Review von Requirements Specification Vers. 1.0

Organizational Aspects:

ProjectName: Light_32_4
Authors: Kronenburg
Status: ready

Remarks:

U1: ich würde *dark* erlauben

U2: light scene kann sich ändern, d.h. "light scene = true" gdw irgendeinen Szenario wurde ausgewählt, also ANY

U3:
- Unterschied und Zusammenhang zwischen `chosenLightScene` und `appliedLightScene` deutlicher machen. Ist `choseLightScene` stets vom Benutzer festgelegt? Hier müssen wohl noch einige Eigenschaften zwischen diesen Größen festgelegt werden.
- zu *Formal*: da es *reoccupied* in der PD heißt, muß eine Aktion betrachtet werden. also $[roomOccupied]$. Auch benötigen wir eine strikte Version der temporallogischen Operatoren, denn sonst gilt $[roomOccupied] \rightarrow \blacklozenge_{\leq T_1} roomOccupied$ und wir können diesen Teil weglassen.
- zu *ML*: auch hier deutlicher machen, daß der Raum zuvor leer gewesen sein muß.
- Neben *Predicates* auch einen Slot für Konstanten, wie z.B. T_1 einführen.
- Es sollte ein Wertebereich für `appliedLightScene` und `chosenLightScene` eingeführt werden.

U4:
- selbe Problem mit dem Wertebereich wie bei U3
- zu *Formal*: auch hier wie bei U3 strikte Operatoren verwenden. Beachte: $[roomOccupied] \land \blacksquare \neg roomOccupied$ ist für eine nicht strikte Version von $\blacksquare$ nicht erfüllbar. Außerdem muß die indizierte Fassung von $\blacksquare$ verwendet werden.

U5: siehe Bemerkung zu U19, kann hier entsprechend verwendet werden

U6:
- Ist `chosenLightScene` hier korrekt, oder muß hier `applidLightScene` stehen?
- zu *Formal*: ich würde *Delayed Implication* inklusive einer Zeitschranke wählen, weil es ansonsten unrealistisch ist. NL mus dann entsprechend angepaßt werden.
- zu *Comment*: habe ich nicht verstanden!

U7a/b:
- genau wie bei `LightScene` müssen auch die Prädikate `chosenLightLevel` und `appliedLightLevel` genauer definiert werden und ihr Zusammenhang klarer gemacht werden.
(Ich glaube, es ist sehr wichtig, diese Prädikate und Größen möglichst früh exakt über Umgebungsgrößen zu definieren, weil man sonst nicht weiß, wovon man redet!)
- zu *Formal*: warum nur bei U7b *DelayedImplication*? Warum bei U7a keine Verzögerung?
- Sind U7a/b Alternativen oder müssen sie zusammenbetrachtet werden?

U8/9: Auf der formalen Ebene betrachtet, bedeuten diese Anforderungen zwei Dinge:

- es wird etwas über den Wertebereich von `LightScene` und `LightLevel` gesagt, nämlich daß dieser Wertebereich einen Wert *default* enthält.
- es wird etwas darüber gesagt, wie der Wert von *default* fest- bzw. nicht festgesetz werden kann.
(Man müßte also festlegen können, welche Komponenten welche Größen beeinflussen bzw. nicht beeinflussen können. Dazu müßte man aber bereits zumindest über *logische* Komnponenten reden können und diese einführen. Das Einführen solcher *logischen Komponenten* erscheint mir recht wesentlich und höchst interessant zu sein. Das sollte man durchaus bereits auf der Anforderungsebene betrachten und formalisieren! Vorstellbar ist zum Beispiel im Falle von U8/9 eine Art `preserve`-Prädikat)

U10:
- Hier treffen im wesentlichen die Bemerkungen zu U8/9 zu.
- zu *Comment:* ich würde sagen *dynamically* trifft zu.

U11: kein Kommentar

U12: Um dies weiter formalisieren zu können, müßte man die Ausdruckstärke unserer Logik erweitern (dies haben wir bereits bei U3 getan!). Wir brauchen PL1, d.h. nicht nur aussagenlogische Prädikate, sondern auch beliebige Prädikate und Terme, d.h. Funktionen. Da dies wohl nicht allzu viel Aufwand bedeutet, sollten wir dies tun.

U13: kein Kommentar

U14: U14 enthält zwei Aspekte:

1. das was in U14(i)-(iii) steht, also die Funktionalität
2. Anforderungen an den Aufbau des Control Panels. Diese sollten auch berücksichtigt werden.

- zu U14(i): ich würde kein neues Anforderungs-Pattern definieren; es werden vielmehr Pattern fürs Auflösen von Konflikten benötigt; zwei stehen bereits dort: *lastAccess* und *higherPriority*.
- zu U14(ii): Comment: geht in Richtung der größeren Ausdrucksstärke

U15: kein Kommentar

U16: explizit nochmal hierhin kopieren

U17:
- wo kommt *lightScene* her?
- ich würde delayed implication pattern vorschlagen

U18:
- statt $i+1$ würde ich j verwenden
- wieder das Problem mit der Ausdruckstärke

U19: Bei den Schaltern handelt es sich um Taster, ich würde dies wie folgt formalisieren:

Predicates:	`switchPushed`:	switch is pushed down
	`lightOn(j)`:	the light is on in hallway j
	`lightOff(j)`:	the light is off in hallway j

$\square([\texttt{switchPushed}] \wedge \texttt{lightOn(j)} \rightarrow_{\leq T_{19}} \texttt{lightOff}(j)$
$\square([\texttt{switchPushed}] \wedge \texttt{lightOff(j)} \rightarrow_{\leq T_{19}} \texttt{lightOn}(j)$
Die Schalter müssen auch noch parametrisiert werden.
Dies sollte annalog für U5 getan werden.
Hierbei handelt es sich sich um ein Muster, das ähnlich zu conditional bounded response ist.

FM1:
- dies ist vermutlich die geeignetste Lösung
- als *Comment* einfügen, daß diese Anforderung im Konfliktfall stets minimale Priorität hat; dies beinhaltet das *whenever possible*.

FM2: kein Kommentar

FM3: `lightOff` ist wohl eine Abkürzung für alle Lichter in einem Raum. Hier ist es mal wieder offensichtlich, daß eine Anbindung an die Umwelt wesentliche Voraussetzung ist, d.h. die Größen wohldefiniert in der Umgebung sind.

FM4,5: siehe Bemerkungen zu U8/U9

FM6:
- etwas verwirrend: *FM has turned the light off*; eher wohl so was wie *FM hat veranlaßt, das Licht auszumachen*.
- bei *NL*: "???" ersetzen durch hallway/room
- hier wird es wieder deutlich, daß wir eine Art Parametrisierung benötigen; denn diese Anforderungen, wie auch fast alle anderen, müssen ja für jeden einzelnen Raum bzw. Gang gelte. Hierzu müssen wir etwas sagen.
- Warum nicht delImpl-Muster verwendet?

FM7:
- Ob hier delImpl oder delEquiv verwendet werden sollte, hängt vor allem davon ab, was *informed* bedeutet
- Es ist die Frage, ob man Anforderungen formalisieren soll oder versuchen soll, dies zu tun, bei denen man die verwendeten Größen nicht alle in der Umwelt verankern kann, wie hier zum Beispiel *informed*.

FM8-11: kein Kommentar

NF1: – NF1a steht so nicht explizit in der Problembeschreibung; die Frage ist, ob man eine solche Anforderung aufnehmen sollte. Wenn man diese beiden unterschiedlichen Größen hat, erscheint mir dies sinnvoll zu sein, um den Zusammenhang klar zu machen.

– zu Comment: ich würde Zeiten einführen, weil eine Zuweisung stets Zeit erfordert.

– zu NF1b: hier verstehe ich den Zusammenhang zwischen Formalisierung und NL nicht, `outdoorLightValue` ist doch kein boolean-Wert; man müßte hierzu den Aktionsbegriff erweitern.

– ich bin mir nicht sicher, ob dies den need aus der Problembeschreibung korrekt erfaßt; man interessiert sich ja für *last correct meassurement* – unabhängig davon, ob man dies entscheiden kann oder nicht

NF2: – wieder das Problem mit der Ausdrucksstärke

– Den comment verstehe ich nicht.

NF3: kein Kommentar

NF4: – zu NF4a: selbe Bemerkungen wie zu NF1a

– zu NF4b: stimmt so nicht; `motionValue` = `occupied` wird gefordert.

NF5: Sollten wir hier nicht auch delImpl verwenden?

NF6-9: kein Kommentar