

Domotic Room Smart City Process Report Template

Nicola Casadei, Marco Benedetti, and Enrico Benini

Alma Mater Studiorum – University of Bologna
via Venezia 52, 47023 Cesena, Italy
`{nicola.casadei4, marco.benedetti7, enrico.benini5}@studio.unibo.it`

Table of Contents

Domotic Room Smart City Process Report Template	1
<i>Nicola Casadei, Marco Benedetti, and Enrico Benini</i>	
1 Introduzione	4
2 Visione	4
3 Obbiettivi	4
4 Requisiti	5
5 Acquisto Hardware	5
5.1 Dispositivi di Computazione	5
5.2 Sensori	6
5.3 Hardware Aggiuntivo	6
6 Analisi dei Requisiti	7
6.1 Casi D'Uso	7
6.2 Scenario	8
6.2.1 Inserimento Range	8
6.2.2 Visualizzazione Stato Realtime	8
6.2.3 Visualizzazione Dati	8
6.3 Modello del Dominio	9
6.3.1 Sistema Embedded	10
6.3.2 Server	13
6.3.3 Web Site	15
6.4 Piani di Test	15
6.4.1 Sistema Embedded	15
6.4.2 Server	15

6.4.3 Web Site	16
7 Problem Analysis	16
7.1 Logic architecture	16
7.2 Abstraction gap	16
7.3 Risk analysis	16
8 Work Plan	16
9 Project	16
9.1 Structure	16
9.2 Interaction	16
9.3 Behavior	16
10 Implementation	16
11 Testing	16
12 Deployment	16
13 Maintenance	16

1 Introduzione

Questo è il template di progetto del corso di smart city dell'università di Bologna. Di seguito saranno consultabile tutto il processo di analisi del progetto: modelli, problemi riscontrati e soluzioni adottate, interazione con l'ambiente, sensori utilizzati e il loro collegamento ...

Per qualsiasi dubbio in merito fare riferimento agli autori.

2 Visione

La visione che guida questo progetto è quella di raggiungere rapidamente l'ideale di città intelligente, quindi con un ambiente aumentato, capace di prendere decisioni e agire tempestivamente per far fronte a casi specifici e capace di comunicare direttamente con chi si trova immerso in esso, per facilitarne la vita di tutti i giorni.

In particolare vogliamo prepararci, imparare a modellare e costruire sistemi che si integreranno in questo contesto, visto l'andamento stesso del mercato che sta sempre più rendendo disponibili risorse di elaborazione e sensoristica a minor prezzo.

3 Obbiettivi

Lo scopo del progetto è quello di implementare concretamente un'applicazione di domotica. Affrontando quindi tutte le problematiche ad essa annesse e fornire una possibile soluzione a queste. Ci auguriamo che questa possa essere di spunto per applicazioni simili e che possa quindi favorirne lo sviluppo.

Sfruttando questo progetto, vogliamo esplorare e apprendere la teoria e i concetti affrontati nel corso di smart city. Quindi tutti gli aspetti riguardanti la gestione di sensori e input provenienti dall'ambiente esterno. Uscendo dalla, tipica, zona di confort classica dei sistemi software.

4 Requisiti

Si vuole monitorare lo stato ambientale di una stanza. In particolare si vogliono monitorare lo stato di: luce, temperatura e movimento, mantenendo la possibilità di aggiungere altre tipologie di sensori.

Il sistema dovrà dare all'utente la possibilità di inserire attraverso un'interfaccia web, per ogni valore misurato, un'apposito range che indichi i valori ammessi all'interno della stanza in modo che, se uno dei valori misurati non è conforme alle specifiche, venga indicata una notifica di allarme sull'interfaccia stessa. Questo con l'idea di simulare la possibilità di eseguire delle azioni collegate all'allarme (ad esempio, accensione delle luci o del riscaldamento)

L'utente potrà inoltre visualizzare all'interno del sito i valori misurati in tempo reale e il valore dei vari sensori nel tempo, potendone quindi consultare la storia.

5 Acquisto Hardware

Sfortunatamente il primo problema che si è incontrato in un progetto come il seguente è stato la necessità di acquistare la parte hardware del sistema che si andrà a costruire. Di conseguenza si è messo in atto un processo di ricerca dei sensori, cavi e quant'altro per riuscire a soddisfare i requisiti

5.1 Dispositivi di Computazione

Prima di tutto necessitiamo di un dispositivo in grado di computare i dati emessi dai vari sensori e che sia interamente programmabile. Nel corso abbiamo visto due possibilità che hanno avuto molto successo recentemente:

- Arduino
- Raspberry Pi

Noi abbiamo scelto la seconda opzione perché abbiamo più familiarità con il dispositivo e perché risulta più facile il riutilizzo dello stesso una volta terminato questo progetto.

Costo del dispositivo: 44,50 €

5.2 Sensori

Un'altra cosa fondamentale riguarda i sensori necessari per catturare i parametri richiesti. Abbiamo Quindi scelto i seguenti sensori

Parametri Ambientali	Sensori Costo
Temperatura	
Luce	
Movimento	

Table 1. Sensor Table

5.3 Hardware Aggiuntivo

Hardware Costo
Breadboard
Wires
Resistors

Table 2. Addictional Hardware

6 Analisi dei Requisiti

6.1 Casi D'Uso



Fig. 1. Casi d'Uso

Nell'immagine sopra si possono vedere quali sono le macro operazioni principali effettuate dal sistema e le interazioni con l'esterno. In particolare gli attori che interagiscono con il sistema saranno:

- La stanza: con questo attore si intendono i vari parametri che si possono rilevare attraverso i sensori e che quindi saranno di input per il sistema.
- Utente: con questo attore rappresenta l'utente che può interagire con il sistema.

Il sistema è stato volutamente suddiviso in tre parti distinte con l'idea di seguire un modello MVC dove però la parte di modello non viene aggiornato solamente attraverso l'input inserito dall'utente, ma anche e soprattutto dall'input dei sensori. L'organizzazione hardware ha fortemente influito su questa suddivisione.

È inoltre possibile visualizzare le macro operazioni che devono essere effettuate e modellate dal sistema. Si vedano gli scenari di seguito per avere

una piú dettagliata visualizzazione dell'interazione tra le varie parti che lo schema sovrastante vuole rappresentare.

6.2 Scenario

In questa sezione verranno illustrati le principali modalità di utilizzo del sistema. Gli scenari elencati di seguito riguardano l'utente e di conseguenza si prevede l'accesso da parte di questo all'interfaccia di input.

Si prevede che il sistema sia opportunamente configurato e settato a livello hardware, senza errori durante la fase di start up.

6.2.1 Inserimento Range

1. Attraverso un apposito menú l'utente é in grado di accedere alla funzionalità di settaggio dei range associati ai parametri ambientali.
2. Il server sa già i sensori che sono collegati al raspberry appena questi inviano qualche dato e quindi l'utente é in grado di visualizzare i controlli relativi ad ogni tipologia di sensore attualmente connesso. Conseguentemente l'utente é in grado di modificare tali intervalli.
3. Al termine della modifica degli intervalli l'utente dovrà confermare le modifiche attraverso un'apposito pulsante.
4. Il sistema mostra un messaggio di conferma o di errore.

6.2.2 Visualizzazione Stato Realtime

All'accesso del sistema l'utente visualizza lo stato realtime dei valori dei sensori ed eventuali notifiche:

- Se i valori vanno oltre gli intervalli correnti.
- Sullo stato dei sensori, se sono attivi al momento oppure no.

In questa modalità l'utente non può effettuare alcuna operazione.

6.2.3 Visualizzazione Dati

Attraverso un apposito menú l'utente é in grado di accedere alla visualizzazione dei dati storici dei vari sensori.

6.3 Modello del Dominio

In questa sezione vogliamo cercare di modellare le entità inserite all'interno dei requisiti senza fare riferimento alla parte tecnologica e alla parte hardware. In questo modo siamo in grado di decidere noi le interfacce con cui vogliamo lavorare e costruire la parte software aumentando il disaccoppiamento con la parte fisica e quindi aumentando anche la possibilità di utilizzare volendo lo stesso software su diverse configurazioni. In questo progetto ad ogni modo si utilizzerà solamente la configurazione che trovate descritta in questo report.

Suddivisione del Sistema: visto che è emerso già dai casi d'uso la separazione del sistema in varie parti, ci è sembrato giusto iniziare a modellare dividendolo fin da subito in modo da:

- semplificare il processo
- suddividere il lavoro di implementazione successivamente tra i membri del gruppo.

In particolare le parti individuate sono tre:

- Sistema Embedded: che nel nostro caso si occuperà di catturare, convertire e inviare i dati alle altre parti
- Server: Sarà la parte che riceve i dati e si occupa di effettuare le varie elaborazioni come ad esempio, il salvataggio dei dati, il calcolo delle statistiche e il controllo dei range. Infine questa parte dovrà rendere disponibili i risultati alla parte successiva.
- Web site: parte che, prende i risultati dalle parti precedenti e li mostra all'utente rispettando i casi d'uso precedenti.

Chiaramente in questa fase tutto viene semplificato e ridotto a poche entità. Questo però non significa che, ogni entità presente negli schemi sottostanti, non si riveli essere poi a sua volta un sottosistema più complesso.

Lo scopo di questa fase è appunto quella di riflettere, in un primo modello, i requisiti. Poi da questo modello iniziare a sviluppare il progetto cercando di mantenere coerenza nelle interfacce principali che indicano le interazioni più importanti.

6.3.1 Sistema Embedded

Per quanto riguarda il sistema embedded è necessario prima effettuare delle indagini su come interagire e comunicare con i sensori in modo da modellare adeguatamente il tutto e quindi assicurarsi che in seguito sarà semplice riuscire a integrare il tutto con la tecnologia che avremo intenzione di utilizzare. Quindi questa è una piccola eccezione che è necessario fare a questo livello, anche se in particolare si fa riferimento a un paradigma più che ad una tecnologia specifica.

Il modello di interrogazione dei sensori è a polling di conseguenza il nostro modello dovrà riflettere questa modalità di interazione. Sfortunatamente questo implica che a livello di modello già **un'entità attiva** che si occupa di reperire i valori visto, che in una modalità a polling devo esplicitamente chiedere ai sensori i valori.

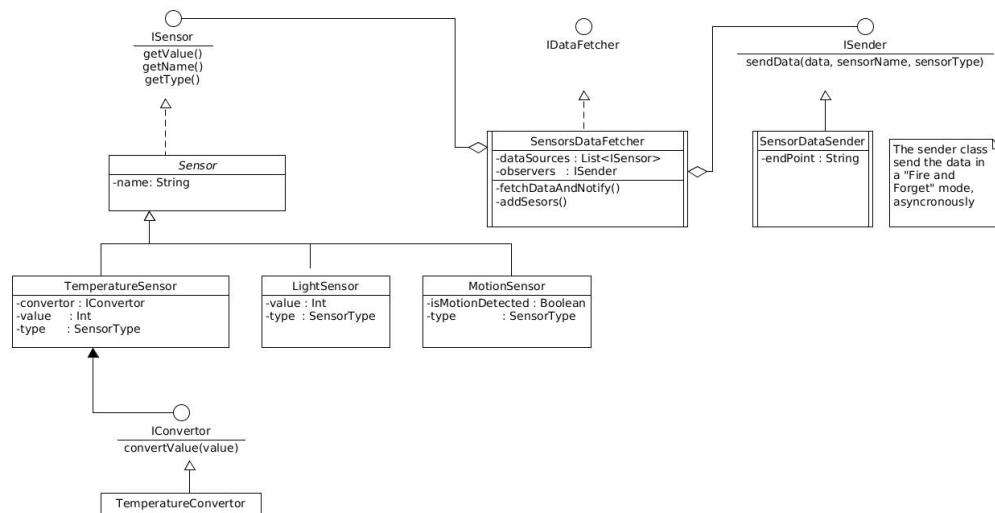


Fig. 2. Sistema Embedded, Struttura

Struttura Dalla struttura si possono subito individuare le entità principali che sono presenti nei requisiti, in particolare la presenza di una specifica gerarchia per i sensori che condividono la stessa interfaccia che consente

agilmente di ottenere il valore corrente del sensore. Sono stati inseriti solamente i sensori citati nei requisiti, ma si può facilmente intuire come qualsiasi tipo di sensore sia facilmente modellabile secondo questa struttura.

Si noti inoltre come viene anche inserita l'entità IConvertor che si occuperà di convertire il valore di uno specifico sensore in un'unità più consona per la sua gestione. Chiaramente questo viene affrontato fin da questo livello perché si immagina l'operazione di conversione come un'operazione quasi istantanea e necessaria.

Infine sono presenti le entità che si occupano, attivamente, di interrogare i sensori ogni intervallo di tempo predefinito e quindi inviarli altrove. In questo caso è stato tutto ridotto ad un singolo endpoint, anche se poi possono essere facilmente più di uno. Come si può visionare nel commento, l'invio dei dati avviene con una metodologia di tipo *fire and forget*, quindi non avvengono reinvii dei dati e vengono ignorati eventuali errori. Chiaramente tutto questo è dovuto all'idea che i cicli di invio siano abbastanza brevi da potersi permettere eventuali perdite.

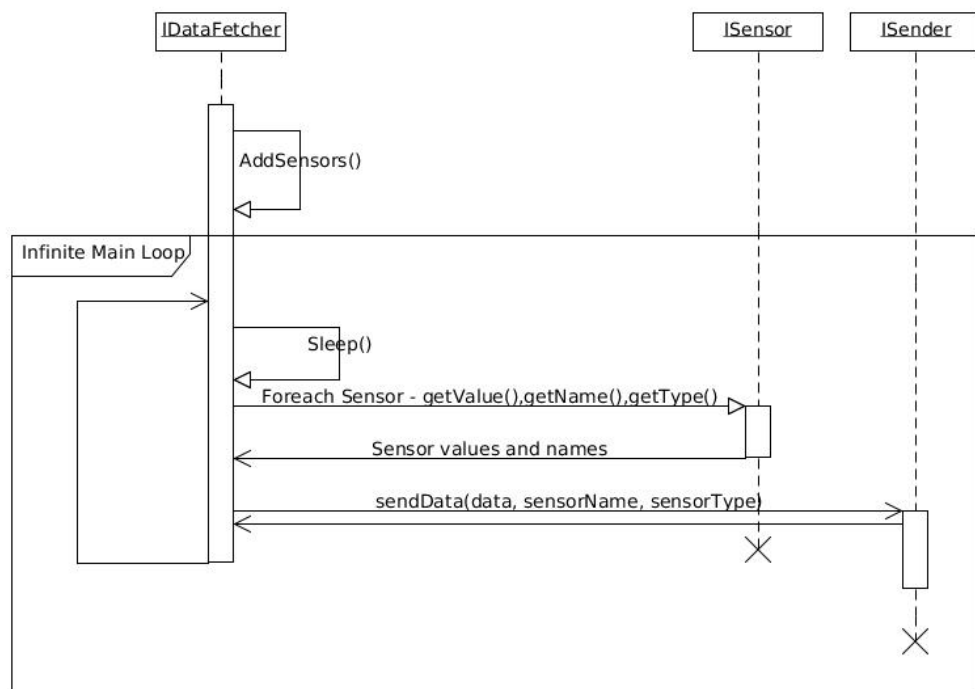


Fig. 3. Sistema Embedded, Interazione

Interazione Prima di iniziare si vuole evidenziare come viene riportato solamente un diagramma di interazione perché é presente solamente un'entità attiva, ma nel futuro potrebbero esserci più task e quindi saranno necessari più diagrammi dell'interazione.

Nello schema di interazione vengono evidenziate le varie fasi del Sistema, in particolare la fase iniziale di setup dove, conoscendo quali sensori sono presenti, questi vengono aggiunti nella memoria dell'entità principale in modo che poi in seguito siano facilmente interrogabili.

Terminata la fase di *Init* inizia il loop infinito che appunto aspetta inizialmente un piccolo lasso di tempo e poi va a interrogare iterativamente tutti i sensori che sono stati aggiunti precedentemente, raccoglie i dati e interroga asincronamente l'entità di invio, per poi ricominciare il ciclo stesso.

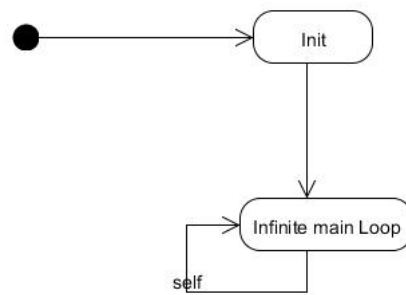


Fig. 4. Sistema Embedded, Comportamento

Comportamento Nel diagramma del comportamento semplicemente viene riportato quanto é stato precedentemente discusso semplicemente attraverso una state machine.

6.3.2 Server

La parte server é quella piú importante di tutto il sistema in quanto é quella che concentra tutta la logica applicativa del sistema stesso.

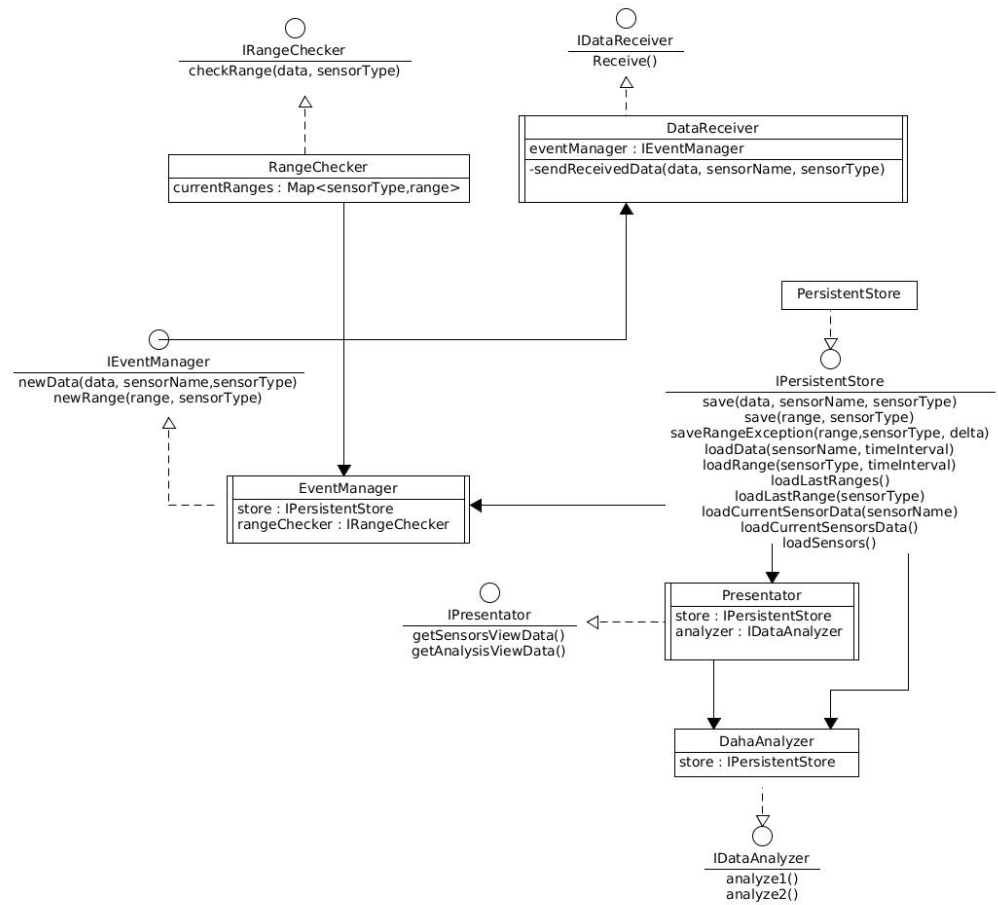


Fig. 5. Server, Struttura

Struttura Le entità principali di questo schema sono:

- IPersistentStore, che si occuperà di salvare opportunamente i dati provenienti dai sensori e dall'utente
- IDataReceiver, che sarà sempre in ascolto di ogni messaggio proveniente dal sistema embedded e quindi notificherà opportunamente il sistema ad ogni nuovo arrivo
- IPresentator che si occuperà di ottenere i dati necessari per le viste da mostrare all'utente quanto una nuova richiesta viene inoltrata.

Chiaramente ogniuna di queste entità é in parte citata nei casi d'uso.

Si veda i diagrammi dell'interazione per i dettagli di come avviene la comunicazione di tutte queste entità al fronte di garantire il funzionamento e il soddisfacimento dei requisiti.

Interazione

Comportamento

6.3.3 Web Site

Dopo un'attenta analisi abbiamo ritenuto superflua la necessità di modellare e analizzare la parte di visualizzazione in quanto non é effettivamente presente nessuna business logic ma solamente la parte che effettua le richieste di dati alle parti precedenti e visualizza tutto su schermo.

La documentazione relativa a questa parte si limiterà solamente a come utilizzare poi il software attraverso l'interfaccia.

6.4 Piani di Test

In questa fase saranno presenti i piani di test che vengono costruiti sulla base dei modelli precedenti, ancora privi di implementazione. Ciò é possibile perché a questo punto abbiamo già individuato le interfacce delle entità principali e le loro operazioni pubbliche.

Chiaramente a questo punto i test non possono funzionare perché manca l'implementazione. Anche in questa fase si divide il tutto sulle 3 parti precedenti.

6.4.1 Sistema Embedded

6.4.2 Server

6.4.3 Web Site

7 Problem Analysis

7.1 Logic architecture

7.2 Abstraction gap

7.3 Risk analysis

8 Work Plan

9 Project

9.1 Structure

9.2 Interaction

9.3 Behavior

10 Implementation

11 Testing

12 Deployment

13 Maintenance

References