



Ricerca di Sistema elettrico

Sviluppo di un sistema di supervisione e controllo BEMS e diagnostica di anomalie su impianti di climatizzazione attraverso algoritmi di data fusion

Fabio Moretti
Stefano Panzieri

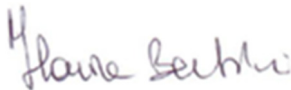
SVILUPPO DI UN SISTEMA DI SERVISIONE E CONTROLLO BEMS E DIAGNOSTICA DI ANOMALIE SU IIMPIANTI DI CLIMATIZZAZIONE ATTRAVERSO ALGORITMI DI DATA FUSION

Fabio Moretti

Stefano Panzieri

Settembre 2014

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico – ENEA
Piano Annuale di Realizzazione 2013
Area: Razionalizzazione e risparmio nell'uso dell'energia elettrica
Progetto: Risparmio di energia elettrica nei settori: civile, industria e servizi
Obiettivo B : Gestione di reti di edifici e "smart village"
Responsabile del Progetto: Ing. Ilaria Bertini, ENEA



Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "Implementazione di un sistema di data fusion per la diagnostica di anomalie sull'impianto termico dell'edificio F40 e implementazione del modulo di supervisione di un BEMS"

Responsabile scientifico ENEA: Dr. Stefano Pizzuti



Responsabile scientifico Università Roma Tre : Prof. Ing. Stefano Panzieri

Sommario

INDICE DELLE FIGURE.....	3
1 INTRODUZIONE	4
2 SISTEMA DI SUPERVISIONE E CONTROLLO BEMS	5
2.1 STATO DELL'ARTE PIATTAFORMA SCADA	5
2.2 SISTEMA DI ATTUAZIONE DEI TERMOSTATI	7
2.3 DIAGNOSTICA TERMICA	13
3 CONCLUSIONI E SVILUPPI FUTURI.....	17

Indice delle figure

Figura 1. Struttura logica della Piattaforma SCADA	5
Figura 2. Database della Piattaforma	7
Figura 3. Schema del sistema di interfacciamento piattaforma-edificio F40.....	8
Figura 4. Set point termostati in inverno	8
Figura 5. Control BooleanLogics	9
Figura 6. Threshold Control Action.....	9
Figura 7. Set PointsProfile	9
Figura 8. HistorianActuatorActions	9
Figura 9. Historian Control Actions.....	9
Figura 10. Diagramma di flusso algoritmo di controllo dei termostati	10
Figura 11. Stored Procedure impostazione set point sulle stanze attive	11
Figura 12. Stored Procedure impostazione set point sulle stanze inattive	11
Figura 13. Stored Procedure impostazione set point su tutte le stanze	11
Figura 14. Diagramma UML della classe dedicata al controllo della climatizzazione	12
Figura 15. JavaDoc dell'applicativo.....	12
Figura 16. Impostazione dinamica set point temperatura di stanza.....	13
Figura 17. Grafico della causa C2.....	15

1 Introduzione

Nell'ambito della realizzazione del dimostrativo dello "Smart Village" nel Centro Ricerche ENEA Casaccia è stato realizzato un sistema di supervisione e attuazione di strategie di ottimizzazione, inoltre, sono state definite regole di diagnostica termica ed applicate all'edificio F40. Il rapporto tecnico fa riferimento PAR 2013 sulla Ricerca di Sistema Elettrico, in particolare al **"Sviluppo di un sistema di supervisione e controllo BEMS e diagnostica di anomalie su impianti di climatizzazione attraverso algoritmi di data fusion"** ed è realizzato dal Dipartimento di Informatica ed Automazione dell'Università degli Studi "Roma Tre".

Il sistema di supervisione BEMS è un'evoluzione della piattaforma SCADA realizzata nelle precedenti annualità. La piattaforma è stata implementata per l'acquisizione dati eterogenei tra loro e l'attuazione di comandi di controllo: mentre l'aspetto dei moduli di acquisizione è consolidato, la realizzazione delle logiche di controllo si è limitato alla predisposizione all'acquisizione di comandi dall'esterno ed alla realizzazione di controlli riguardanti l'illuminazione indoor, basati su scheduling prefissati. Nell'attività di quest'anno l'attenzione è stata focalizzata sulla realizzazione del sistema di controllo della climatizzazione delle stanze dell'edificio F40. In particolare tale sistema è stato implementato in modo tale da interfacciarsi in maniera trasparente con i tool di ottimizzazione utilizzati nel C.R. ENEA (es. MATLAB) e permette la regolazione dei termostati delle stanze secondo profili prestabiliti dal processo di ottimizzazione, permettendo quindi una gestione più efficiente della climatizzazione interna. Tale sistema inoltre si predispone ad una futura applicazione dell'approccio demand/response.

Le regole di diagnostica termica hanno riguardato l'identificazione di guasti relativi alla centrale termica, in particolare il timer, il sistema di controllo della centrale e la valvola di miscelazione del fluido di mandata. Tali regole sono state realizzate, come per la diagnostica elettrica, secondo la logica fuzzy.

2 Sistema di supervisione e controllo BEMS

Il sistema di supervisione e controllo BEMS è l'integrazione della piattaforma SCADA con il sistema di controllo avanzato. Nel documento viene inizialmente mostrato lo stato di sviluppo della piattaforma e le migliorie apportate e successivamente l'implementazione dell'algoritmo di controllo dei termostati e dei fancoil utilizzato nell'edificio F40.

2.1 Stato dell'arte Piattaforma SCADA

La struttura logica della piattaforma è rimasta invariata ed è mostrata nella Figura 1:

- **Presentation Layer.** Interfaccia l'utente finale con la piattaforma tramite web application (servlet, html, jsp).
- **ApplicationLayer.** Esegue l'attività di business intelligence della piattaforma Smart Town. In tale strato vengono eseguiti i processi di analisi e fusione dei dati, del situation assessment per la diagnostica degli edifici, di realizzazione di logiche di controllo, di gestione di allarmi ed anomalie.
- **Data Layer.** Strato relativo al datawarehouse, ovvero lo storage dei dati. E' stato realizzato un database dalla struttura modulare cui possano confluire dati di diversa natura provenienti da diverse sorgenti.
- **Sensor/ActuatorLayer.** Strato contenente i moduli di interfacciamento con i sensori. Ogni modulo è dedicato ad una specifica categoria di sensori ed è completamente indipendente dagli altri nella modalità di comunicazione e di interfacciamento.

Per ulteriori approfondimenti si rimanda al report RdS/2012/230 *"Realizzazione di una piattaforma integrata per il data fusion di segnali provenienti da sistemi sensoriali per applicazioni di smart city integrate nella rete della pubblica illuminazione"*.



Figura 1. Struttura logica della Piattaforma SCADA

A livello progettuale il progetto è stato rivisto rispetto alla versione precedente e suddiviso in 5 sottoprogetti:

- **Communication.** Java Application. Dedicato alla gestione della comunicazione tra i moduli ed il DB, alla gestione dei log di errore ed alla definizione delle variabili statiche condivise.
- **DataAcquisition.** Java Application. Dedicato esclusivamente all'interfacciamento con i sensori o i BEMS per l'acquisizione dei dati e l'inserimento nel DB.
- **BuildingDiagnostics.** Java Application. Dedicato all'attività di diagnostica avanzata degli edifici.
- **BuildingControl.** Java Application. Dedicato all'attuazione dei comandi di controllo.
- **ServerSmartTown.** Java Web Application. Dedicato all'interfacciamento web ed alla erogazione di servizi web tramite apposite servlet e web services.

Gli interventi maggiori sono stati effettuati a livello di struttura di database. Sebbene l'engine MySQL sia rimasto invariato, è stata ristrutturata l'organizzazione delle tabelle e ne sono state introdotte di nuove per venire incontro alle esigenze delle attività di diagnostica ed attuazione. La Figura 2 mostra parte della ristrutturazione del database (per esigenze di leggibilità è stata omessa la struttura riguardante l'acquisizione dati, si rimanda al RdS/2012/230 per ulteriori approfondimenti riguardanti tale aspetto), in cui le tabelle vengono raggruppate secondo il task cui fanno riferimento. La tabella Scada è il cuore del DB, contenente le informazioni relative ai sensori installati insieme alla tabella building, cui fanno riferimento i task di diagnostica, attuazione e KPI.

Diagnostics contiene il numero più consistente di tabelle, la struttura portante è costituita da preprocessing, processing, situation e causes, correlate tra loro da tabelle di congiunzione. Tale struttura rispecchia la suddivisione della fase di diagnostica nelle tre fasi principali della data fusion (cfr. 2.3). Ogni fase ha la propria tabella dedicata alla storicizzazione degli eventi rilevati. La tabella di preprocessing è utilizzata per attuare delle eventuali operazioni al fine di rendere omogenei i dati in fase processing.

Actuation contiene le tabelle per la gestione delle logiche di attuazione: thresholds, logiche di confronto e scheduling. Inoltre è stato introdotto uno storico delle attuazioni sia ad alto livello (attuazione del profilo giornaliero dei termostati), sia a livello di singolo sensore.

KPI e Alarms sono gruppi di tabelle tuttora in fase di evoluzione in cui vengono gestiti gli allarmi e le analisi globali sulle performance degli edifici.

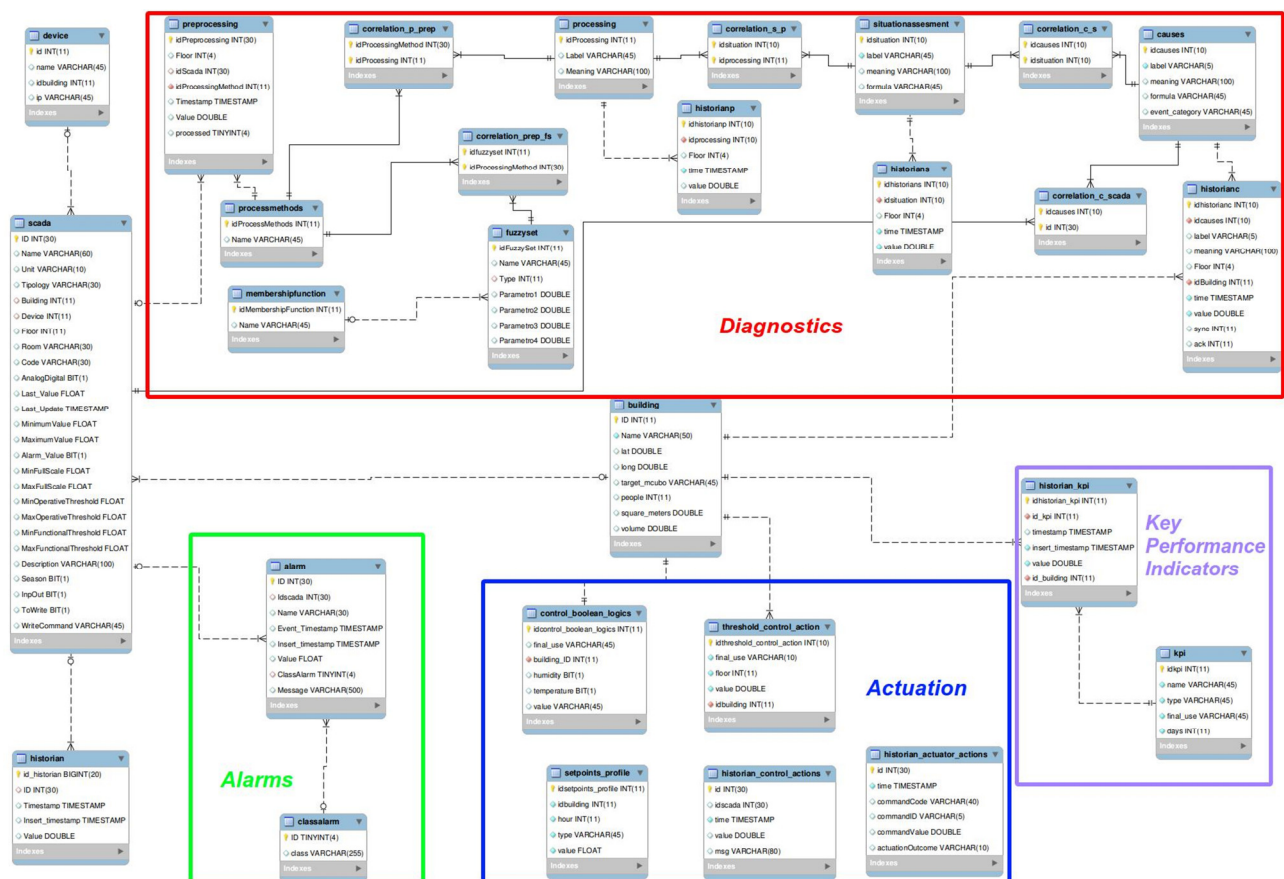


Figura 2. Database della Piattaforma

2.2 Sistema di attuazione dei termostati

L'applicativo di regolazione dei termostati dell'edificio F40 è inserito nel business intelligence layer della piattaforma, agisce sulla regolazione dei setpoint dei termostati di stanza dell'edificio e si interfaccia con il datawarelayer per l'acquisizione dati.

L'applicativo, come il resto dei progetti, è stato realizzato in Java e comunica con il Building Energy Management System (BEMS) tramite RESTful Web Services (WS) come mostrato nella Figura 3.

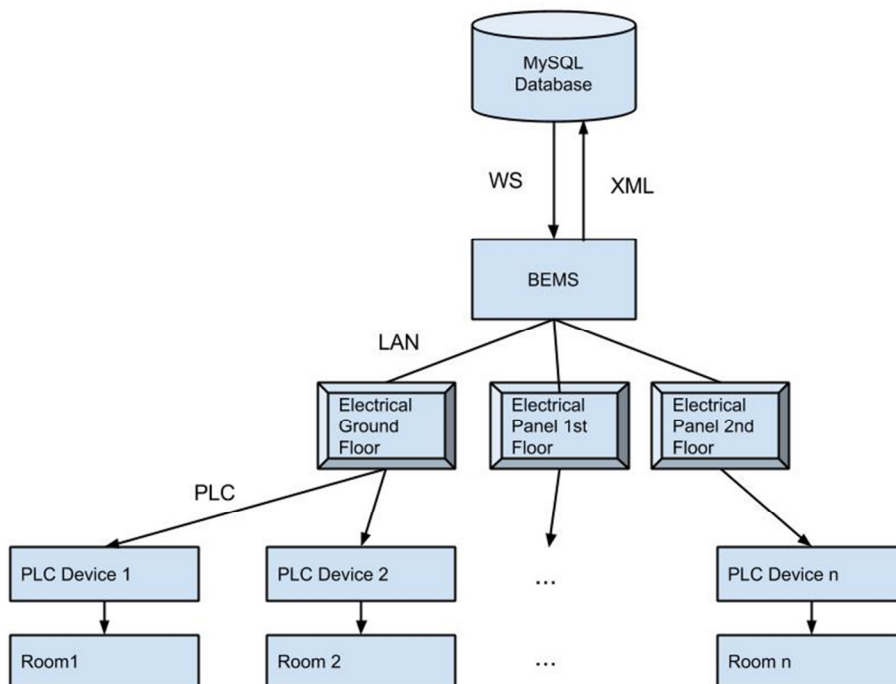


Figura 3. Schema del sistema di interfacciamento piattaforma-edificio F40

Il BEMS successivamente tramite Power Line Communication (PLC) instrada i comandi di attuazione ai device situati in ogni stanza. La Figura 4 mostra il profilo giornaliero dei termostati invernali attualmente utilizzato nell’edificio F40.

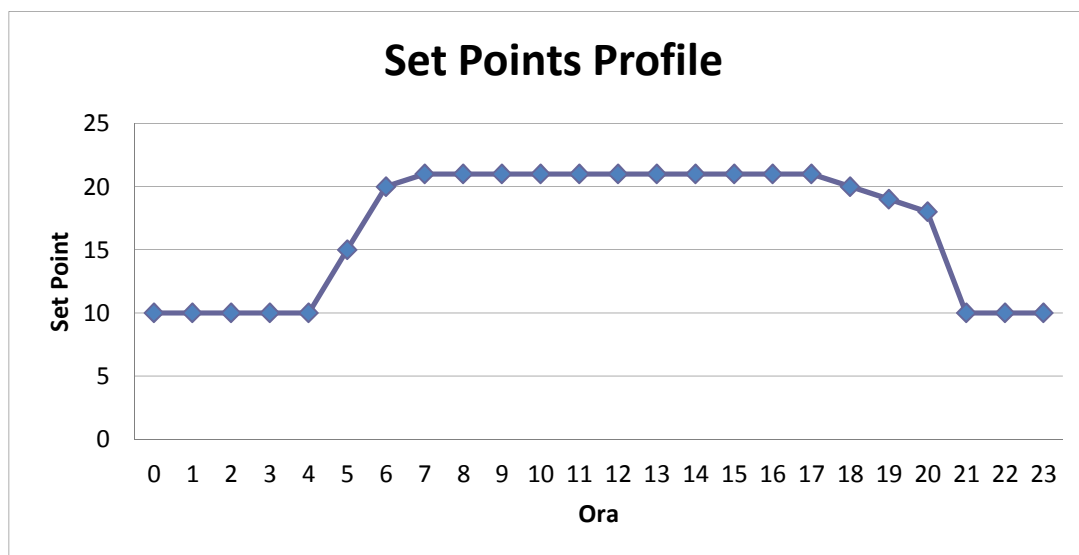


Figura 4. Set point termostati in inverno

Entrando nel dettaglio, di seguito sono mostrati i campi delle tabelle dedicate all’attuazione. La Figura 5 mostra la tabella **Control Boolean Logics**, contenente le logiche con cui devono essere gestita l’impostazione dei setpoint: per ogni final_use , in questo caso specifico set_point_termostato, vengono definite le variabili che determinano l’impostazione del valore, attualmente temperatura ed umidità interna, misurata in un punto rappresentativo dell’edificio. Il valore dei campi delle variabili è booleano, e

sta ad indicare il valore che il set point deve assumere quando ci si trova al di sotto (0) o al di sopra (1) della soglia di quella variabile specifica, definita nella tabella **Threshold Control Actions** (Figura 6). Tale tabella contiene, per ogni edificio ed ogni piano, i valori di soglia da prendere in considerazione. La Figura 7 mostra la tabella dedicata allo scheduling di controllo, utilizzata per il controllo statico orario giornaliero dei set point dei termostati. La Figura 8 e la Figura 9 mostrano rispettivamente lo storico delle attuazioni impartite sui sensori e quelle impartite dal supervisore che sia automatizzato o meno.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idcontrol_boolean_logics	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
final_use	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
idBuilding	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
humidity	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
temperature	BIT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
value	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 5. Control BooleanLogics

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idthreshold_control_action	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
final_use	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
floor	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
value	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
idbuilding	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'

Figura 6. Threshold Control Action

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idsetpoints_profile	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
idbuilding	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
hour	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
type	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
value	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Figura 7. Set PointsProfile

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
idhistorian_actuator_actions	INT(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
time	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
commandCode	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
commandID	VARCHAR(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
commandValue	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
actuationOutcome	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 8. HistorianActuatorActions

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
historian_control_actions	INT(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
idscada	INT(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
time	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
value	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
msg	VARCHAR(80)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Figura 9. Historian Control Actions

L'applicativo è stato implementato in Java, è un thread inserito all'interno del progetto Building Control descritto precedentemente e fa uso di Stored Procedures, implementate in SQL, presentate successivamente.

La Figura 10 mostra il diagramma di flusso dell'algoritmo: dopo una fase di inizializzazione viene processato un check nel database e viene verificato se i dati di presenza del personale sono stati immagazzinati o meno. Tali dati, infatti, vengono acquisiti interfacciandosi con il sistema di badge del C.R. Casaccia e deve sottostare ai tempi di processamento di tale sistema, per cui non è garantita la presenza dei dati nei tempi

richiesti dall’algoritmo di controllo. Nell’eventualità che tali dati non siano stati immagazzinati viene applicato un profilo di default per tutte le stanze, altrimenti, vengono controllati i dati di temperatura ed umidità per verificare se ci troviamo in una giornata con condizioni meteo anomale rispetto alla stagione o meno: nel caso in cui ci troviamo di fronte ad una giornata ordinaria viene seguito un determinato profilo giornaliero, altrimenti viene applicata per quella determinata fascia oraria un set point adattato a tale caso specifico. Tale set point viene applicato solo alle stanze attive, ovvero quelle in cui è presente all’interno almeno un dipendente. Essendo una presenza basata sul badge, la climatizzazione della stanza viene mantenuta anche qualora il dipendente si allontani per tempi più o meno brevi dalla sua postazione.

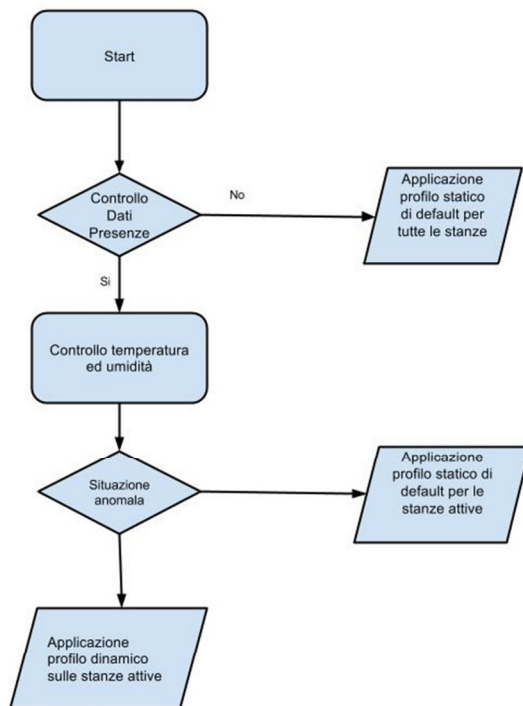


Figura 10. Diagramma di flusso algoritmo di controllo dei termostati

Mentre la logica di gestione del controllo da applicare è realizzata in Java, l’impostazione dei set point nelle stanze attive è delegata alle Stored Procedures (SP), la Figura 11 mostra la SP dedicata all’attuazione dei set point solo sulle stanze attive dell’F40, la Figura 12 mostra la SP relativa alle stanze inattive mentre la Figura 13 mostra la SP chiamata quando non vengono rilevati dati di presenza.

```

DDL: 1 -----
2 -- Routine DDL
3 -- Note: comments before and after the routine body will not be stored by the server
4 -----
5 DELIMITER $$
6
7 • CREATE DEFINER='root'@'localhost' PROCEDURE `set_point_insert_active_rooms`
8 (IN ToWrite_in bit, IN Last_value_in double, IN season_in bit, IN date_in VARCHAR(12), IN hour_in VARCHAR(12))
9 BEGIN
10 DROP TABLE IF EXISTS Temp;
11 CREATE TABLE Temp as SELECT sc.id as id, st.stanza as stanza
12 FROM building.stanzeattive st, smarttowndb.scada sc
13 where st.stanza = sc.room
14 and st.data = date_in
15 and st.ora = hour_in
16 and name like '%Set_point%'
17 and input = 1
18 and building = 1
19 and floor > -1;
20
21 update smarttowndb.scada set towrite = ToWrite_in, Last_Value = Last_value_in, season = season_in
22 where id IN (SELECT id FROM Temp)
23 or (
24 room not in (SELECT codice FROM building.stanze)
25 and name like '%Set_point%'
26 and input = 1);
27
28 INSERT INTO historian_control_actions (idscada,value,msg)
29 SELECT id,Last_value,CONCAT('SetPoint Fancoil of Room ',room,' set to ', Last_value)
30 FROM smarttowndb.scada WHERE ToWrite = 1 and InpOut = 1 and season = season_in;
31
32 END

```

Figura 11. Stored Procedure impostazione set point sulle stanze attive

```

DDL: 1 -----
2 -- Routine DDL
3 -- Note: comments before and after the routine body will not be stored by the server
4 -----
5 DELIMITER $$
6
7 • CREATE DEFINER='root'@'localhost' PROCEDURE `set_point_insert_non_active_rooms`
8 (IN ToWrite_in bit, IN Last_value_in double, IN season_in bit)
9 BEGIN
10 update smarttowndb.scada set towrite = ToWrite_in, Last_Value = Last_value_in, season = season_in
11 where id NOT IN (SELECT id FROM Temp)
12 and room IN (SELECT codice FROM building.stanze)
13 and name like '%set_point%'
14 and input = 1
15 and building = 1
16 and floor > -1;
17
18 INSERT INTO historian_control_actions (idscada,value,msg)
19 SELECT id,Last_value,CONCAT('SetPoint Fancoil of Room ',room,' set to ', Last_value)
20 FROM smarttowndb.scada WHERE ToWrite = 1 and InpOut = 1 and season = season_in;
21
22 END

```

Figura 12. Stored Procedure impostazione set point sulle stanze inattive

```

DDL: 1 -----
2 -- Routine DDL
3 -- Note: comments before and after the routine body will not be stored by the server
4 -----
5 DELIMITER $$
6
7 • CREATE DEFINER='root'@'localhost' PROCEDURE `set_point_insert_all`(IN ToWrite_in bit, IN Last_value_in double, IN season_in bit)
8 BEGIN
9 update smarttowndb.scada set towrite = ToWrite_in, Last_Value = Last_value_in, season = season_in
10 where name like '%set_point%'
11 and input = 1
12 and building = 1
13 and floor > -1;
14
15 INSERT INTO historian_control_actions (idscada,value,msg)
16 SELECT id,Last_value,CONCAT('SetPoint Fancoil of Room ',room,' set to ', Last_value)
17 FROM smarttowndb.scada WHERE ToWrite = 1 and InpOut = 1 and season = season_in;
18
19 END

```

Figura 13. Stored Procedure impostazione set point su tutte le stanze

La Figura 14 mostra nel dettaglio il diagramma UML dell'applicativo mentre la Figura 15 la relativa documentazione. Nel diagramma cui viene esplicitata la dipendenza dalla classe astratta Thread, necessaria

al fine di rendere tale classe un applicazione indipendente dalle altre (ad esempio quella relativa alla regolazione delle luci) all'interno del package BuildingControl. La classe include anche DataBaseManager, utilizzata per la gestione della connessione al database ed al caricamento dei relativi driver. La classe DateTime viene utilizzata per la gestione dell'orario e del calendario, mentre le altre sono classi base di Java necessarie per il corretto funzionamento dell'applicazione.

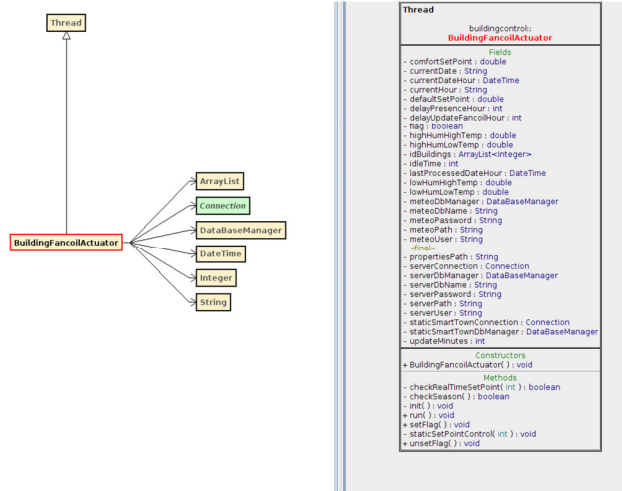


Figura 14. Diagramma UML della classe dedicata al controllo della climatizzazione

Constructors

Constructor and Description
BuildingFancoilActuator()

Method Summary

Modifier and Type	Method and Description
boolean	checkRealTimeSetPoint(int idBuilding) If we are in an anomaly situation, different setPoint are used
boolean	checkSeason() Is it a summer?
void	init()
void	run()
void	setFlag() staticSetPointControl(int idBuilding) Reads from DB a static 24h profile and use it as default
void	unsetFlag()

Methods inherited from class java.lang.Thread
activeCount, checkAccess, clone, countLockWaits, currentThread, destroy, dumpStack, enumerate, getAllStackFrames, getContextClassLoader, getDaemon, getExceptionHandler, getID, getName, getPriority, getStackTrace, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, toString, yield

Methods inherited from class java.lang.Object
equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

BuildingFancoilActuator
public BuildingFancoilActuator()

Method Detail

run
public void run()
Specified by: run in interface java.lang.Runnable
Overrides: run in class java.lang.Thread

staticSetPointControl
public void staticSetPointControl(int idBuilding)
Reads from DB a static 24h profile and use it as default

checkRealTimeSetPoint
public boolean checkRealTimeSetPoint(int idBuilding)
If we are in an anomaly situation, different setPoint are used
Parameters: idBuilding-
Returns: -100 if we are not in anomaly situation, setpoint to be used otherwise

checkSeason
public boolean checkSeason()

init
public void init()

setFlag
public void setFlag()

Figura 15. JavaDoc dell'applicativo

Infine, a titolo di esempio circa la funzionalità del sistema sviluppato, nella figura 16 si riporta l'andamento di una settimana dei set point di una stanza tipo impostati tramite il software implementato e descritto in questo paragrafo.

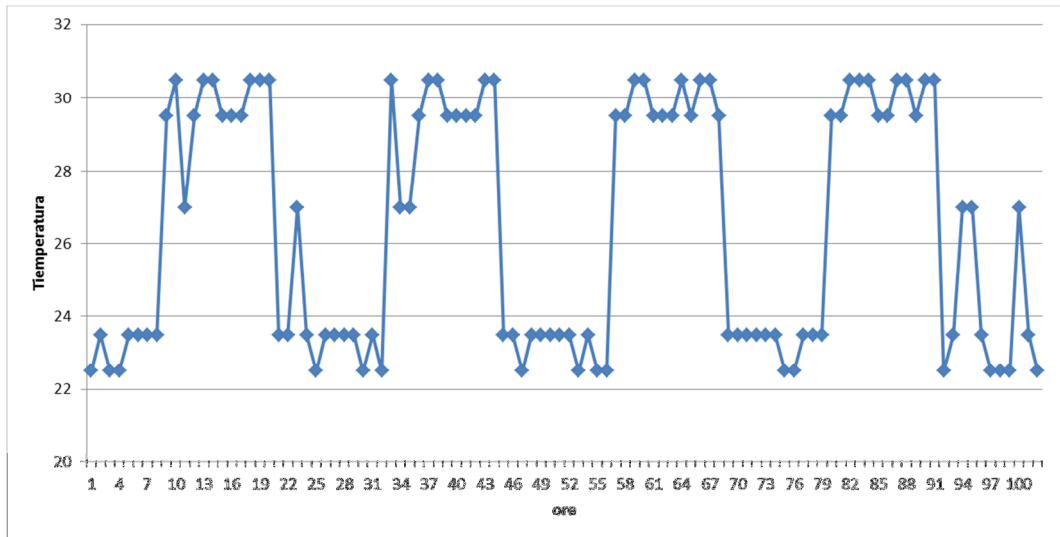


Figura 166. Impostazione dinamica set point temperatura di stanza

2.3 Diagnostica termica

L'attività relativa alla diagnostica termica è incentrata sull'applicazione di regole basate sulla logica fuzzy riguardanti la centrale termica. Tale attività è strettamente correlata alla diagnostica elettrica relativa all'annualità precedente e si basa nuovamente sul principio di suddivisione della diagnostica nella fase di preprocessing, situazioni e cause. La data fusion è ottenuta secondo la già sperimentata logica fuzzy. Le membership functions ed i relativi parametri sono stati scelti a valle di un benchmarking creato appositamente nell'edificio F40, in cui, quando possibile, sono state provocate intenzionalmente delle anomalie.

Le cause prese in esame sono:

- **C1 :Malfunzionamento timer in centrale termica invernale**

Tale causa mira ad individuare un comportamento corretto dell'orologio che regola l'accensione e lo spegnimento dell'impianto di riscaldamento della centrale termica. Le variabili fuzzy utilizzate sono mostrate in Tabella 1: l'orario di lavoro è espresso come funzione gaussiana del tempo, centrata in 12, mentre il consumo termico è espresso come sigmoide.

Tabella 1. Funzioni fuzzy dei preprocessing per la causa C₁

Label	Significato	MembershipFunction	Parametri
P1	Working Time	$e^{-\frac{(x-m)^2}{2s^2}}$	m = 12 s = 5
P2	Consumo termico caldo	$\frac{1}{1 + e^{\frac{(x-c)}{t}}}$	c = 1 t = 0.5

Una volta definiti i preprocessing, sono state identificate le situazioni ad essi correlate, ottenute relazionando i preprocessing secondo la logica booleana di AND,OR,NOT (^, v, -).

- Consumo termico alto al di fuori dell'orario di lavoro: $S_1 = (-P_1 \wedge P_2)$
- Consumo termico basso durante l'orario di lavoro: $S_2 = (P_1 \wedge -P_2)$

La causa risultante è data dall'OR logico delle tre situazioni:

$$C_1 = S_1 \vee S_2$$

• **C2: Scorretta gestione della distribuzione del calore**

Tale anomalia mira ad identificare una scorretta gestione della impianto ed è quindi dipendente anche dall'occupazione dell'edificio, come mostrato nella Tabella 2. La funzione di occupazione non è stata fuzzyficata in quanto già normalizzata.

Tabella 2. Funzioni fuzzy dei preprocessing per la causa C₁

Label	Significato	MembershipFunction	Parametri
P1	Working Time	$e^{\left(\frac{-(x-m)^2}{2s^2}\right)}$	m = 12 s = 7
P2	Percentuale di Occupazione	%	
P3	Consumo termico caldo	$\frac{1}{1 + e^{\left(\frac{(x-c)}{t}\right)}}$	c = 1 t = 0.5

Una volta definiti i preprocessing, sono state identificate le situazioni ad essi correlate:

- Consumo termico alto al di fuori dell'orario di lavoro in assenza di occupazione: $S_1 = (-P_1 \wedge P_2 \wedge -P_3)$
- Consumo termico basso durante l'orario di lavoro in presenza di occupazione: $S_2 = (P_1 \wedge -P_2 \wedge P_3)$

La causa risultante è data dall'OR logico delle tre situazioni:

$$C_2 = S_1 \vee S_2$$

La Figura 177 mostra l'andamento dell'indice della causa C₂: nell'esempio è stato considerato un intervallo di tempo di 24 ore

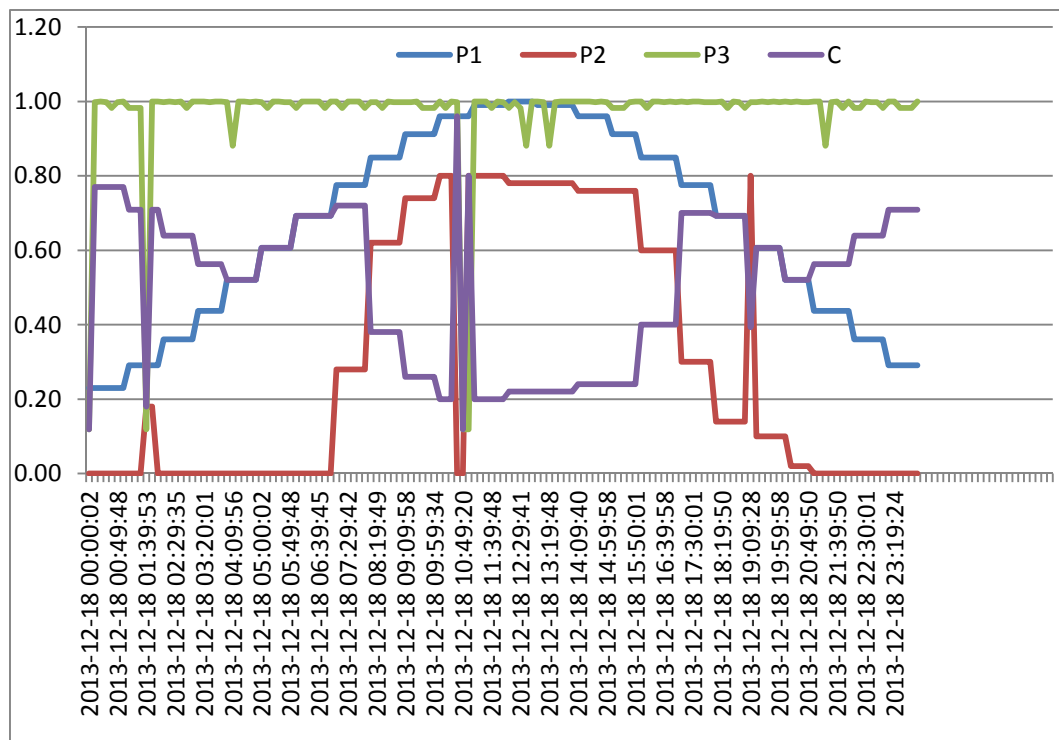


Figura 177. Grafico della causa C2

- **C3: Scorretto funzionamento della valvola miscelatrice del fluido freddo in centrale termica**

Tale causa mira ad identificare uno scorretto funzionamento della valvola di miscelazione del fluido. Per identificare l'anomalia sono stati utilizzati il delta tra il set point della temperatura di mandata impostato e la temperatura effettivamente misurata e la percentuale di chiusura della valvola, come mostrato nella Tabella 3. Il delta è stato fuzzyficato con una sigmoide.

Tabella 3. Funzionifuzzy dei preprocessing per la causa C₃

Label	Significato	MembershipFunction	Parametri
P1	Differenza tra SetPoint della mandata impostato e Temperatura misurata	$\frac{1}{1 + e^{\left(\frac{x-c}{t}\right)}}$	c = 1 t = 0.5
P2	Percentuale di Chiusura della Valvola	%	

Una volta definiti i preprocessing, sono state identificate le situazioni ad essi correlate:

- La valvola non miscela (sulla base del valore percentuale di apertura rilevata) quando la differenza tra la temperatura di mandata e quella di set point è minore di un valore di benchmark: $S_1 = (-P_1 \wedge P_2)$
- La valvola miscela (sulla base del valore percentuale di apertura rilevata) quando la differenza tra la temperatura di mandata e quella di set point è maggiore di un valore di benchmark: $S_2 = (P_1 \wedge P_2)$

La causa risultante è data dall'OR logico delle tre situazioni:

$$C_3 = S_1 \vee S_2$$

- **C4: Staratura del termostato di stanza in periodo estivo**

La staratura del termostato di stanza viene identificata qualora per una determinata zona dell’edificio venga rilevata una differenza sostanziale tra la temperatura interna e quella impostata in una o più stanze. Anche in questo caso il delta di temperatura viene valutato tramite una sigmoide. L’ipotesi che tale comportamento anomalo sia riconducibile ad altre cause (ad esempio un malfunzionamento della ventola del fancoil di stanza) sono state prese in considerazione nella diagnostica della climatizzazione dal punto di vista elettrico.

Tabella 4. Funzioni fuzzy dei preprocessing per la causa C4

Label	Significato	MembershipFunction	Parametri
P1..Pn	Differenza tra Set Point di stanza impostato e Temperatura Interna	$-e^{\left(\frac{x^2}{2s^2}\right)}$	s = 5

La situazione relativa a tale preprocessing è:

- Una o più stanze di una zona termica presentano una temperatura interna maggiore del setpoint impostato: $S_1 = (P_1 \text{ XOR } P_2 \text{ XOR } \dots \text{ XOR } P_n)$

La causa è direttamente relazionata all’unica situazione.

- **C5: Malfunzionamento del sistema di raffrescamento**

Tale causa avviene qualora per una determinata zona in tutte le stanze viene rilevata una differenza tra il set point e la temperatura interna maggiore di una soglia (Tabella 5).

Tabella 5. Funzioni fuzzy dei preprocessing per la causa C5

Label	Significato	MembershipFunction	Parametri
P1..Pn	Differenza tra Set Point di stanza impostato e Temperatura Interna	$-e^{\left(\frac{x^2}{2s^2}\right)}$	s = 5

- *Tutte le stanze di una zona hanno una temperatura maggiore del setpoint impostato:*
 $S_1 = (P_1 \wedge P_2 \wedge \dots \wedge P_n)$

Anche questo caso la causa è direttamente relazionata all’unica situazione.

3 Conclusioni e sviluppi futuri

Nel documento sono state descritte le attività che hanno riguardato nel dettaglio

1. Implementazione di un sistema di data fusion per la diagnostica di anomalie sull'impianto termico dell'edificio F40;
2. Implementazione del modulo di supervisione del Building Energy Management System (BEMS) dell'edificio F40.

Nella prima linea di attività sono state definite nuove regole di diagnostica termica attraverso algoritmi di *"data fusion"* che saranno successivamente implementate nella piattaforma.

Nella seconda è stato realizzato l'upgrade della piattaforma Smart Town e l'implementazione del software di controllo della climatizzazione per l'attuazione real time dei set point di temperatura di stanza attraverso la comunicazione con il BEMS.

Gli sviluppi futuri riguarderanno la definizione e l'integrazione nella piattaforma di un sistema di gestione delle previsioni meteorologiche, la creazione di ulteriori viste nel database dedicate all'estrazione di KPI relativi agli edifici ed un sistema di diagnostica low level dedicato alle anomalie di basso livello (sensoristica malfunzionante, installazione errata ecc..).