Property

Formal : $\Box(safeMode \rightarrow dimVal.envEntity = 100)$
NL : Whenever the dimmable light is in fail safe mode the dim value is 100 percent.
References : BD/Actuators:above the figure
CustomerReq : Ist das korrekt, daß dies immer der Fall ist im safeMode? Hat dieser Modus noch weitere Konsequenze?

Description Class LightButtonSet

Local Paramters:
Nat numButtons

Intention: Combines a number of push buttons with a light

CustomerReq: Hier muß geklärt werden, wie das Zusammenspiel zwischen den Push-Buttons und dem Licht ist. Handelt es sich um Domänenwissen, was hier angegeben wird oder um Anforderungen. Deswegen ist dies noch nicht überarbeitet.

Signature

Object *light* : *Light*

Intention	: The light that is controlled by the push buttons.
Scope	: eh

Timed Predicate *pushButton*

NL	: any button is pushed
Intention	: true iff any button is pushed
Scope	: eh

Time Constant T_1

Intention	: The delay between pushing a button and the change of the light. Here the light existing in the real world is considered.
Scope	: eh

Time Constant T_2

Intention	: The delay between pushing a button and the change of the status line of the light. Here the signal sent to the machine and not the light existing in the real world is considered.
Scope	: eh

DomKnow

Property

Formal	: $\Box(T_2 > T_1)$
NL	: The time constant T_2 is always greater than the time constant T_1.

Property

Formal	: $\Box(light.envEntity = 0 \wedge [pushBut] \rightarrow \Diamond_{\leq T_1} light.envEntity = 1)$
NL	: The light (existing in the real world) always changes from off to on within time T_1 if any pushbutton is pushed.
Comment	: Hierbei handelt es sich um reines Domänenwissen, von dem die Maschine nichts direkt sieht. Die entsprechende Aussage, die auch für die Maschine relevant ist, ist weiter unten angegeben.

Property

Formal	: $\Box(light.envEntity = 1 \wedge [pushBut] \rightarrow \Diamond_{\leq T_1} light.envEntity = 0)$
NL	: The light (existing in the real world) always changes from on to off within time T_1 if any pushbutton is pushed.
Comment	: Hierbei handelt es sich um reines Domänenwissen, von dem die Maschine nichts direkt sieht. Die entsprechende Aussage, die auch für die Maschine relevant ist, ist weiter unten angegeben.

Property D_2

Formal	: $\Box(light.measuredEntity = 0 \wedge [pushBut] \rightarrow \Diamond_{\leq T_2} light.measuredEntity = 1)$
NL	: The value of the status line (which is the measured entity of the light) always changes from off to on within time T_2 if any pushbutton is pushed.
References	: BD/Rooms, BD/Offices, BD/Hallway

Property D_3

Formal	: $\Box(light.measuredEntity = 1 \wedge [pushBut] \rightarrow \Diamond_{\leq T_2} light.measuredEntity = 0)$
NL	: The value of the status line (which is the measured entity of the light) always changes from on to off within time T_2 if any pushbutton is pushed.
References	: BD/Rooms, BD/Offices, BD/Hallway

Description Class DimmableLightButtonSet

Derived from the following classes:
Description Class LightButtonSet

Intention: Extends the LightButtonSet for DimmableLight

Signature

Object *LightButtonSet* :: *light* : *DimmableLight*

Intention	: The dimmable light that is controlled by the push buttons.

DomKnow

Property D_1

Formal	: $\Box(\forall d : light.impRel = 0 \wedge [pushBut] \wedge light.dim = d \rightarrow \Diamond_{\leq LightButtonSet::T\_1} light.status = d)$
NL	: Whenever the impulse relais is off and any push button is pushed the ceiling light is dimmed within the reaction time of the push button to the value given by the dimmer.
Comment	: Das muß noch überarbeitet werden.
CustomerReq	: Ist das Domänenwissen oder eine Anforderung?

Property D_2

Formal	: $\Box(light.impRel = 1 \wedge [pushBut] \Diamond_{\leq LightButtonSet::T\_1} light.status = 0)$
NL	: Whenever the impulse relais is on and the push button is pushed the ceiling light is off within the reaction time of the push button.
CustomerReq	: Ist das Domänenwissen oder eine Anforderung?

Description Class Door

CustomerReq: In der BD wird im Fließtext stets gesagt (bei Rooms and Hallway), daß eine Tür nur einen closed contact enthalt. Bei der Beschreibung der Sensoren tauchen aber noch weitere Tür-Sensoren auf. Vermutlich gehören die auch noch dazu, oder?

CustomerReq: Sind die Türen der Räume und des Flures gleich ausgestattet? Im Fließtext ist immer nur von dem door closed contact die Rede, bei der Sensorenbeschreibung sind aber die folgenden drei Objekte angegeben. Wie ist es nun tatsächlich?

Signature

Object *doorClosedContact* : *DoorContact*

Intention	: representing the closed contact of a door. It is true iff if the door is completely closed.
Scope	: eh
References	: BD/Sensors

Object *doorOpenContact* : *DoorContact*

Intention	: representing the open contact of a door. It is true iff if the door is open for at least 30 degree.
Scope	: eh
References	: BD/Sensors

Object *doorLockContact* : *DoorContact*

Intention	: representing the lock contact of a door. It is true iff if the door is locked by a key.
Scope	: eh
References	: BD/Sensors

Description Class Area

Intention: Some general considerations on areas with light; possible areas are currently rooms and hallways.

CustomerReq: Muß man bei Licht und Temperatur aus Sicht der Maschine unterschiedliche Occupancy-Zustände betrachten?

CustomerResp: Ja!

Signature

Timed Predicate: *occupied*

NL	:	area occupied
Intention	:	the area is occupied by a person
Scope	:	eh
References	:	PD/U1, PD/U2, PD/U3, PD/U4, PD/FM3, PD/FM6
DeveloperReq	:	Klarer machen, daß damit die Größe aus der realen Welt gemeint ist.

Timed Predicate: *allLightsOff*

NL	:	all ceiling lights off
Intention	:	all ceiling lights in an area are off
Scope	:	eh
References	:	PD/FM1, PD/FM3, PD/FM6
DeveloperReq	:	Klarer machen, daß damit die Größe aus der realen Welt gemeint ist.

Timed Predicate: *safeLight*

NL	:	safe light
Intention	:	light is sufficient to move safely
Scope	:	eh
References	:	PD/U17

Time Constant *T_1*

Intention	:	The time delay between occupation of an area and the establishment of safe light.

Timed Function $T\_2() - > Time$

Intention	:	The time delay between a sufficiently long non-occupation of an area and switching off the light

ReqSpec

Property P_1

Formal	:	$\Box(occupied \Rightarrow_{\leq T\_1} safeLight)$
NL	:	Whenever the area is occupied for at least time T_1, the light is sufficient to move safely within this time span and remains sufficient to move safely at least as long as the precondition is true.
References	:	PD/U1, PD/U17
PatternInst	:	DelayedImplication (occupied, safeLight, T_1)

Property P_2

NonFormal	:	Use daylight to achieve the desired light whenever possible.
References	:	PD/FM1
Comment	:	Formalization waits for a refined version.

Property P_3

Formal	:	$\Box(\neg occupied \Rightarrow_{\leq T\_2} allLightsOff)$
NL	:	Whenever the area is continuously not occupied for at least T_2 minutes, then all lights are off within this time span and remain off at least as long as the area is not occupied.
References	:	PD/FM2, PD/FM3, PD/FM6
PatternInst	:	DelayedImplication (occupied, allLightsOff, T_2)

Property P_4

NonFormal	:	The value T_2 can be set by the facility manager for each area separately.
References	:	PD/FM4, PD/FM5
CustomerReq	:	Wo kann der Wert gesetzt werden? Kann der Wert wirklich für jede Area oder für jeden Raumtyp separat gesetzt werden?
CustomerResp	:	Eingabe übers Ctrl Panel. Eigener Wert für jede Area.

Description Class RoomMalfunctions

Comment: Erstmal nur hier rein geschrieben, damit es irgendwo steht. Hier muß noch eine genaue Zuordnung erfolgen.

Signature

Timed Predicate $malfunctionAllTempSens$

NL	: all temperature sensors of a room doe not work correctly
Intention	: is true iff all temperature sensors of a room does not work correctly.
Scope	: eh
References	: PD/NFH3
CustomerReq	: Wie kann das festgestellt werden? Deshalb: eh. Wie kann dieses Prädikat wieder zurückgesetzt werden?

Time Constant : $malDelay$

Intention	: maximal delay to an appropriate reaction in the case of a malfunction of all temperature sensors of a room
Scope	: eh
References	: PD/NFH3
Comment	: Ergänzung zur angegebenen Referenz

Domain $ValvePosRange$

Intention	: Set of possible positions of a radiator valve
References	: PD/NFH3

Timed Function $valvePos() \rightarrow ValvePosRange$

Intention	: current position of the radiator valve
References	: PD/NFH3
CustomerReq	: Muß man bei der Ansteuerung der Radiatoren unterscheiden oder werden sie immer gleich eingestellt, was ist sinnvol? (Frage an den Domänenexperten)

Timed Function $malValPos(TEMP, ValvePosRange) \rightarrow ValvePosRange$

Intention	: In the case of a malfunction of the temperature sensors this function computes the position of the valves.
Scope	: mh
References	: PD/NFH3
Comment	: Diese Größe wird vom Kunden erwähnt.
CustomerReq	: Bisher wird nur die Einstellung von einem Nachbarraum betrachtet, weil noch nicht klar ist, welche Nachbarräume betrachtet werden sollen. Insbesondere gibt es in den Räumen ja mehrere Heizungen. Welche davon ist relevant, oder haben sie immer dieselbe Einstellung?
CustomerReas	: Unklarheit in der PD

Timed Function $setNeighRooms() \rightarrow ValvePosRange$

Intention	: Current setting of a radiator valve in a neighbour room.
Scope	: mh
References	: PD/NFH3
Comment	: Diese Größe wird vom Kunden erwähnt.
CustomerReq	: Bisher nur eine betrachtet, weil nicht klar ist, wieviele relevant sind.

Property

Formal	: $\Box(malfunctionAllTempSens \Rightarrow \_malDelay valvePos = malValPos(outTemp, setOfNeighRooms))$
NL	: Whenever all temperature sensors of a room do not work correctly for at least malDelay time units, then eventually within this time span, the position of the valves is determined depending on the outdoor temperature and the settings of the neighbour rooms.
References	: PD/NFH3
CustomerReq	: Steht diese Anforderung in Konflikt mit NFH2. Oder fällt diese Situation noch nicht unter "whole control system has a serious malfunction"? Sollte in die Berechnung von malValPos nicht auch der Aspekt "occupancy" berüchsichtigt werden? Welche Nachbarräume sollen mit berücksichtigt werden, auch die weiter entfernten oder die ein Stockwerk drüber oder drunter?
CustomerReas	: Unklarheit in der PD, Ergänzung zur PD

Property

Formal	: $\Box(occMal \leftrightarrow (stateManualEnteredOcc \rightarrow manualEnteredOcc) \wedge (\neg stateManualEnteredOcc \rightarrow (daytime \rightarrow DefaultOccDaytime) \wedge (\neg daytime \rightarrow DefaultOccNighttime)))$
NL	: The malfunction case of the occupancy status is determined as follows: - If a value for occupancy is entered manually, then this value is valid. - If no value for occupancy is entered manually, then it is distinguished between daytime and nighttime: * if it is daytime then the default value for occupancy during daytime is valid * otherwise if it is nightime then the default value for occupancy during nighttime is valid
References	: PD/NFH4
Comment	: Als Muster einführen, wie man eine Größe ausgehend von einer Fallunterscheidung definieren kann.

Property

Formal	: $\Box(malMotDet \Rightarrow \_malDelay(occUsed \leftrightarrow occMal))$
NL	: Whenever there is a malfunction of the motion detector for at least malDelay time units, then eventually within this time span, the status assumed by the machine corresponds with status of the malfunction case and remains so at least as long as there is a malfunction of the motion detector.
References	: PD/NFH4
PatternInst	: DelImp
Comment	: Die Formulierung ist noch verbesserungswürdig.

Description Class Room

Derived from the following classes:
Description Class Area

Intention: Contains the light relevant specification of environment and requirements common to any kind of room.

Signature

Object $cp : CtrlPanelLight$
Intention :

Object $win, wall : CeilingLightButtonSet(numButtons = 1)$
Intention :

Object $md : MotDet$
Intention :

Value $AMBIENT$

NL	: ambient
Intention	: a value that can be set by the control panel for ceiling lights meaning that the actually set ambient light level should be maintained
References	: PD/U14

Timed Predicate $userInformed$

NL	: user is informed
Intention	: the user is informed in a specific way
Scope	: eh
References	: PD/U11

Timed Function $curStandLightScene() \rightarrow LIGHT\_SCENES$

NL	: current standard light scene
Intention	: the standard light scene that is currently valid
Scope	: mh
References	: PD/NF2

Timed Function $faultStandLightScene() \rightarrow LIGHT\_SCENES$

NL	: fault standard light scene
Intention	: the light scene that has to become standard light scene in the case of a malfunction of the outdoor light sensors
Scope	: mh
References	: PD/NF2

Timed Function $noFaultStandLightScene() \rightarrow LIGHT\_SCENES$

NL	: no fault standard light scene
Intention	: the light scene that has to become standard light scene in the case of no malfunctions
Scope	: mh
References	: PD/U4

Timed Function $applLightScene() \rightarrow LIGHT\_SCENES$

NL : applied light scene

Intention : light scene applied by the control system

Scope : mh

Timed Function $ambLightLev() \rightarrow AMB\_LIGHT\_LEV$

NL : ambient light level

Intention : the ambient light level in a room

Scope : eh

References : PD-Sec3.1.1/U14(iii), PD-Sec3.1.1/U16(ii)

Time Constant T_3

Intention : The time delay between occupation of a room, when a light scene is chosen, and the establishment of the chosen light scene

Time Constant T_4

Intention : The time delay during which the reoccupation of a room leads to reestablishing the previous light scene and the time delay between a sufficiently often occupation and the application of the chosen light scene

Time Constant T_5

NL : immediately

Intention : The time delay between a sufficiently long non-occupation and the application of the standard light scene.

Time Constant T_6

Intention : The time delay between setting and applying a certain light level

ReqSpec

Property P_1

Formal : $\square(curStandLightScene = (0, 100, 100, \texttt{undef}))$

NL : The standard light scene in the case of a malfunction of the outdoor light sensors is that all ceiling lights are on and the task light is off (in the case of offices).

References : PD-Sec3.2.1/NF2

Comment : Das müssen wir anders formulieren.

Property P_2

Formal : $\square(\forall l \in LIGHT\_SCENES(occupied \wedge cp.chosenLightScene = l) \Rightarrow_{\leq T_3} applLightScene = l)$

NL : Whenever the room is occupied and light scene l is chosen for at least time T_3, this light scene l is applied within this time span and remains applied at least as long as the precondition is true.

References : PD-Sec3.1.1/U2,U1,U6,U12

PatternInst : ConditionalContinuity (occupied, cp.chosenLightScene)

Comment : P_2 hat Prioritaet gegenueber Area::P_1

Property P_3

Formal : $\square(([occupied] \wedge \blacklozenge\_ \leq \{(0, T\_4]\} occupied \wedge cp.chosenLightScene = l) \rightarrow \lozenge_{\leq T_5} applLightScene = l)$

NL : Whenever the room becomes occupied now and was occupied sometimes within the last T_4 minutes and the light scene l was last chosen, the light scene l is applied immediately.

References : PD-Sec.3.1.1/U3

Property P_4

Formal : $\square([occupied] \wedge \blacksquare\_ \leq \{(0, T\_4]\} \neg occupied) \rightarrow \lozenge\_ \leq T\_5\, applLightScene = curStandLightScene)$

NL : Whenever the room becomes occupied now and has not been occupied during the last T_4 minutes, the current standard light scene is applied immediately.

References : PD-Sec.3.1.1/U4

Property P_5

NonFormal : The light scenes can be chosen by using the control panel.

References : PD-Sec.3.1.1/U6

Property P_6

NonFormal : For each room a default light scene can be chosen (not by using the control panel).

References : PD-Sec3.1.1/U8

Property P_7

NonFormal : For each room a default ambient light level can be set (not by using the control panel).

References : PD-Sec3.1.1/U9

Property P_8

NonFormal : The value T_4 can be set for each room separately (not by using the control panel).

References : PD-Sec3.1.1/U10

Property P_9

Formal : $\square(\forall ll : (cp.winCeilLightSwitch = 0 \vee cp.winCeilLightSwitch = 100) \wedge ll = cp.winCeilLightSwitch) \Rightarrow \_ \leq T\_6\} win.light.status = ll)$

NL : Whenever the window ceiling light switch in the control panel is on (off) for at least T_6 time units, the window ceiling light will be on (off) within this time span and remains on (off) as long as the precondition is true.

References : PD-Sec3.1.1/U14(ii), PD-Sec3.1.1/U16(i)

Property P_10

Formal	: $\Box(\forall l : (cp.wallCeilLightSwitch = 0 \vee cp.wallCeilLightSwitch = 100) \wedge l = cp.wallCeilLightSwitch) \Rightarrow \_ \leq T\_6 wall.light.status = l)$
NL	: Whenever the wall ceiling light switch in the control panel is on (off) for at least T_6 time units, the wall ceiling light will be on (off) within this time span and remains on (off) as long as the precondition is true.
References	: PD-Sec3.1.1/U14(ii), PD-Sec3.1.1/U16(i)

Property P_11

Formal	: $\Box(cp.winCeilLightSwitch = ambient \Rightarrow \_ \leq T\_6\} win.light.status = \mathtt{undef})$
NL	: Whenever the window ceiling light switch in the control panel is ambient for at least T_6 time units, the window ceiling light will be undefwithin this time span and remains undefas long as the precondition is true.
References	: PD-Sec3.1.1/U14(ii), PD-Sec3.1.1/U16(i)

Property P_12

Formal	: $\Box(cp.wallCeilLightSwitch = ambient \Rightarrow \_ \leq T\_6 wall.light.status = \mathtt{undef})$
NL	: Whenever the wall ceiling light switch in the control panel is ambient for at least T_6 time units, the wall ceiling light will be undefwithin this time span and remains undefas long as the precondition is true.
References	: PD-Sec3.1.1/U14(ii), PD-Sec3.1.1/U16(i)

Property P_13

Formal	: $\Box(\forall ll : cp.ambLightLev = ll \Rightarrow \_ \leq T\_6 ambLightLev = ll)$
NL	: Whenever the ambient light level in the control panel is "ll" for at least T_6 time units, the ambient light level will be "ll" within this time span and remains "ll" as long as the precondition is true.
References	: PD-Sec3.1.1/U14(iii), PD-Sec3.1.1/U16(ii)

Property P_14

Formal	: $\Box(cp.chosenLightScene = (0, cp.winCeilLightSwitch, cp.wallCeilLightSwitch, cp.ambLightLev))$
NL	: The light scene chosen by the control panel corresponds with the settings of the switches of the control panel.
Comment	: Das müssen wir anders aufschreiben.(MK)

Description Class RoomHeating

Local Paramters:
Nat numWinSect

Intention: Contains the specification of environment and requirements common to any kind of room.

CustomerReq: Are there rooms, where some window sections have blinds and some not?

Comment: If not: distinguish between BlindedWindowSection and WindowSection, and use above only one parameter.

Signature

Object $cp : CtrlPanelHeating$
Intention :

Object $ws : WindowSection[numWinSect]$
Intention :
Comment : Not for some northern rooms!

Object $comfortMode : ComfortMode$
Intention :

Object $standbyMode : StandbyMode$
Intention :

Object $offMode : OffMode$
Intention :

Object $nonFreezMode : NonFreezeMode$
Intention :

Object $airTemp : RoomTempSensor[3]$
Intention :

Timed Function $temp() \rightarrow TEMP$
Intention : The currently measured room air temperature
References : ??

Time Constant : $timeSmall$
NL : immediately
Intention : small time delay between an event and an appropriate reaction
Scope : eh
DeveloperReq : Wozu?
CustomerReq : Welche Wert darf diese Konstante annehmen?
DeveloperReas : zur PD

Timed Function $outTemp() \rightarrow TEMP$

Intention	: value delivered by the outdoor temperature sensor
Scope	: ev
References	: PD/NFH3
Comment	: Es gibt sicherlich noch weitere Referenzen. Hier muß ein entsprechendes Objekt aggregiert werden.
DeveloperReq	: Sollte das nicht inzwischen ein Sensor machen ???????
CustomerReq	: Gibt es nur einen Temperatursensor für draußen?

DomKnow

Property D_1

Formal	: $\Box(meanVal(airTemp[??].measured) \Leftrightarrow_{\leq T} temp)$
NL	: The room temperature is always equal to the mean value of all air temperature sensors with respect to a time delay of T.

Description Class Room

Local Paramters:
Nat numWinSect
Derived from the following classes:
Description Class RoomLight
Description Class RoomHeating
Description Class Occupancy
Description Class AdaptiveOccupancyTimes
Description Class ExpectedUsage

Intention: Contains the specification of environment and requirements common to any kind of room.

CustomerReq: Are there rooms, where some window sections have blinds and some not?

Comment: If not: distinguish between BlindedWindowSection and WindowSection, and use above only one parameter.

Signature

Object $controlPanel : CtrlPanel(numWinSect)$

Intention	:
CustomerReq	: Ist die Anzahl der WindowSection gleich der Anzahl der Blinds? Ich glaube nicht! (MK)

DomKnow

Object $ws : WindowSection[numWinSect]$

Intention	:
Comment	: Not for some northern rooms!

ReqSpec

Property P_17

Formal	: $\Box(\forall s, 1 \leq s \leq numSunBlinds \forall x : (roomOccupied \wedge cp.sunBlindSwitch(s) = x) \Rightarrow_{-} \leq Tsb[s].status = x$
NL	: If the room is occupied, the status of each sunblind is equal to the corresponding switch status in the control panel.
References	: PD/UH6
CustomerReq	: - We suppose that there should be a possibility to reenable amnual or automatic sun blind control even before the room is left empty. - What does 'the instruction to move up or down' mean? Completely up (as supposed) or 'up as long as key pressed'?

Property

Formal : $\Box((controlPanel.adaptive.transferedValue = 1 \rightarrow dt1 = dt1Adaptive) \wedge (controlPanel.adaptive.transferedValue = 0 \rightarrow (expectedUsage \rightarrow dt1 = controlPanel.expectedUsageTimes.dt1.value) \wedge (\neg expectedUsage \rightarrow dt1 = controlPanel.notExpectedUsageTimes.dt1.value)))$

NL : The value dt1 used for determining occupancy is specified as follows:

- If the facility manager has chosen the adaptive mode for this room, dt1 is the value computed by the adaptive learning function.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is expected, dt1 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is expected.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is *not* expected, dt1 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is *not* expected.

Property

Formal : $\Box((controlPanel.adaptive.transferedValue = 1 \rightarrow dt2 = dt2Adaptive) \wedge (controlPanel.adaptive.transferedValue = 0 \rightarrow (expectedUsage \rightarrow dt2 = controlPanel.expectedUsageTimes.dt2.value) \wedge (\neg expectedUsage \rightarrow dt2 = controlPanel.notExpectedUsageTimes.dt2.value)))$

NL : The value dt2 used for determining occupancy is specified as follows:

- If the facility manager has chosen the adaptive mode for this room, dt2 is the value computed by the adaptive learning function.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is expected, dt2 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is expected.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is *not* expected, dt2 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is *not* expected.

Property

Formal	:	$\Box((controlPanel.adaptive.transferedValue = 1 \rightarrow dt3 = dt3Adaptive) \wedge (controlPanel.adaptive.transferedValue = 0 \rightarrow (expectedUsage \rightarrow dt3 = controlPanel.expectedUsageTimes.dt3.value) \wedge (\neg expectedUsage \rightarrow dt3 = controlPanel.notExpectedUsageTimes.dt3.value)))$
NL	:	The value dt3 used for determining occupancy is specified as follows:

- If the facility manager has chosen the adaptive mode for this room, dt3 is the value computed by the adaptive learning function.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is expected, dt3 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is expected.
- If the facility manager has *not* chosen the adaptive mode for this room and usage of this room is *not* expected, dt3 is the value as it is computed by the entered values of the user and the facility manager for the case that usage of the room is *not* expected.

Description Class Office

Intention: Additional requirements for rooms that are offices

Signature

Object $tl : LightButtonSet(numButtons = 1)$

Intention : The task light with its push button

Object $Room :: cp : OfficeCtrlPanel$

Intention :

DomKnow

Property D_1

Formal	:	$\Box(allLightsOff \leftrightarrow win.light.envEntity = 0 \wedge wall.light.envEntity = 0 \wedge tl.envEntity = 0)$
NL	:	All lights are off if and only if both ceiling lights and the task light are off.

ReqSpec

Property P_1

NonFormal	:	The control panel should be installed moveably like a telephone in the offices.
References	:	PD-Sec3.1.1/U13

Property P_2

Formal	:	$\Box(\forall b : cp.taskLightSwitch = b \Rightarrow_{\leq Room::T\_6} tl.envEntity = b)$
NL	:	Whenever the task light switch in the control panel is on (off) for at least Room::T_6 time units, the task light will be on (off) within this time span and remains on (off) as long as the precondition is true.
References	:	PD-Sec3.1.1/U14(i)
Comment	:	This could lead to potential conflicts with other requirements that also try to control the task light. Should be replaced by a formulation that allows other source predicates to have influence on the target predicate. Last access or higher priority wins? -¿ New pattern definition! This is an extended interpretation of need U14. Does it correspond with the customer's intention?

Description Class NoOffice

DomKnow

Property D_1

Formal	: $\Box(allLightsOff \leftrightarrow win.light.envEntity = 0 \wedge wall.light.envEntity = 0)$
NL	: All lights are off if and only if both ceiling lights are off.

ReqSpec

Property P_1

NonFormal	: The control panel should be installed near a door to the hallway.
References	: PD-Sec3.1.1/U15

Description Class HallwaySection

Derived from the following classes:
Description Class Area

CustomerReq: There are no temperature restrictions for hallways! Are you sure?

Signature

Object $eastMd, westMd, middleMd : MotDet$

Intention	:

Object $light : CeilingLightButtonSet(\texttt{undef})$

Intention	:

Timed Predicate: $nearHallwayDoor$

NL	: near a hallway door
Intention	: a person is near a hallway door to this hallway section
Scope	: eh
References	: PD/U18

Timed Predicate: $hwLightUncontroll$

NL	: hallway light uncontrollable
Intention	: the light is neither controllable automatically nor manually
Scope	: eh
References	: PD/NF5

Time Constant T_3

Intention	: The time delay between a sufficiently long occupation of an adjacent hallway section and switching on the lights

Time Constant T_4

Intention	: The time delay between the loss of ligth control and switching on the lights.

DomKnow

Property D_1

Formal	: $\Box(allLightsOff \leftrightarrow light.envEntity = 0)$
NL	: All lights are off if and only if the hallway light is off.

ReqSpec

Property P_1

Formal	:	$\Box(nearHallwayDoor \Rightarrow_{\leq T\_3} light.envEntity = 1)$
NL	:	Whenever a person is continuously in the near of a hallway door of the hallway section for at least T_3 time units, then the light in the hallway section will be on within this time span, and remains on at least as long as the person is near the hallway section.
References	:	PD-Sec3.1.1/U18
PatternInst	:	DelayedImplication (nearHallwayDoor, light.envEntity = 1)
Comment	:	"before" is here interpreted geographically and not temporally. The time T_3 should be set to value reasonable for typical pedestrian speeds.

Property P_2

Formal	:	$\Box(hwLightUncontroll \Rightarrow_{\leq T\_4} light.envEntity = 1)$
NL	:	Whenever the lights are neither controllable automatically nor manually for at least T_4 time units, then eventually within this time span, the light in the hallway section is on and remains so at least as long as the lights are neither controllable automatically nor manually.
References	:	PD-Sec3.2.1/NF5
PatternInst	:	DelayedImplication (hwLightUncontroll, light.envEntity = 1, T_4)

Description Class StairCase

Signature

Object $md : MotDet$

Intention	:	sensor placed in front of the adjacent hallway door for detecting motions

Main Description Class Floor

Signature

Object *o435* : *Office*
NL : office 435
Intention : the office in floor 32/4 with No 435
Scope : eh
References : BD/Figure 1

Object *o433*, *o432*, *o429*, *o425*, *o423*, *o421*, *o419*, *o417*, *o415*, *o424o416*, *o414*, *o412* : *Office*
Intention :

Object *m427*, *cl411*, *cl426*, *cl422*, *cl418*, *cl411* : *NoOffice*
Intention :

Domain *ROOMS* = {o435, *o433*, *o432*, *o429*, *o425*, *o423*, *o421*, *o419*, *o417*, *o415*, *o424*, *o416*, *o414*, *o412*, *p413*, *m427*, *cl411*, *cl426*, *cl422*, *cl418*, *cl411*}
Intention :

Domain *OFFICES* = {o435, *o433*, *o431*, *o429*, *o425*, *o423*, *o421*, *o419*, *o417*, *o415*, *o424*, *o416*, *o414*, *o412*}
Intention :

Domain *MeetingRooms* = {*m*427}
Intention :

Domain *ComputerLabs* = {*cl*411, *cl426*, *cl422*, *cl418*, *cl411*}
Intention :

Domain *PeripheralRooms* = {*p*413}
Intention :

Object *fm* : *FacilityManager*(*ROOMS*)
Intention : represents the facility manager
Scope : eh
Comment : Der Domänenparameter ist rausgeflogen −¿ anpassen.
DeveloperReq : Brauchen wir den Domänenparameter noch?

Object *d_421_423* : *Door*
Intention :

Object *d_423_425* : *Door*
Intention :

Object *d_431_433* : *Door*
Intention :

Object *d_433_435* : *Door*
Intention :

Object *d_414_416* : *Door*
Intention :

Object *d_416_418* : *Door*
Intention :

Object *d_418_420* : *Door*
Intention :

Object *d_420_422* : *Door*
Intention :

Object *d_o435*, *d_o433*, *d_o432*, *d_o429*, *d_o425*, *d_o423*, *d_o421*, *d_o419*, *d_o417*, *d_o415*, *d_o424*, *d_o416*, *d_o414*, *d_o412*, *d_m427*, *d_cl411*, *d_cl426*, *d_cl422*, *d_cl418*, *d_cl411* : *Door*
Intention :

Object *h1* : *HallwaySection*
NL : hallway 1
Intention : hallway in the east of floor 32/4
References : InAr/2.1

Object *h2* : *HallwaySection*
NL : hallway 2
Intention : hallway in the middle of floor 32/4
References : InAr/2.1

Object *h3* : *HallwaySection*
NL : hallway 3
Intention : hallway in the west of floor 32/4
References : InAr/2.1

Object *ols1* : *OutdoorLightSensor*
NL : outdoor light sensor 1
Intention : outdoor light sensor in the north-east direction
References : InAr/2.1 (ols1)

Object *ols2* : *OutdoorLightSensor*
NL : outdoor light sensor 2
Intention : outdoor light sensor in the south-west direction
References : InAr/2.1 (ols2)

Object *ols3* : *OutdoorLightSensor*
NL : outdoor light sensor 3
Intention : outdoor light sensor in the north-west direction
References : InAr/2.1 (ols3)

Object *ols4* : *OutdoorLightSensor*
NL : outdoor light sensor 4
Intention : outdoor light sensor in the south-east direction
References : InAr/2.1 (ols4)

Object *ols5* : *OutdoorLightSensor*
NL : outdoor light sensor 5
Intention : outdoor light sensor in the north direction
References : InAr/2.1 (ols5)

Object *ols6* : *OutdoorLightSensor*

NL	:	outdoor light sensor 6
Intention	:	outdoor light sensor in the south direction
References	:	InAr/2.1 (ols6)

Object *scw* : *StairCase*

NL	:	western staircase
Intention	:	staircase in the west of floor 32/4
References	:	InAr/2.1

Object *sce* : *StairCase*

NL	:	eastern staircase
Intention	:	staircase in the east of floor 32/4
References	:	InAr/2.1

Time Constant T_1

Intention : The time delay between the outdoor light value returned by the sensor and the outdoor light value used by the system.

Time Constant T_2

Intention : The time delay between a malfunction and the information of the facility manager

Time Constant T_3

Intention : The time delay between a malfunction of the outdoor light sensor and the time a user is informed about this malfunction

Time Constant T_4

Intention : The time delay between a correct sensor operation (sensor fault) and the corresponding system behaviour.

Time Constant T_5

Intention : The time delay between a malfunction of the outdoor light sensor and the corresponding system behaviour

Time Constant T_6

Intention : The time delay between a malfunction of the outdoor light sensor and switching on the lights if the hallway is occupied.

Timed Predicate: *malfunction*

NL	:	malfunction
Intention	:	a malfunction occurs
Scope	:	eh
References	:	FM7

Timed Predicate: *safeLight*

NL	:	safe light
Intention	:	light is sufficient to move safely
Scope	:	eh
References	:	PD/U17

Timed Predicate *outLightSensorFaultEnv*

Intention	:	This predicate is true iff some outdoor light sensors have a malfunction. Note that this predicate represents the situation as it is in the real world and not as it is seen by the machine.
Scope	:	mh

Timed Predicate *outLightSensorFaultMach*

Intention	:	This predicate is true iff some outdoor light sensors have a malfunction. Note that this predicate represents the situation as it is seen by the machine.
Scope	:	mh

ReqSpec

Initialization

Property I1

Formal	:	$\forall o \in Offices : o.DT1 = fm.OfficesTempSetting.DT1Default$
NL	:	For all offices the initial value of DT1 is the default value of DT1 set by the facility manager for offices.
References	:	PD/Par7
CustomerReq	:	Ist das die gewünschte Initialisierung?
CustomerReas	:	Unklarheit in der PD

Property I2

Formal	:	$\forall o \in Offices : o.DT2 = fm.OfficesTempSetting.DT2Default$
NL	:	For all offices the initial value of DT2 is the default value of DT2 set by the facility manager for offices.
References	:	PD/Par7
CustomerReq	:	Ist das die gewünschte Initialisierung?
CustomerReas	:	Unklarheit in der PD

Property I3

Formal	:	$\forall o \in Offices : o.DT3 = fm.OfficesTempSetting.DT3Default$
NL	:	For all offices the initial value of DT3 is the default value of DT3 set by the facility manager for offices.
References	:	PD/Par7
CustomerReq	:	Ist das die gewünschte Initialisierung?
CustomerReas	:	Unklarheit in der PD

Property I4

Formal	:	$\forall o \in MeetingRooms : o.DT1 = fm.MeetRoomTempSetting.DT1Default$
NL	:	For all meeting rooms the initial value of DT1 is the default value of DT1 set by the facility manager for meeting rooms.
References	:	PD/Par7
CustomerReq	:	Ist das die gewünschte Initialisierung?
CustomerReas	:	Unklarheit in der PD

Property I5

Formal : $\forall o \in MeetingRooms : o.DT2 = fm.MeetRoomTempSetting.DT2Default$
NL : For all meeting rooms the initial value of DT2 is the default value of DT2 set by the facility manager for meeting rooms.
References : PD/Par7
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I6

Formal : $\forall o \in MeetingRooms : o.DT3 = fm.MeetRoomTempSetting.DT3Default$
NL : For all meeting rooms the initial value of DT3 is the default value of DT3 set by the facility manager for meeting rooms.
References : PD/Par7
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I7

Formal : $\forall o \in ComputerLabs : o.DT1 = fm.CompLabTempSetting.DT1Default$
NL : For all computer labs the initial value of DT1 is the default value of DT1 set by the facility manager for computer labs.
References : PD/Par7
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I8

Formal : $\forall o \in ComputerLabs : o.DT2 = fm.CompLabTempSetting.DT2Default$
NL : For all computer labs the initial value of DT2 is the default value of DT2 set by the facility manager for computer labs.
References : PD/Par7
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I9

Formal : $\forall o \in ComputerLabs : o.DT3 = fm.CompLabTempSetting.DT3Default$
NL : For all computer labs the initial value of DT3 is the default value of DT3 set by the facility manager for computer labs.
References : PD/Par7
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I10

Formal : $\forall o \in PeripheralRooms : o.appliedDT1 = fm.PeriRoomTempSetting.DT1Default$
NL : For all peripheral rooms the initial value of DT1 is the default value of DT1 set by the facility manager for peripheral rooms.
References : PD/Par7
DeveloperReq : Dies impliziert aufgrund der Anforderungen in RoomTemperature, daß auch der initiale Wert von DT1.transfered entsprechend ist. Sollen wir dies hier auch noch explizit hinschreiben? Ich würde dies vorschlagen, da wir ja auch was zu den initialen Werten aller timed Größen sagen sollten.
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I11

Formal : $\forall o \in PeripheralRooms : o.appliedDT2 = fm.PeriRoomTempSetting.DT2Default$
NL : For all peripheral rooms the initial value of DT2 is the default value of DT2 set by the facility manager for peripheral rooms.
References : PD/Par7
DeveloperReq : Dies impliziert aufgrund der Anforderungen in RoomTemperature, daß auch der initiale Wert von DT2.transfered entsprechend ist. Sollen wir dies hier auch noch explizit hinschreiben? Ich würde dies vorschlagen, da wir ja auch was zu den initialen Werten aller timed Größen sagen sollten.
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

Property I12

Formal : $\forall o \in PeripheralRooms : o.appliedDT3 = fm.PeriRoomTempSetting.DT3Default$
NL : For all peripheral rooms the initial value of DT3 is the default value of DT3 set by the facility manager for peripheral rooms.
References : PD/Par7
DeveloperReq : Dies impliziert aufgrund der Anforderungen in RoomTemperature, daß auch der initiale Wert von DT3.transfered entsprechend ist. Sollen wir dies hier auch noch explizit hinschreiben? Ich würde dies vorschlagen, da wir ja auch was zu den initialen Werten aller timed Größen sagen sollten.
CustomerReq : Ist das die gewünschte Initialisierung?
CustomerReas : Unklarheit in der PD

End of Initialization

Property P44

Formal : $\Box(buildingNonFreezePeriod.actual \rightarrow \forall r \in Roomsr.opMode = nonFreeze)$
References : PD/FMH5

Property P4

Formal : $\Box(adaptiveDelayTimes \rightarrow \forall r \in Roomsr.appliedDT1 = learnDT1(r))$

NL : If the facility manager has decided that the delay times have to be learned the current applied value of DT1 is the one returned by the learning function learnDT1.

References : PD/Par7, PD/FMH6

Comment : Beachte: diese Anforderung steht im Konflikt mit der Anforderung, die in RoomTemperature gestellt wird. Die Lösung dieses Konfliktes, nämlich daß die Lernerei Priorität hat, muß noch geeignet dokumentiert werden.

CustomerReq : Kann man von einer gemeinsamen Lernfunktion für alle Räume ausgehen? Was sind die Parameter, von denen die Lernfunktion abhängen soll? Sind die Beispiele in der PD konkrete Vorgaben?

CustomerReas : Unklarheit in der PD

Property P5

Formal : $\Box(adaptiveDelayTimes \rightarrow \forall r \in Roomsr.appliedDT2 = learnDT2(r))$

NL : If the facility manager has decided that the delay times have to be learned the current applied value of DT2 is the one returned by the learning function learnDT2.

References : PD/Par7, PD/FMH6

Comment : Beachte: diese Anforderung steht im Konflikt mit der Anforderung, die in RoomTemperature gestellt wird. Die Lösung dieses Konfliktes, nämlich daß die Lernerei Priorität hat, muß noch geeignet dokumentiert werden.

CustomerReq : Kann man von einer gemeinsamen Lernfunktion für alle Räume ausgehen? Was sind die Parameter, von denen die Lernfunktion abhängen soll? Sind die Beispiele in der PD konkrete Vorgaben?

CustomerReas : Unklarheit in der PD

Property P6

Formal : $\Box(adaptiveDelayTimes \rightarrow \forall r \in Roomsr.appliedDT3 = learnDT3(r))$

NL : If the facility manager has decided that the delay times have to be learned the current applied value of DT3 is the one returned by the learning function learnDT3.

References : PD/Par7, PD/FMH6

Comment : Beachte: diese Anforderung steht im Konflikt mit der Anforderung, die in RoomTemperature gestellt wird. Die Lösung dieses Konfliktes, nämlich daß die Lernerei Priorität hat, muß noch geeignet dokumentiert werden.

CustomerReq : Kann man von einer gemeinsamen Lernfunktion für alle Räume ausgehen? Was sind die Parameter, von denen die Lernfunktion abhängen soll? Sind die Beispiele in der PD konkrete Vorgaben?

CustomerReas : Unklarheit in der PD

Property P7

NonFormal : Energy consumption should be reduced as far as possible during maintaining of comfortable working conditions for all users.

References : PD/FMH1

Comment : Mal sehen, wie man das konkretisieren kann!

Property P8

NonFormal : The use of solar radiation for heating during all times if heating is required or not disallowed, and the use of storage capacity of the building should always be preferred against the usage of the central heating unit.

References : PD/FMH2

Comment : Mal sehen, wie man das konkretisieren kann!

Property P_1

Formal : $\Box(outLightSensorFaultMach \leftrightarrow ols1.malfunction \vee ols2.malfunction \vee ols3.malfunction \vee ols4.malfunction \vee ols5.malfunction \vee ols6.malfunction)$

NL : Some outdoor light sensor has a malfunction. This is a specification, i.e. it describes the situation as it should be seen by the machine.

Property P_4

NonFormal : If a malfunction occurs, the control system supports the facility manager by finding the reason.

References : PD-Sec3.1.2/FM8

Comment : We suppose that "finding the reason" means locating the fault in the future control system architecture. But architecture is something unknown to this specification level.

Property P_5

NonFormal : The system provides reports on current and past energy consumption.
References : PD-Sec3.1.2/FM9

Property P_6

NonFormal : All malfunctions and unusual conditions are stored and reported on request
References : PD-Sec3.1.2/FM10

Property P_7

NonFormal : Malfunctions that the system cannot detect can be entered manually.
References : PD-Sec3.1.2/FM11

Property P_10

Formal : $\Box(\forall r \in ROOMS : outLightSensorFault \Rightarrow_{\leq T\_5} r.curStandLightScene = r.faultStandLightScene)$
NL : Whenever some outdoor light sensors are not working correctly for at least T_5 time units, then eventually within this time span, the standard light scene is "all ceiling lights on" and remains so at least as long as the outdoor light sensor does not work correctly.
References : PD-Sec3.2.1/NF2
PatternInst : DelayedImplication (outLightSensorFault, curStandLightScene = faultStandLightScene, T_5)

Property P_11

Formal : $\Box(\forall h \in HALLWAYS : outLightSensorFault \land h.occupied \Rightarrow_{\leq T\_6} h.light.envEntity = 1)$
NL : Whenever some outdoor light sensor are not working correctly and the hallway is occupied for at least T_6 time units, then eventually within this time span, the light in the hallway is on and remains on at least as long as the precondition is true.
References : PD-Sec3.2.1/NF3
PatternInst : DelayedImplication (outLightSensorFault / occupied, h.light.envEntity = 1, T_6)
Comment : Aehnlichkeit zu NF2ii: evtl. Area::Timed Predicate faultLight und verfeinern als Room::curStandLightScene = faultStandLightScene, bzw. HallwaySection::light.envEntity = 1

Property P_12

NonFormal : All hardware connections have to be made according to DIN standards.
References : PD-Sec3.2.2/NF6

Property P_13

NonFormal : No hazardous condition for persons, inventory or building are allowed.
References : PD-Sec3.2.2/NF7
Comment : Formalize?

Property P_14

NonFormal : The system warns about unreasonable inputs.
References : PD-Sec3.2.3/NF9
Comment : Formalize?

Comment: Beziehen sich diese non functional needs nur auf die Temperatursteuerung oder auch auf die Lichtsteuerung?

Comment: Das hier muß noch "einsortiert werden".

ReqSpec

Property P1

NonFormal : If only one single sensor of a group is damaged, the other ones shall provide makeshift operation, otherwise default values respectively manually entered values shall be used.
References : PD/NFH1
Comment : Aufgrund der Fülle der Fragen wird der Versuch einer Formalisierung unterlassen.
CustomerReq : Um welche Sensoren geht es hier?
Was ist hier mit "Gruppe" von Sensoren gemeint? Ich kann mir als Gruppe die Außenhelligkeitssensoren oder die Temperatursensoren in einem Raum vorstellen. Es ist auch die Rede von einer Gruppe von Bewegungsmeldern in einem Raum, aber auf die hat das Kontrollsystem ja nur gebündelt Zugriff.
Was heißt "damaged"? Wie kann das System das feststellen, nur über die manuelle Eingabe oder sollen auch Plausibilitätstest gemacht werden? Wie könnten die aussehen? (Hierzu brauchen wir den Domänenexperten!) Es gibt keine Sensoren, die von sich aus mitteilen, daß etwas kaputt ist.
Es werden drei mögliche Reaktionen auf das Erkennen eines "Schadens" angegeben: - makeshift operation - default values - manually entered values Was bedeutet "makeshift operation"? Wer kann die Werte wo eingeben? In welcher Relation stehen die drei Alternativen, d.h. welche ist wann anzuwenden.
CustomerReas : Unklarheit in der PD

Property P2

NonFormal	:	If the whole control system has a serious malfunction, complete manually operation should be possible.
References	:	PD/NFH2
Comment	:	Aufgrund der Fülle der Fragen wird der Versuch einer Formalisierung unterlassen.
CustomerReq	:	Was bedeutet "whole control system has a serious malfunction"? Wie kann dies festgestellt werden? Was bedeutet "complete manually operation", d.h. welche Sachen sollen auf welche Art noch gesteuert werden können? Ist eine manuelle Steuerung nur in einem solchen Fall möglich?
CustomerReas	:	Unklarheit in der PD

Property P3

NonFormal	:	If only one sensor does not work correctly, the other ones shall be used to get an estimation.
References	:	PD/NFH3
CustomerReq	:	In welchem Zusammenhang steht dieser Need mit NFH1? Besagen beide Needs dasselbe? Geht es hier nur um die indoor temp. sensors? Ist hier "estimation" nur eine andere Umschreibung für "makeshift operation" in NFH1? Wie kann das Fehlverhalten eines Sensors festgestellt werden?
CustomerReas	:	Unklarheit in der PD

Property

NonFormal	:	All hardware connections have to be made according to DIN standards.
References	:	PD/NFH5
Comment	:	In diesem Formalismus nicht weiter formalisierbar, da hier kein Verhalten des Systems beschrieben wird.

Property

NonFormal	:	No hazardous conditions for persons, inventory, or building is allowed.
References	:	PD/NFH6
CustomerReq	:	Dies muß konkretisiert werden; hierzu ist die Hilfe des Domänenexperten erforderlich.

Property

NonFormal	:	The control panel should be easy and intuitive to use.
References	:	PD/NFH7
Comment	:	In diesem Formalismus nicht weiter formalisierbar, da hier kein Verhalten des Systems beschrieben wird. Solche Anforderungen sollten wir an eigener Stelle sammeln.

Property

NonFormal	:	The control system shall be designed such that it can be easily changed and expanded.
References	:	PD/NFH8
Comment	:	In diesem Formalismus nicht weiter formalisierbar, da hier kein Verhalten des Systems beschrieben wird.

Property

NonFormal	:	In the case of malfunctions the system shall provide stepwise degradation of functionality. The use of manually entered or default values should provide makeshift operation.
References	:	PD/NFH9
CustomerReq	:	Dies muß konkretisiert werden. In welchem Zusammenhang stehen diese Needs mit denen in NFH1, NFH2, NFH3, NFH4. Besagen die nicht alle dasselbe, wenn man sie konkretisieren würde?

Index