

**Activity 04**  
**Smart building as a JaCaMo MAS**

Elena Yan  
elena.yan@studio.unibo.it

25 gennaio 2023

## Indice

<b>1</b>	<b>Analisi del problema</b>	<b>3</b>
<b>2</b>	<b>Architettura</b>	<b>3</b>
2.1	Organizzazione . . . . .	5
<b>3</b>	<b>Comportamento</b>	<b>7</b>
<b>4</b>	<b>Applicazione</b>	<b>8</b>

## 1 Analisi del problema

Il secondo punto dell'Activity 04 consiste nell'estendere il servizio della *smart-room* del punto precedente in un'ottica di *smart-building* progettando come un'organizzazione MAS (*Multi-Agent-System*) utilizzando il framework JaCaMo.

La programmazione Multi Agente include tre dimensioni fondamentali: gli agenti, l'ambiente e l'organizzazione. *JaCaMo*[3] è un *framework* che include strumenti per la programmazione di questi tre dimensioni.

In un'ottica *smart-building*, significa che ogni stanza dell'edificio avrà il proprio sistema di *smart-room*, composto da agenti che coordinano l'attività dei sensori e degli attuatori per raggiungere l'obiettivo comune dell'organizzazione.

## 2 Architettura

Nell'ambito della progettazione multi-agente, un sistema viene modellato e progettato come un'organizzazione di agenti autonomi situati e interagenti in un ambiente. Per cui l'architettura del sistema è composta da tre principali livelli, come rappresentato nella figura 1:

- **l'organizzazione**, dove vengono esplicitamente catturati gli aspetti principali che caratterizzano le funzionalità del sistema nel suo complesso. In questo livello vengono definite i diversi gruppi dell'organizzazione, lo schema che devono seguire e le missioni da svolgere.
- **gli agenti**, che rappresentano le entità progettate per raggiungere autonomamente gli obiettivi assegnati, incapsulano un proprio flusso logico di controllo e prendendo decisioni su come comportarsi e interagire.
- **l'ambiente logico**, rappresenta il contesto in cui gli agenti sono situati e agiscono per raggiungere i loro obiettivi. Questo livello include le risorse e gli strumenti che gli agenti possono utilizzare per svolgere i loro compiti. Nel caso della *smart-building*, l'ambiente logico è costituito dagli artefatti *LampThingProxyArtifact*, *LightThingProxyArtifact* e *PresDetectThingProxyArtifact* che fungono da intermediari tra gli agenti e i dispositivi fisici, consentendo loro di comunicare e di eseguire azioni sul dispositivo fisico. Oltre a questi, c'è un ulteriore artefatto *SmartBuildingGUI* che rappresenta l'applicazione con cui è possibile visualizzare e gestire tutte le stanze dell'organizzazione.

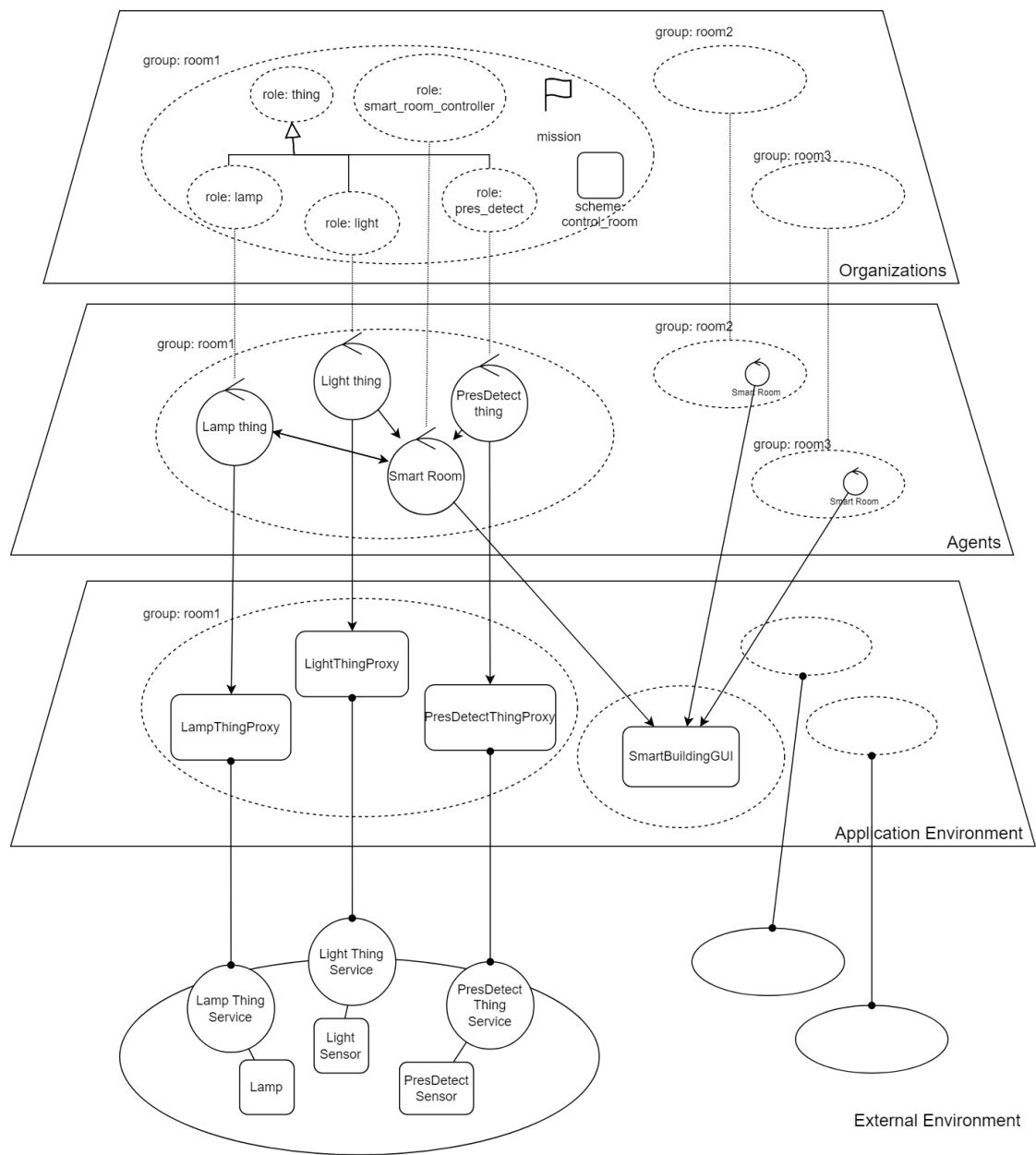


Figura 1: Panoramica dell'architettura del sistema

## 2.1 Organizzazione

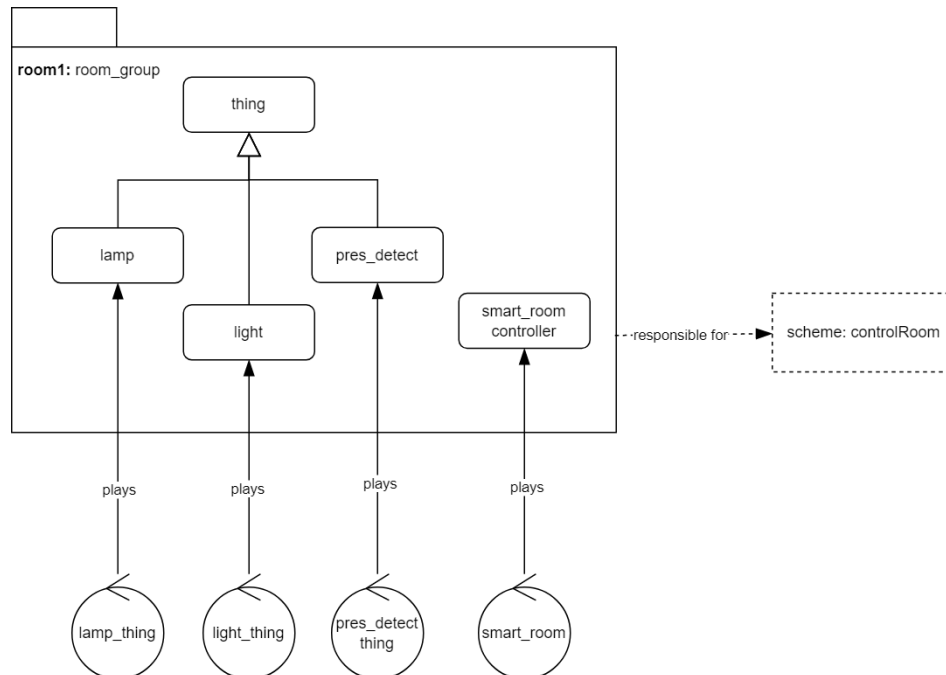


Figura 2: Formazione del gruppo all'interno dell'organizzazione

L'organizzazione della *smart-building* è costituita dalle diverse stanze dell'edificio, ogni stanza viene rappresentata come un gruppo dell'organizzazione. Come rappresentato nel diagramma in figura 2, all'interno del gruppo sono definiti i seguenti ruoli:

- **smart\_room\_controller**: gli agenti con questo ruolo sono responsabili del controllo delle condizioni dell'ambiente e della gestione delle politiche di accensione o spegnimento della luce;
- **thing**: rappresenta un generico ruolo per indicare i dispositivi della stanza. Questo ruolo viene ereditato dai seguenti sotto-ruoli:
  - **lamp**: gli agenti con questo ruolo sono responsabili dell'esecuzione delle azioni sulla luce;
  - **light**: gli agenti con questo ruolo sono responsabili della rilevazione della luminosità all'interno delle stanze e comunicano i valori rilevati all'agente *smart\_room\_controller*;
  - **pres\_detect**: gli agenti con questo ruolo sono responsabili della rilevazione della presenza di persone all'interno delle stanze e comunicano i valori rilevati all'agente *smart\_room\_controller*.

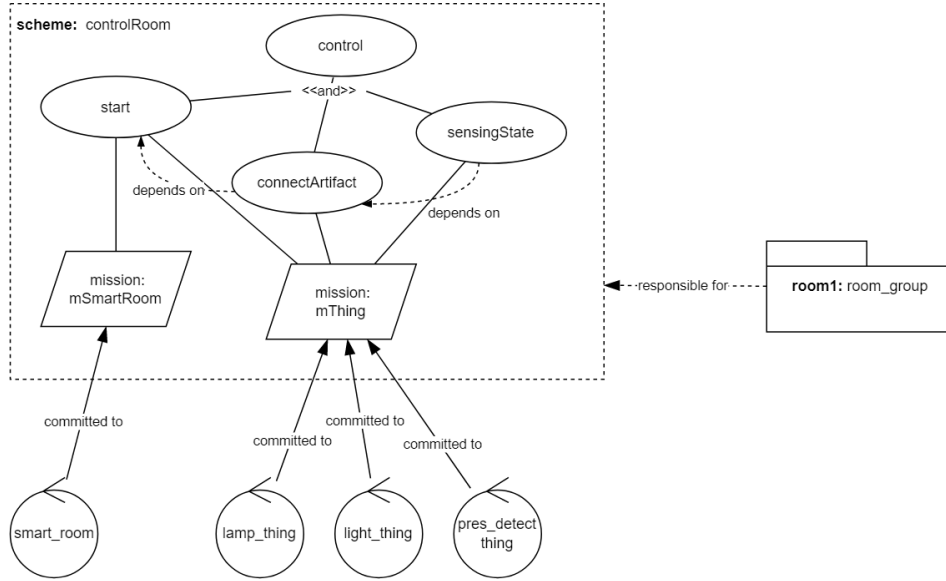


Figura 3: Specifica funzionale dell'organizzazione

Una volta che sono stati definiti i ruoli e la formazione del gruppo, è possibile definire la specifica funzionale (*functional specification*) dell'organizzazione. La specifica funzionale definisce il comportamento come una lista di schemi. Lo schema raggruppa un albero di decomposizione degli obiettivi in cui la radice rappresenta un obiettivo globale e le foglie sono obiettivi che possono essere soddisfatti dagli agenti. Come si può vedere nella figura 3, l'obiettivo comune è rappresentato dal controllo della stanza e viene decomposto in una sequenza di tre sotto-obiettivi:

- **start**, l'obiettivo di avviare il controllo della stanza con le opportune configurazioni;
- **connectArtifact**, l'obiettivo di stabilire la connessione tra gli agenti e gli artefatti *proxy*, in modo da poter comunicare con i dispositivi fisici;
- **sensingState**, l'obiettivo di rilevare i valori dal dispositivo ed l'invio dell'informazione alla *smart-room-controller*.

Questi obiettivi hanno una relazione di dipendenza per esprimere che il raggiungimento di un obiettivo dipende dal raggiungimento di un altro obiettivo.

Successivamente abbiamo un elenco di missioni, in cui ogni missione raccoglie gli obiettivi che possono essere assegnati a un agente che partecipa all'organizzazione. L'agente incaricato della missione cercherà di raggiungere tutti gli obiettivi contenuti nella missione. Nel nostro caso, abbiamo due missioni:

- **mSmartRoom**, viene incaricato all'agente *smart-room-controller*. Questa missione comprende solo l'obiettivo *start*.

- **mThing**, viene incaricato ai dispositivi della stanza, ovvero agli agenti *lamp\_thing*, *light\_thing* e *pres\_detect\_thing*. Questa missione comprende gli obiettivi *start*, *connectArtifact* e *sensingState*.

### 3 Comportamento

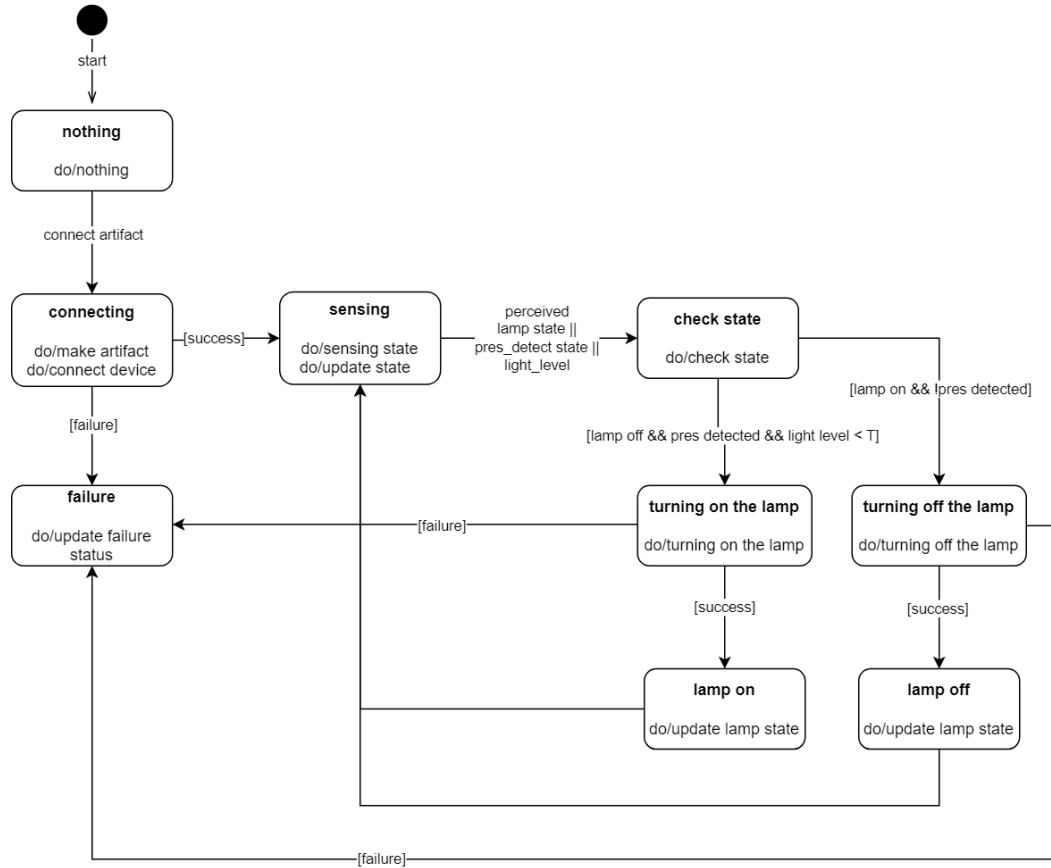


Figura 4: Comportamento del sistema

Il comportamento del sistema, come visibile in figura 4, è descritto come segue:

1. Una volta che sono stati creati tutti gli agenti e gli artefatti ed è stato formato il gruppo seguendo le specifiche dell'organizzazione, il sistema inizierà ad funzionare. Tutti gli agenti come primo passo, saranno obbligati ad eseguire l'obiettivo *start*.
2. Gli agenti *lamp\_thing*, *light\_thing* e *pres\_detect\_thing* eseguiranno la missione *mThing*. Il primo obiettivo della missione *connectArtifact* consiste nella creazione e della connessione con gli artefatti *proxy*.

3. Se la connessione al dispositivo fallisce, viene comunicato l'evento all'agente *smart\_room*, il quale si occupa di aggiornare lo stato della stanza e di mostrarlo nell'applicazione della *smart\_building* tramite l'artefatto *SmartBuildingGUI*.
4. Una volta connesso, l'agente si registra alle proprietà osservabili in modo che l'agente può ricevere informazioni in tempo reale sullo stato del dispositivo e reagire di conseguenza.
5. Successivamente, gli agenti *lamp\_thing*, *light\_thing* e *pres\_detect\_thing* andranno a chiedere i valori dei dispositivi seguendo l'obiettivo *sensingState* e mandano un messaggio di tipo *achieve* all'agente *smart\_room*.
6. Ogni volta che l'agente *smart\_room* riceve un messaggio di aggiornamento dei dati, l'agente valuta le politiche di controllo della luce e, se necessario, invia un messaggio di accensione o spegnimento all'agente *lamp\_thing*. Viene aggiornato il proprio stato interno ed anche lo stato della stanza nell'applicazione della *smart\_building*.

## 4 Applicazione

Una volta che sono stati avviati tutti i servizi delle stanze di una struttura, è possibile visualizzare le stanze associate tramite l'applicazione della *smart-building*, come illustrato in figura 5. Per ogni stanza è mostrato il suo stato, lo stato della luce e se ci sono persone all'interno della stanza. Inoltre, è possibile anche spegnere o accendere la luce di ogni stanza direttamente dall'interfaccia attraverso i pulsanti presentati.

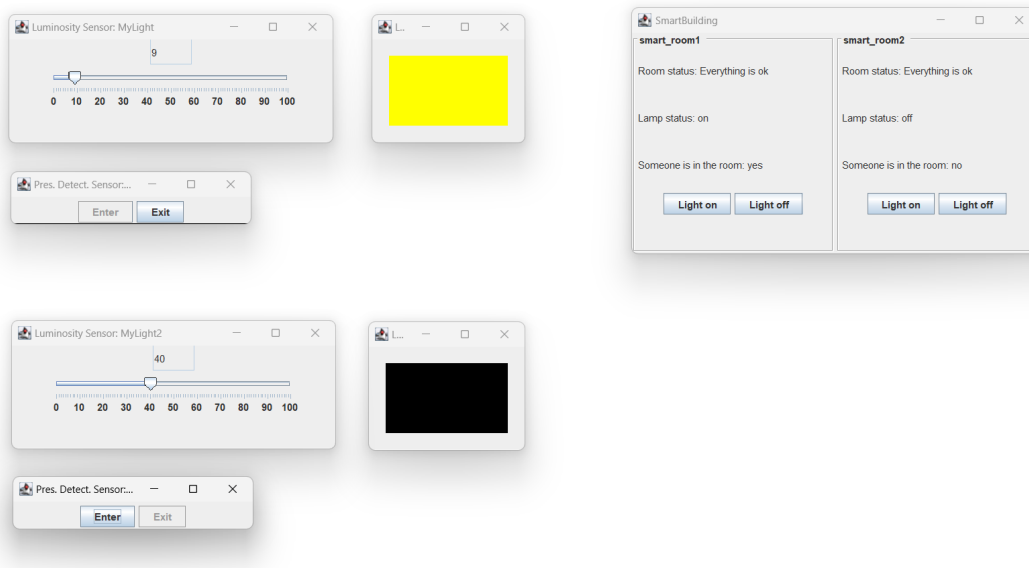


Figura 5: Applicazione della *smart-building*



Se un servizio non è connesso alla *smart-building*, allora nell'applicazione viene mostrato un messaggio che segnala l'errore di connessione. Nella figura 6 viene mostrato il messaggio di errore di connessione al servizio della lampada.

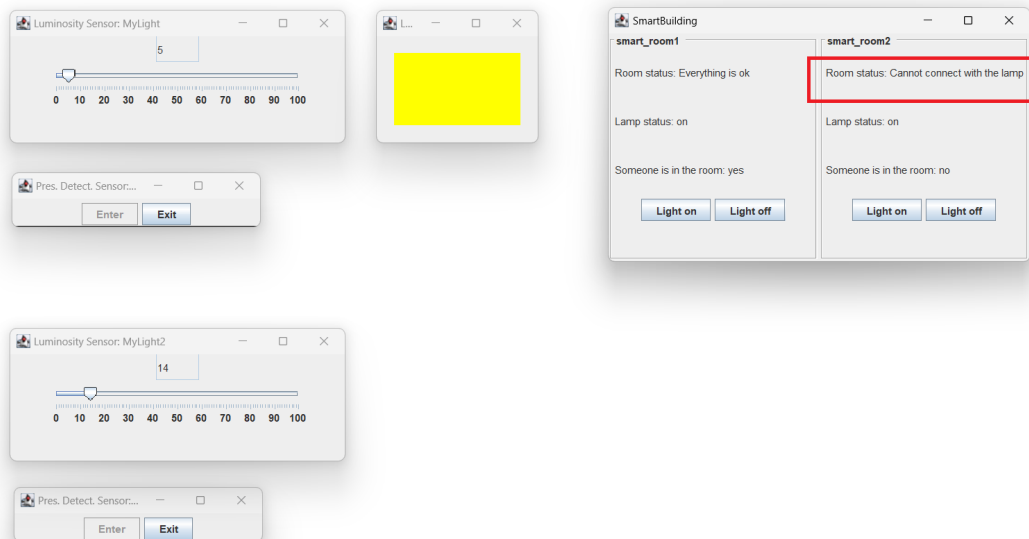


Figura 6: Messaggio di errore di connessione per il servizio della lampada

Nel caso in cui si voglia accendere o spegnere le luci, ma il servizio della lampada non è attualmente operativo per qualche motivo, nell'applicazione verrà visualizzato un messaggio di errore che indica che l'operazione non è riuscita, come nell'esempio mostrato in figura 7.

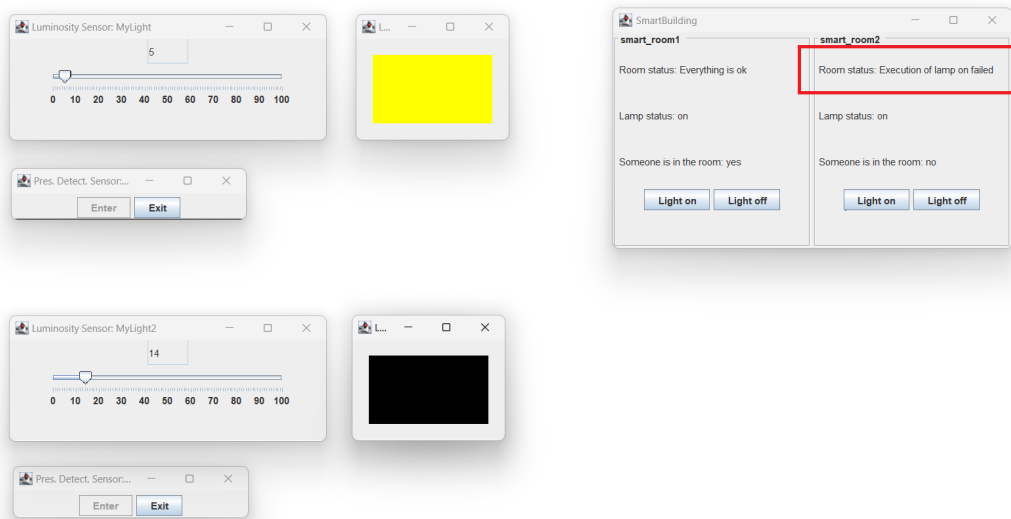


Figura 7: Messaggio di errore in caso di problemi nell'esecuzione dell'operazione di accensione delle luci

Quando il problema è stato risolto e le luci tornano a funzionare, allora anche lo stato della stanza torna alla normalità.

## Riferimenti bibliografici

- [1] Olivier Boissier, Rafael H. Bordini, Jomi Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747–761, June 2013. Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.
- [2] Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. Moise tutorial (for moise 0.7). <https://moise.sourceforge.net/doc/tutorial.pdf>. Accessed January 2022.
- [3] JaCaMo. Jacamo project — multi-agent programming framework. <https://jacamo.sourceforge.net/>. Accessed December 2022.