

**POLITECNICO DI MILANO**

Scuola di Ingegneria Industriale e dell'informazione

Corso di Laurea Magistrale in  
Computer Science and Engineering - Ingegneria Informatica



**ANALISI E PROGETTAZIONE DI  
UN'INTERFACCIA HMI PER UNA MACCHINA  
PER IL TELECONTROLLO**

Relatore:

Prof. ssa Raffaella MIRANDOLA

Elaborato di Laurea Magistrale di:

Alessandro PICCA

Matricola n. 835622

Anno Accademico 2015 - 2016



*Ai miei nonni.*



# Abstract

In recent years the remote control, that is the interaction at a distance with a more or less complex system, both via data read operations (monitoring) and through states setting operations by remote status (control), is in a widely expansionist phase and it is adapting to a great variety of uses in everyday life: aqueducts, railways, traffic light systems, medical equipment, only to name a few of them.

In this paper we will describe the entire design process - feasibility study, requirements analysis, implementation, testing – of a graphical HMI (Human-Machine Interface) interface, that has to be applied to the machine STCE/RTU already in use by Selta SpA for remote control and monitoring of electrical substations.

The biggest problem that the machine in question actually has, is the fact that, due to various hardware limitations, has so far not been possible to design an HMI suitable for the machine in question, so, at the present state of things, a graphical output can be obtained only by relying on external tools. Now, new technologies and improvements in hardware have enabled the realization of an HMI directly on STCE/RTU, which can provide a visualization as immediate, realistic and understandable as possible, without affecting in any way the functionality of the machine.

It was also tried to add a feature to the machine: the possibility to "edit" a self-created network , which will later be the subject of monitoring by the machine itself.

In the six chapters that compose this document are developed, analyzed and clarified the concepts that have been exposed above.



## Sommario

In questi ultimi anni il telecontrollo, ovvero l'interazione a distanza con un impianto più o meno complesso, sia mediante operazioni di lettura dati (monitoraggio) che attraverso operazioni di modifica dello stato da remoto (regolazione) è in una fase ampiamente espansionistica e si sta adattando ai più disparati usi nella quotidianità: acquedotti, reti ferroviarie, impianti semaforici, apparecchiature sanitarie, solo per citarne alcuni.

In questo elaborato si descrive l'intero iter progettuale – studio di fattibilità, analisi dei requisiti, implementazione, testing - di un'interfaccia grafica HMI (Human-Machine Interface) da applicare alla macchina STCE/RTU già in uso da parte di Selta SpA per il telecontrollo e il monitoraggio delle sottostazioni elettriche.

La problematica maggiore che presenta la macchina in questione è il fatto che, a causa di varie limitazioni hardware, non è stato finora possibile progettare una HMI sulla macchina in questione, ma, allo stato attuale delle cose, un output grafico è ottenibile solamente appoggiandosi a strumenti esterni.

Ora, le nuove tecnologie e migliorie a livello hardware, hanno permesso la realizzazione di un HMI direttamente sulla STCE/RTU, che possa fornire una visualizzazione quanto più immediata, realistica e comprensibile possibile, senza andare ad intaccare la funzionalità dell'apparecchiatura.

Si è inoltre cercato di aggiungere una funzionalità alla macchina, ovvero la possibilità di “editare” una propria rete elettrica, che sarà poi oggetto di monitoraggio da parte dell'apparecchiatura stessa.

Nei sei capitoli di cui si compone questo documento si sviluppano, si analizzano e si chiariscono i concetti qui sopra esposti.





## Ringraziamenti

*In questo spazio vorrei ringraziare tutti quanti mi hanno supportato, sostenuto, consigliato e aiutato durante il mio percorso di studi e in particolare in questo ultimo periodo.*

*Innanzitutto vorrei ringraziare Selta SpA, in particolare Gianluca, Danilo, Laura e tutti i membri del laboratorio di ricerca e sviluppo, sia per la possibilità concessami di poter svolgere questo lavoro in azienda, sia per il prezioso e continuo aiuto e supporto ricevuti durante tutti questi mesi.*

*Ci tenevo a ringraziare particolarmente la Prof.ssa Mirandola, che ha accettato di fare da relatrice per questo lavoro di tesi, per la sua disponibilità e il suo supporto durante questo lavoro di tesi.*

*Un grosso grazie va anche alla mia famiglia, sia per i sacrifici che hanno fatto per permettermi di raggiungere questo traguardo, che per la fiducia risposta in me.*

*Inoltre un ringraziamento va ai miei compagni di corso di studi, in particolare Eliseo, Francesco e Leonardo, nonché a tutti i miei amici: sarebbe troppo lungo citarli tutti, ma credo che ognuno di loro potrà riconoscersi in queste poche righe; sicuramente il loro contributo e la loro amicizia è stata preziosa in questo percorso.*

*Infine, grazie a Sonia, che mi è vicina ogni giorno, e che nei momenti difficili ha sempre trovato le parole o i gesti giusti per motivarmi e spingermi ad andare avanti; buona parte di questo traguardo è anche merito suo.*

*Alessandro*



## Indice dei contenuti

Capitolo 1: Introduzione .....	1
Capitolo 2: Stato dell'arte.....	3
2.1 Il telecontrollo: definizione formale e caratteristiche generali degli apparati preposti.....	4
2.2 Struttura dell'architettura e motivazioni dell'assenza di una configurazione universale.....	5
2.3 Descrizione di una sottostazione elettrica e dei suoi principali componenti.....	6
2.4 La macchina STCE-RTU.....	7
2.4.1 Caratteristiche generali e configurazioni.....	7
2.4.2 Configurazione "Stand-alone".....	8
2.4.3 Configurazioni "Gateway" e "Proxy".....	9
2.4.4 Architettura della macchina e struttura hardware.....	11
2.4.5 Struttura Software.....	12
2.4.6 Calcolo dei punti e limitazioni.....	14
Capitolo 3: Requisiti e specifiche del software: analisi del problema e formulazione di una soluzione.....	15
3.2 Definizione del problema.....	16
3.2.1 Maggiori difficoltà incontrate da dover risolvere .....	16
3.2.2 Descrizione generale.....	17
3.2.3 Obiettivi.....	18
3.2.4 Glossario.....	18
3.2.5 Panoramica.....	20
3.3 Requisiti, limitazioni e assunzioni del sistema.....	20
3.3.1 Requisiti funzionali.....	20
3.3.1.1 Acquisizione punti e loro stato da un database di campo.....	21
3.3.1.2 Editor del proprio campo personalizzato.....	22
3.3.1.3 Aggiornamento buffer degli eventi.....	22
3.3.1.4 Monitoraggio.....	22
3.3.1.5 Visualizzazione dettagli.....	23
3.3.2 Requisiti non funzionali.....	23
3.3.2.1 Interfacce di sistema, interfaccia utente e page sketches.....	24
3.3.2.2. Requisiti di database e di performance.....	26
3.3.3 Vincoli di design e conformità agli standard.....	27
3.3.4 Assunzioni e limitazioni.....	27
3.3.5 Caratteristiche di funzionamento.....	28
3.4 Soluzione proposta e sua formalizzazione.....	28
3.4.1 Identificazione e descrizione dei principali attori del sistema.....	29
3.4.2 Analisi function point.....	29
3.4.3 Scenari.....	32
3.4.4 Use Cases e Sequence Diagrams.....	34
3.4.4.1 Visualizzazione informazioni nella pagina utente.....	35
3.4.4.2 Creazione di una nuova rete.....	36
3.4.4.3 Avvio monitoraggio.....	37
3.4.4.4 Stop monitoraggio.....	38
3.4.4.5 Apertura pagina dettaglio.....	39
3.4.4.6 Apertura buffer degli eventi.....	40
3.4.5 Attributi propri del software.....	41
Capitolo 4: Specifiche e proprietà del software realizzato.....	42
4.1. Scelte implementative.....	43

4.2 Elenco dei widget utilizzati e loro significato.....	45
4.3 Schema architetturale del sistema proposto .....	51
4.4 Soluzione client-side e caratteristiche minime del sistema client.....	53
4.5 Vincoli tecnici e strutturali della macchina .....	54
4.6 Modifiche alle specifiche della sezione dei requisiti.....	55
4.7 Operazione di avvio e descrizione del sistema.....	55
4.8 Use Case diagram.....	62
4.9 Flowchart del processo del software.....	64
4.10 Schema di raggiungibilità tra le varie pagine.....	66
Capitolo 5: Sperimentazione e testing.....	67
5.1 Introduzione .....	67
5.2 Il software testing.....	67
5.3 Demo e testing del software realizzato.....	70
5.3.1 Caratteristiche delle risorse utilizzate.....	70
5.3.2 Definizione della modalità di testing effettuata.....	71
5.3.3 Test dei requisiti e delle funzionalità.....	71
Capitolo 6: Conclusioni e sviluppi futuri.....	74
Bibliografia e Sitografia.....	76

## Indice delle illustrazioni

Figura 2.1: Esempio di utilizzo dell'apparato STCE/RTU in sistemi di telecontrollo.....	7
Figura 2.2: Esempio di configurazione sistemistica “stand-alone”.....	9
Figura 2.3: Esempio di configurazione con apparato concentratore, valida per configurazioni di tipo “gateway” o “proxy”.....	11
Figura 3.1: page sketch della pagina principale/ pagina utente.....	24
Figura 3.2: page sketch della pagina di monitoraggio.....	25
Figura 3.3: page sketch della pagina di dettaglio.....	26
Figura 3.4: Sequence diagram relativo alla visualizzazione pagina personale dell'utente.....	35
Figura 3.5: Sequence diagram relativo alla creazione di una nuova rete da parte dell'utente.....	36
Figura 3.6: Sequence diagram relativo all'operazione di avvio monitoraggio.....	37
Figura 3.7: Sequence diagram relativo all'operazione di interruzione del monitoraggio.....	38
Figura 3.8: Sequence diagram relativo all'apertura, e conseguente visualizzazione, della pagina di dettaglio del componente selezionato.....	39
Figura 3.9: Sequence diagram relativo alla visualizzazione e consultazione del buffer degli eventi.....	40
Figura 4.1: Widget componente di allarme.....	45
Figura 4.2: Widget componente di collegamento ad angolo retto.....	45
Figura 4.3: Widget componente autotrasformatore.....	45
Figura 4.4: Widget componente batteria condensatore.....	46
Figura 4.5: Widget di componente collegamento.....	46
Figura 4.6: Widget di componente collegamento con sbarra 1.....	46
Figura 4.7: Widget di componente collegamento con sbarra 2.....	46
Figura 4.8: Widget di doppia sbarra.....	46
Figura 4.9: Widget di componente vuoto a sfondo bianco.....	47
Figura 4.10: Widget di componente vuoto a sfondo grigio.....	47
Figura 4.11: Widget di etichetta.....	47
Figura 4.12: Widget di componente ad incrocio.....	47
Figura 4.13: Widget di interruttore.....	47
Figura 4.14: Widget di label con testo.....	48
Figura 4.15: Widget di linea.....	48
Figura 4.16: Widget di linea con incrocio a destra.....	48
Figura 4.17: Widget di linea con incrocio a sinistra.....	48
Figura 4.18: Widget di divieto di accesso.....	48
Figura 4.19: Widget di predispositore.....	49
Figura 4.20: Widget di allarme di impianto.....	49
Figura 4.21: Widget di allarme di impianto 2.....	49
Figura 4.22: Widget di sezionatore.....	49
Figura 4.23: Widget di collegamento a “T”.....	49
Figura 4.24: Widget di tag vuota.....	50
Figura 4.25: Widget con simbolo di tensione.....	50
Figura 4.26: Widget con simbolo di massa a terra.....	50
Figura 4.27: Widget di variatore.....	50
Figura 4.28: Schema dell'architettura client/server del sistema.....	51
Figura 4.29: Parte della schermata della pagina utente che mostra le credenziali dell'utente attivo.....	56
Figura 4.30 Parte della schermata della pagina utente che mostra il bottone di scelta della funzionalità.....	56
Figura 4.31: Griglia con widget dei componenti.....	57

Figura 4.32: Esempio di componente selezionato.....	58
Figura 4.33: Esempio di componente ruotato.....	58
Figura 4.34: Parte della griglia 20x20 sulla quale comporre la rete.....	58
Figura 4.35: Pulsanti di reset e di salvataggio della rete.....	58
Figura 4.36: Icona di redirectione alla pagina di monitoraggio.....	59
Figura 4.37: Icona di redirectione alla pagina iniziale.....	59
Figura 4.38: Pagina di gestione attività di monitoraggio.....	60
Figura 4.39: Rete in fase di monitoraggio.....	61
Figura 4.40: Schermata buffer degli eventi.....	62
Figura 4.41: Use Case Diagram della soluzione proposta.....	63
Figura 4.42: Diagramma di flusso delle funzionalità del software.....	65
Figura 4.43: Grafo di raggiungibilità tra le varie pagine del sistema.....	66
Figura 5.1: Propagazione dei difetti nelle varie fasi del ciclo di vita del software.....	69

## Indice delle tabelle

Tabella 3.3: Tabella descrittiva dello use case “Visualizzazione informazioni nella pagina utente...”	35
Tabella 3.4: Tabella descrittiva dello use case “Creazione di una nuova rete”	36
Tabella 3.5: Tabella descrittiva dello use case “Avvio monitoraggio”	37
Tabella 3.6: Tabella descrittiva dello use case “Stop monitoraggio”	38
Tabella 3.7: Tabella descrittiva dello use case “Apertura pagina dettaglio”	39
Tabella 3.8: Tabella descrittiva dello use case “Apertura buffer degli eventi”	40
Tabella 4.1: Vincoli della macchina	54





# Capitolo 1: Introduzione

In un contesto come quello attuale, caratterizzato da una notevole diffusione dei sistemi embedded dovuta alla disponibilità di nuove tecnologie e sistemi adatti a diversi contesti di applicazione, i dispositivi HMI (Human Machine Interface) rappresentano validi strumenti per gestire diversi tipi di sistemi complessi, quali, ad esempio, impianti di gestione delle risorse, strumenti di diagnostica, reti o sistemi integrati di building automation; questi tipi di soluzione possono essere controllati in maniera semplice ed efficace da un operatore attraverso i dispositivi HMI.

Il mercato HMI sta seguendo un andamento destinato ad un approccio completamente diverso nella gestione e visualizzazione dei dati: la sempre crescente possibilità di utilizzo dell'hardware, sia a livello di riduzione del costo della memoria, sia grazie alla presenza di microprocessori più potenti, sta portando a una gestione più centralizzata della visualizzazione di dati provenienti da remoto, senza dover quindi demandare tale funzione ad un centro di controllo esterno.

Il mercato HMI si sta sviluppando anche verso applicazioni capaci di fornire caratteristiche quali la gestione distribuita di sistemi complessi (accesso remoto), e la configurazione automatica.

Un altro concetto fondamentale legato a quello di HMI è sicuramente quello di telecontrollo, anch'esso in crescita esponenziale, probabilmente a causa della perdurante crisi, che ha condotto le imprese del settore a definire nuovi approcci sia nei settori emergenti sia in quelli più tradizionali, consentendo loro di cogliere nuove opportunità offerte dall'efficienza energetica, dalla sostenibilità e di manifestare un interesse crescente verso il concetto di Smart Grid.

Questo lavoro, nel dettaglio, tratta dell'attività di telecontrollo sulle sottostazioni elettriche, fondendo i concetti definiti in precedenza per quanto riguarda un'HMI con le caratteristiche di immediatezza, chiarezza e semplicità tipiche di un'interfaccia grafica.

Il lavoro in sé è comprensivo della parte di analisi dei requisiti e del design dell'interfaccia, nonché del suo successivo testing sulla macchina per la quale è stata ideata, partendo dallo stato attuale dell'arte e cercando di capire dove si sarebbe potuto intervenire per apportare migliorie e innovazioni.

Nello specifico, questo documento si compone così:

- capitolo 2: panoramica sullo stato dell'arte e descrizione generica del funzionamento dell'attività di telecontrollo e di una sottostazione elettrica; descrizione sia a livello software che hardware dell'architettura della macchina per cui viene progettata l'interfaccia;

- capitolo 3: analisi dei requisiti e delle specifiche che richiede il software progettato, strategie di design, assunzioni e vincoli propri del sistema, analisi preliminare di tipo Function Point;

- capitolo 4: descrizione dettagliata della soluzione proposta: scelte implementative, migliorie, funzionalità, schema architetturale e modifiche dei requisiti proposti a valle, diagrammi per facilitare la comprensione della soluzione;

- capitolo 5: descrizione della tipologia di testing effettuata, nonché dell'attività di testing vera e propria, svolta sulla macchina STCE/RTU utilizzando una rete elettrica standard e verifica del raggiungimento degli obiettivi descritti dai requisiti funzionali;

- capitolo 6: conclusioni relative a lavoro svolto, criticità incontrate, obiettivi raggiunti ed elencazioni dei possibili sviluppi futuri.

## Capitolo 2: Stato dell'arte

Gli ambienti di applicazione dei sistemi di telecontrollo sono molto vasti e diversificati, ad esempio si possono citare <sup>[1]</sup> : i settori di pubblica utilità (produzione e distribuzione dell'energia, gestione delle risorse idriche, trasporto ed immagazzinamento dei gas), il settore agricolo (irrigazione automatizzata e telecontrollata), il settore ambientale (rilevamento dei parametri di inquinamento urbano), il settore medico (apparecchiature di dialisi, particolari tipi di diagnostica, ecc.), oltre agli ambiti più comuni della quotidianità (teleriscaldamento delle abitazioni, impianti semaforici, gestione dell'illuminazione pubblica, building automation).

In questo capitolo presentiamo lo stato dell'arte riguardante tutto ciò che concerne l'ambito del telecontrollo, dandone in primo luogo, nella sezione 2.1, una definizione formale e descrivendo le caratteristiche generali che devono avere le apparecchiature preposte a questa attività.

In seguito, nella sezione 2.2, descriveremo genericamente i vincoli che presenta un'architettura preposta al telecontrollo, spiegando brevemente le motivazioni per cui non è possibile realizzare un'architettura standard comune a tutti gli ambiti applicativi del telecontrollo citati in precedenza.

Nella sezione 2.3 realizzeremo una breve descrizione di una sottostazione elettrica, descrivendo sommariamente i suoi componenti principali e le loro caratteristiche.

Nella sezione 2.4.1 descriveremo i tratti essenziali della macchina per cui è stata progettata la Gui (*Graphical User Interface*) e nelle sezioni successive enumereremo e descriveremo le sue diverse configurazioni possibili. Nella sezione 2.4.4 effettueremo una panoramica sull'architettura della macchina e sulla sua struttura hardware, mentre nella sezione 2.4.5 ci focalizzeremo su struttura e operazioni della macchina a livello software.

Infine, nella sezione 2.4.6 spiegheremo come avviene attualmente il calcolo dei punti del campo, operazione fondamentale nell'attività di telecontrollo, e quali limitazioni incontra attualmente questa operazione e le altre correlate ad essa.

## **2.1 Il telecontrollo: definizione formale e caratteristiche generali degli apparati preposti**

Il telecontrollo si configura come una soluzione di automazione che, grazie all'utilizzo di un software situato in un *centro di controllo*, permette la raccolta di dati ed informazioni da processi che si svolgono in postazioni remote. In tutti gli ambiti in cui assume una certa utilità l'utilizzo del telecontrollo, riveste un'importanza fondamentale il *centro di controllo*, che deve poter scambiare informazioni sia in ingresso che in uscita con tutte le postazioni remote.

L'*unità remota*, a sua volta, deve poter rilevare i valori delle variabili indipendenti di processo, che devono dar luogo alle opportune regolazioni da effettuare sugli attuatori (ad esempio apertura/chiusura di una valvola, posizionamento, ecc..) e che devono essere testati, per trasmettere al *centro di controllo* l'eventuale superamento di predeterminate soglie. All'unità di controllo, inoltre, devono essere trasmessi gli stati/valori di variabili significative, segnalazioni di servizio e allarmi. La modalità di trasmissione degli stati/valori delle variabili prescelte deve essere stabilita a priori: si può trasmettere in modo ciclico, a cadenza stabilita, oppure su evento, ovvero quando cambia lo stato/valore di una variabile.

La moderna gestione di impianti, attrezzature automatizzate, reti elettriche ecc..., richiede che le *apparecchiature periferiche di telecontrollo* siano in grado di gestire le informazioni da e verso l'impianto in modo sempre più integrato con gli altri dispositivi elettronici presenti in campo (impianto, stazione elettrica o altro), rappresentati principalmente da apparati trasduttori, da apparati di controllo automatico e protezione, da apparati di registrazione, da sistemi di supervisione locale.

Assume inoltre sempre maggiore importanza la capacità delle singole apparecchiature di telecontrollo di comunicare con i centri di gestione della rete in modo rapido, puntuale e flessibile, per consentire l'impiego delle più svariate reti di trasmissione dati e la connessione di tutti i centri interessati.

Di pari passo le funzionalità descritte in precedenza, fornite dalle apparecchiature periferiche di telecontrollo, sono ora completate da prestazioni quali la registrazione cronologica degli eventi, la generazione automatica di sequenze di regolazione e comando, la lettura e la modifica dei parametri di funzionamento dei dispositivi esterni collegati.

Accanto alle tradizionali caratteristiche di affidabilità e di sicurezza, le apparecchiature periferiche di telecontrollo devono pertanto presentare:

- elevate capacità elaborative, modulabili in funzione delle specifiche esigenze;
- grandi capacità di comunicazione, per adattarsi alle evoluzioni dei protocolli e delle sistemistiche di rete;
- un'architettura modulare e ad intelligenza distribuita, per risolvere in modo ottimale le problematiche di installazione;
- elevate capacità di discriminazione cronologica, unitamente alla possibilità di sincronizzazione con orologi ad elevata precisione, come quello diffuso dal segnale radio satellitare GPS.

L'aumentata complessità degli apparati richiede infine adeguati strumenti di configurazione e diagnostica, in grado di garantire interventi di manutenzione semplici, tempestivi ed efficaci.

## ***2.2 Struttura dell'architettura e motivazioni dell'assenza di una configurazione universale***

L'architettura tipo<sup>[2]</sup> di un sistema di telecontrollo e supervisione, come detto in precedenza, deve essere strutturata in funzione delle peculiarità e delle

caratteristiche specifiche del processo cui il sistema è destinato, nonché delle problematiche geografiche ed orografiche del territorio, al fine di soddisfare al meglio le esigenze operative, funzionali, logistiche e logiche. Pertanto risulta semplice capire come non possa esistere una “configurazione universale” che soddisfi tutti i requisiti e si adatti pienamente ad ogni tipo di applicazione. Viceversa, l’architettura di un sistema di telecontrollo dipende dalle soluzioni e dalle regole progettuali, spesso frutto dell’esperienza specifica, che di volta in volta vengono identificate, verificate ed adottate.

### **2.3 Descrizione di una sottostazione elettrica e dei suoi principali componenti**

La macchina oggetto di studio di questo lavoro opera principalmente sulle sottostazioni elettriche <sup>[3]</sup>, elementi costituenti i nodi della rete di trasmissione dell'energia elettrica, che possono avere tutte o alcune delle seguenti funzioni:

- Interconnessione di più linee elettriche in AT allo stesso livello di tensione, creando un nodo della rete (tramite le sbarre);
- Interconnessione di più linee elettriche in AT allo stesso livello di tensione (tramite i trasformatori);
- Rifasamento (fornitura di tutta o di parte della potenza reattiva elettrica necessaria al carico) della potenza apparente della rete;
- Conversione da corrente continua in corrente alternata e viceversa.

Le sottostazioni elettriche sono costituite sia da componenti a bassa e media tensione (BT e MT), che compongono il cosiddetto *sistema secondario*, sia da componenti ad alta tensione (AT), che compongono il *sistema primario*. Quest'ultimo è generalmente composto dalle sbarre e dai montanti: le prime, che possono essere singole oppure doppie – queste ultime garantiscono maggiore flessibilità di esercizio ed affidabilità del sistema - costituiscono il nodo della rete elettrica di trasmissione, mentre i montanti comprendono tutte le apparecchiature AT che sono collegate alle sbarre.

La sottostazione si ottiene semplicemente componendo in modo opportuno i montanti ed inserendo i moduli terminali di sbarra.

## 2.4 La macchina STCE-RTU

### 2.4.1 Caratteristiche generali e configurazioni

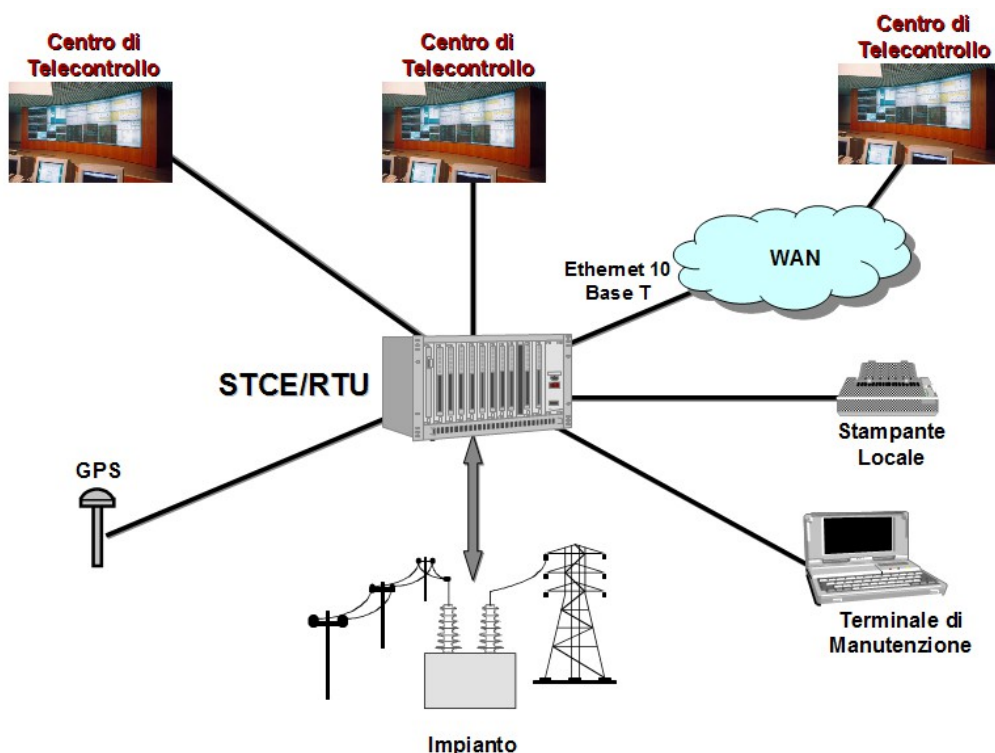


Figura 2.1: Esempio di utilizzo dell'apparato STCE/RTU in sistemi di telecontrollo

L'apparato STCE/RTU <sup>[4]</sup>, macchinario di proprietà di Selta SpA, è un apparato periferico di telecontrollo destinato principalmente ad applicazioni di tipo elettrico, ma progettato in modo tale da consentirne l'impiego anche in sistemi di tipo diverso (controllo di gasdotti, oleodotti, ecc..). Il sistema consente di realizzare l'acquisizione di stati del campo, sia di tipo digitale che analogico, e di imporre sul campo comandi e regolazioni.

Le informazioni acquisite possono essere elaborate localmente oppure trasmesse ad un centro di supervisione e controllo, locale oppure remoto.

Le tipologie di configurazione della STCE/RTU si possono ricondurre sostanzialmente a tre classi:

- Configurazione “Stand-alone”:
- Configurazione “Gateway”:
- Configurazione “Proxy”

## **2.4.2 Configurazione “Stand-alone”**

Questa configurazione prevede che l'attività di telecontrollo della stazione sia realizzata da un unico apparato STCE/RTU, completamente concentrato in un telaio oppure distribuito in più armadi, afferente direttamente ai centri di telecontrollo tramite rete dedicata o LAN/WAN.

Questa architettura è la più classica tra quelle presentate, e prevede l'interfacciamento della macchina al campo attraverso unità periferiche di I/O parallele (UIO/P) e la comunicazione con altri apparati tramite periferiche di I/O seriali (UIO/S).

Infine, la comunicazione con il centro di controllo è gestita dall'Unità Centrale CPU2000 con protocollo di telecontrollo IEC 60870-5-101 oppure IEC 60870-5-104



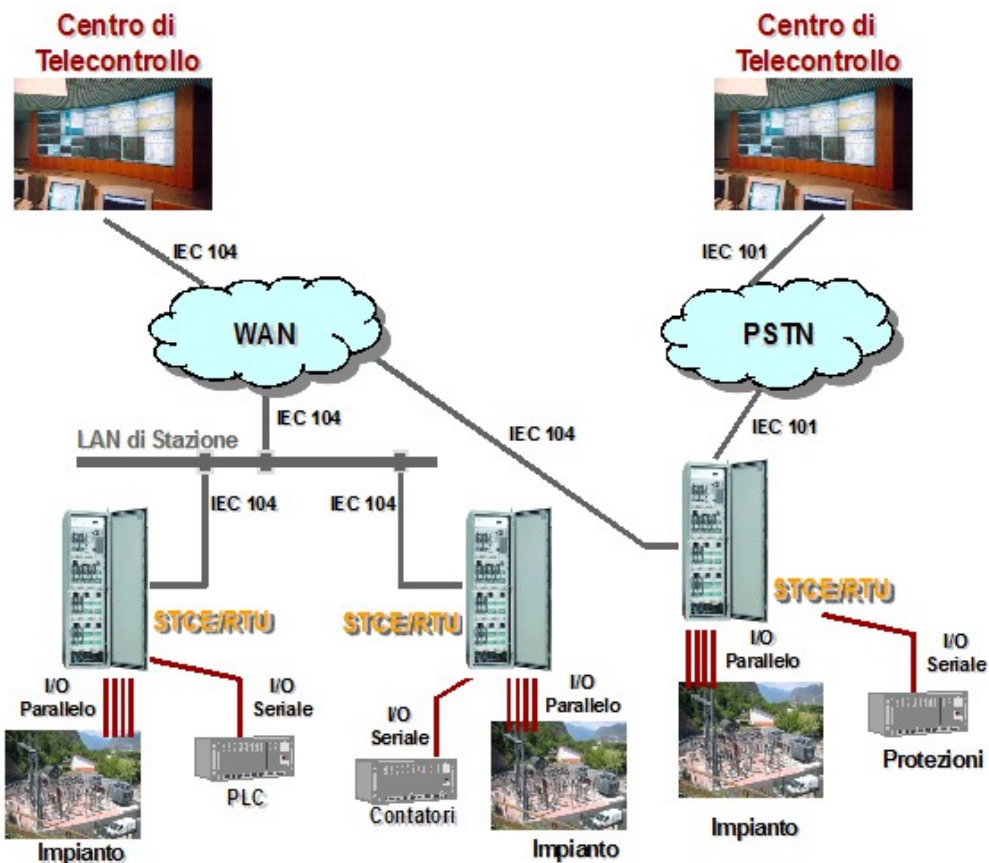


Figura 2.2: Esempio di configurazione sistemistica "stand-alone"

### 2.4.3 Configurazioni "Gateway" e "Proxy"

Queste due configurazioni presentano diverse somiglianze dal punto di vista dell'architettura sostanziale, mentre differiscono in maniera significativa dal punto di vista funzionale; in entrambi i casi il sistema risulta però diviso nei seguenti due livelli:

- Livello distribuito: costituito da apparati STCE/RTU distribuiti in impianto e direttamente interfacciati all'I/O (in parallelo o in serie). Per il collegamento con l'apparato concentratore possono essere utilizzati sia il protocollo IEC 60870-5-101, sia IEC 60870-5-104, istanziati su ciascuna unità centrale.
- Livello concentratore: costituito da un solo apparato STCE/RTU che si interfaccia tramite le unità di I/O seriale ai diversi apparati distribuiti e tramite l'unità centrale al centro di telecontrollo.

La differenza funzionale delle due configurazioni è invece data dal fatto che, mentre nella modalità *gateway* l'apparato concentratore reinstrada e transcodifica le informazioni ricevute/trasmesse da/verso il centro, permettendo la modifica degli identificativi di protocollo utilizzati dal centro, e quindi la “*personalizzazione*” degli indirizzi di impianto, fungendo inoltre da convertitore di protocollo, nella modalità *proxy* l'apparato concentratore effettua esclusivamente l'operazione di reinstradamento delle informazioni ricevute/trasmesse da/verso il centro, lasciando inalterati gli identificativi di protocollo utilizzati dal centro. Nella modalità *gateway*, dal punto di vista del centro, i diversi apparati STCE/RTU possono essere visti come un solo apparato *stand-alone*.

Dopo aver descritto questi due tipi di configurazione, è opportuno fare alcune piccole considerazioni, comuni ad entrambe le tipologie:

- L'apparato con funzione di concentratore realizza la comunicazione con il posto centrale implementando il relativo protocollo, ma non svolge funzioni di telecontrollo;
- L'apparato concentratore conosce l'associazione tra ogni punto (generico circuito di ingresso/uscita verso il campo) e il relativo apparato STCE/RTU, in modo da mettere fuori attendibilità tutti i punti acquisiti dal generico apparato in caso di fault dello stesso o del relativo collegamento;
- L'apparato concentratore prevede su di esso la realizzazione di un buffer eventi nel quale confluiscono tutti gli eventi generati sugli apparati STCE/RTU distribuiti in impianto; analogamente esso mantiene un'immagine aggiornata del database globale dei punti.

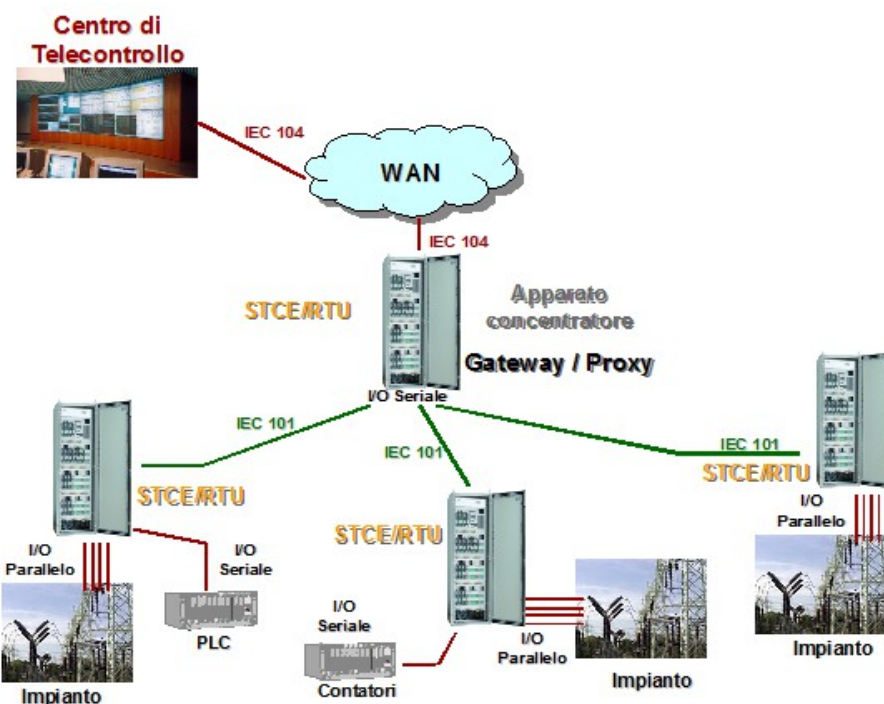


Figura 2.3: Esempio di configurazione con apparato concentratore, valida per configurazioni di tipo "gateway" o "proxy"

## 2.4.4 Architettura della macchina e struttura hardware

L'apparato STCE/RTU, come già accennato in precedenza, presenta un'architettura basata su due livelli, uno *centralizzato* ed uno *distribuito* sulle unità periferiche, interconnessi mediante un bus seriale ridondato (Bus di apparato).

Il primo livello è costituito da un elaboratore centrale (CPU), completamente ridondabile, a cui sono demandate tutte le funzioni di concentrazione delle informazioni raccolte dagli impianti e di comunicazione con i centri di supervisione.

Ad esso sono inoltre demandate tutte le attività ausiliarie, quali la sincronizzazione con il segnale satellitare GPS, l'eventuale interfacciamento con la stampante locale e la comunicazione con il terminale di manutenzione. Il secondo livello è costituito dall'insieme delle unità di interfacciamento al campo, tutte dotate di elaboratore locale, a cui è demandato il compito di

standardizzare le informazioni raccolte dagli impianti o di trasdurre le informazioni ad essi destinate: le unità periferiche di I/O, equipaggiate in funzione delle caratteristiche e delle dimensioni dell'impianto controllato, comprendono: i circuiti di condizionamento dei segnali in ingresso, i circuiti di pilotaggio dei segnali in uscita e le vie seriali per la connessione di trasduttori, contatori, protezioni, automatismi o RTU remote.

La struttura hardware risulta molto semplice, grazie a diversi fattori, tra i quali è opportuno citare:

- L'integrazione delle funzioni centralizzate essenziali in un unico modulo: l'unità centrale;
- Il possibile raddoppio dell'unità centrale, consistente in un semplice inserimento di una seconda unità centrale nella posizione pre-assegnata del sub telaio base;
- L'eventuale raddoppio del modulo di alimentazione, mediante l'inserimento di un secondo convertitore nel modo descritto appena sopra per l'unità centrale;
- La possibilità di inserire in qualunque posizione prevista per il livello periferico tutte le unità periferiche, sia seriali che parallele.

Il sub telaio base contiene: un convertitore, un'unità centrale, un'unità di servizio e fino a 7 unità periferiche, mentre i subtelai estensione (fino ad un massimo di 3) contengono ciascuno 14 unità periferiche.

Mentre le unità periferiche seriali consentono alla RTU di interfacciare localmente trasduttori di misura, dispositivi di protezione o altri apparati di campo, oppure di acquisire informazioni da altri apparati RTU di livello gerarchico inferiore, le unità periferiche parallele si interfacciano direttamente al campo e realizzano il prelievo dal campo di stati logici e di grandezze analogiche per l'elaborazione e la successiva trasmissione ai centri di controllo.

### **2.4.5 Struttura Software**

A livello software, le operazioni principali svolte dalla STCE/RTU sono:

- L'acquisizione continua dello stato del campo attestato ai gruppi di I/O;

- La gestione del database dinamico corrente, sia riguardo lo stato dei punti di campo che quello dei punti calcolati; nel paragrafo successivo verrà spiegato come avviene il suddetto calcolo;
- La gestione del database diagnostico, contenente lo stato diagnostico di tutte le unità;
- La gestione del database semistatico, contenente il log degli eventi di campo;
- La gestione del database statico, contenente la configurazione di apparato e il log degli eventi di diagnostica;
- La trasmissione al/ai centro/i delle informazioni di campo e loro classificazione secondo due distinte categorie (classe 1 e classe 2);
- La gestione della stampante locale;
- La gestione delle logiche di automazione;
- La gestione della sincronizzazione oraria e del GPS;
- La gestione della diagnostica di apparato e dell'eventuale dualizzazione dei componenti.

Esiste inoltre un log locale, su memoria non volatile, relativo ad eventi quali: diagnostica di apparato, variazione di ingressi digitali, superamento di soglie di allarme per misure, ecc.. Il log locale si configura come un buffer di capacità 2000 a coda FIFO, consultabile tramite web server; per ogni record vengono memorizzati il timestamp, il tipo di informazione, l'indice identificativo ed il nuovo stato o valore.

Infine esiste un log diagnostico, simile al precedente ma con capacità limitata a 200, che segnala l'insorgenza di un evento diagnostico, ma ne notifica il rientro solo alla risoluzione di tutte le cause scatenanti. Anch'esso, come il precedente, è visualizzabile tramite web server.

## 2.4.6 Calcolo dei punti e limitazioni

I punti restituiti come output dal campo sono calcolabili mediante espressioni booleane programmabili tramite configuratore; in particolare le operazioni logiche effettuabili sui punti del database di campo sono le classiche operazioni logiche, e quindi: NOT, AND, OR, XOR e le loro concatenazioni.

Il limite che presenta la macchina è quello di non riuscire attualmente a fornire all'utente una visualizzazione grafica di quanto viene restituito dal campo: le limitazioni imposte dall'hardware, quali il costo eccessivo della memoria e la limitata capacità dei microprocessori, hanno fatto sì che fino ad oggi la funzione di visualizzazione grafica della “fotografia del campo” fosse demandata ad una macchina esterna.

Per questi motivi, quindi, non è possibile attualmente avere un quadro facilmente visibile nell'immediato della situazione del campo oggetto del monitoraggio.

Negli ultimi tempi però le miglorie hardware hanno reso possibile la creazione di interfacce grafiche posizionabili direttamente sulla macchina, senza dover accedere a centri di controllo esterni, per cui questo lavoro di tesi è incentrato proprio sulla progettazione e sull'implementazione di tale migloria.

## Capitolo 3: Requisiti e specifiche del software: analisi del problema e formulazione di una soluzione

Questa parte del documento si prefigge lo scopo di descrivere e spiegare in maniera più accurata le funzionalità dell'interfaccia grafica progettata per la macchina STCE/RTU di proprietà di Selta SpA e utilizzata in operazioni di telecontrollo su sottostazioni elettriche.

L'analisi dei requisiti e dei vincoli è basata sia sull'accordo con Selta SpA, riguardante le caratteristiche e le funzionalità che deve avere il software in oggetto, nonché le modalità con cui i requisiti e le funzionalità devono essere soddisfatti, sia sull'analisi delle caratteristiche offerte della soluzione attualmente in uso sul macchinario STCE/RTU.

Il documento mostra non soltanto le caratteristiche principali del sistema, ma anche le strategie di design, le assunzioni ed i vincoli che rendono più facile l'implementazione di questo software, oltre ad un'analisi preliminare di tipo Function Point riguardante l'effort necessario alla realizzazione dell'intero lavoro.

I destinatari di questo documento sono gli utenti che si interfacciano con la macchina, e che grazie al documento stesso dovrebbero avere la sicurezza di conoscere e comprendere il software in tutti i suoi aspetti.

Prima di proseguire, è opportuno fare cenno del fatto che per la stesura di questo documento si fa riferimento all'IEEE Recommended Practice for Software Requirements Specifications:  
<http://standards.ieee.org/findstds/standard/830-1998.html>

## **3.2 Definizione del problema**

In questa sezione verrà effettuata un'analisi del problema in questione, considerando in primo luogo le criticità che esso presenta attualmente, accompagnate dalle soluzioni proposte per risolverle, descrivendo poi genericamente l'interfaccia da implementare e quali sono gli obiettivi che essa deve necessariamente raggiungere, oltre a specificare un glossario valido per il resto del documento ed una panoramica sul resto del capitolo.

### **3.2.1 Maggiori difficoltà incontrate da dover risolvere**

In questa parte del documento si vogliono elencare i principali fattori che hanno portato all'insorgere di difficoltà durante tutte le fasi del lavoro, da quella di analisi a quella di stesura dei requisiti fino a quella di progettazione vera e propria.

- Il fattore temporale e la gestione del tempo a disposizione hanno giocato un ruolo rilevante: il fatto di avere scadenze abbastanza stringenti ha rappresentato una *challenge* importante da superare; la cosa è stata gestita stabilendo “*checkpoint*” intermedi e facendo a cadenza quasi quotidiana un report scritto su quanto fatto e quanto ancora da fare.

- La conoscenza iniziale della macchina sulla quale doveva essere implementata l'interfaccia era pressoché nulla, e ciò ha comportato un lavoro supplementare di studio dell'apparato per arrivare a possederne una conoscenza quantomeno sufficiente, incidendo parecchio sul percorso di sviluppo del software.

- Il dover pensare ad un'interfaccia che coniugasse la necessità di immediatezza nell'acquisizione dell'informazione con una qualità grafica accettabile e con la possibilità di adattare l'interfaccia stessa ad eventuali sviluppi futuri ha sottratto un po' di tempo alla fase implementativa, aggiungendone altrettanto a quella pre-progettuale.



- Nonostante il lavoro svolto sia inserito in un contesto aziendale, e quindi in un'ottica di lavoro di team, il lavoro di gruppo all'interno di un'azienda, diversamente dall'ambito universitario che ho finora vissuto, prevede che dopo la definizione degli obiettivi comuni, tutte le componenti di un progetto sviluppino la propria sezione di lavoro in maniera autonoma; per questo motivo la gestione della parte di coding è stata gestita individualmente, e ciò ha rappresentato una novità ed un motivo di sfida personale per il sottoscritto.

### **3.2.2 Descrizione generale**

L'interfaccia HMI per la macchina che effettua telecontrollo lavora nel seguente modo: essa deve acquisire i punti di un campo dal database, monitorarne lo stato e restituire all'utente, tramite una visualizzazione grafica quanto più in real-time e di immediata comprensione possibile, l'output relativo agli stati/valori delle variabili del campo stesso, in modo che l'utente possa intervenire in caso di problemi, guasti, malfunzionamenti o semplice volontà di modifica di parametri.

Esiste un'altra possibilità fornita all'utente, ovvero la possibilità di “editare” una propria rete e, dopo averla salvata, di monitorarla direttamente; per fare ciò vengono messe a disposizione una serie di widget in formato scalable vector graphics (.svg), alcuni di esse ruotabili in diverse posizioni, che potranno poi essere trascinati sulla “griglia” nella quale verrà composta la rete.

In seguito verranno fornite informazioni più dettagliate riguardo tutte le possibili opzioni disponibili utilizzando questo software.

### 3.2.3 Obiettivi

Nella concezione in cui il software è stato sviluppato, i suoi obiettivi risultano essere:

- Possibilità di monitoraggio di una rete dopo averne acquisito i punti;
- Visualizzazione dei dettagli di ogni singolo punto/ componente semplicemente al passaggio del mouse;
- Possibilità di creare una propria rete data una serie di componenti e, in seguito, di monitorarla;
- Aggiornamento buffer eventi ciclico, con coda FIFO avente capacità di 2000 record.
- Operazioni di login e logout di un utente all'interno del sistema (operazioni già presenti allo stato attuale delle cose nel sistema e lasciate inalterate).

### 3.2.4 Glossario

Di seguito il glossario dei termini utilizzati e il loro significato all'interno del presente documento:

- Campo/ Impianto: oggetto dell'attività di monitoraggio da parte della macchina e sede dei processi.
- Evento: variazione di stato di un ingresso o variazione di una soglia di una misura alla quale viene associato un attributo orario secondo un sistema di riferimento preciso e univoco.
- Utente: persona autorizzata previa riconoscimento o login e che quindi può avere accesso alle funzionalità della macchina.

- RTU: acronimo di Remote Terminal Unit, dispositivo elettronico a microprocessore situato in un punto remoto rispetto ad un processo, in cui viene concentrata l'informazione del processo stesso. L'RTU rileva l'informazione di processo che viene inviata al centro di controllo per poter essere elaborata e visualizzata dall'utente che effettua il monitoraggio del processo

- HMI: acronimo di Human Machine Interface, strato che separa l'essere umano dalla macchina che esso sta utilizzando e che mette in contatto le due parti mediante messaggi visivi (in genere forniti da un monitor), messaggi sonori (diffusi mediante altoparlanti, sirene, ricetrasmittenti) e azioni di controllo (per mezzo di tastiere, pulsanti, mouse, interruttori). Un sottoinsieme dell'HMI è rappresentato dalle GUI (Graphical User Interface), che consentono all'utente di interagire con la macchina in maniera user-friendly attraverso la gestione e manipolazione di elementi grafici; quest'ultima tipologia di interfaccia è quello che si è cercato di ottenere con il presente lavoro.

- FIFO: Acronimo di First-In-First-Out: in una coda con capacità finita indica la politica di sovrascrittura, ovvero a partire dall'elemento che è in coda da più tempo.

- Rete: L'accezione con cui viene usato il termine in questo capitolo corrisponde a quanto descritto nel capitolo precedente in merito alla sottostazione elettrica.

- Widget: componente grafico di un'interfaccia utente, che ha lo scopo di migliorare l'interazione tra l'utente e il programma.

### **3.2.5 Panoramica**

Il resto del capitolo è organizzato nella seguente maniera:

- Sezione 3, Vincoli e specifiche del sistema: descrive quali sono i requisiti, sia funzionali che non funzionali, che il software deve obbligatoriamente avere e ne dà una descrizione esaustiva; comprende inoltre una descrizione delle limitazioni che ha il sistema e che non sono in alcun modo evitabili, nonché di tutte le sue caratteristiche che possono in qualsiasi modo influenzare lo sviluppo del presente software e delle assunzioni effettuate in fase preliminare.

- Sezione 4, Definizione di una soluzione proposta: questa sezione contiene scenari, casi d'uso e modelli dell'interfaccia, oltre ad una sottosezione in cui si identificano i principali attori che entrano in contatto con il software e ad un'altra sottosezione dedicata alla descrizione di attributi propri del software stesso. Infine, nella sezione è presente anche un'analisi, fatta a priori e con il metodo Function Point, della stima dei costi necessari per l'intero lavoro.

### **3.3 Requisiti, limitazioni e assunzioni del sistema**

Il sistema proposto ha senso di esistere unicamente come interfaccia grafica alla macchina STCE/RTU di Selta SpA, sulla quale può essere installato nella maniera spiegata in seguito. Ogni altro utilizzo del software in questione è privo di significato, oltre che non autorizzato.

#### **3.3.1 Requisiti funzionali**

Il sistema presenta i seguenti requisiti funzionali, i quali devono essere rispettati in ogni singola parte dal software in oggetto.

### **3.3.1.1 Acquisizione punti e loro stato da un database di campo**

Il sistema deve necessariamente permettere l'acquisizione da database, su richiesta dell'utente, dei punti che formano il campo e delle loro caratteristiche peculiari, in modo da consentire un'efficace operazione di monitoraggio.

Nello specifico, le tipologie di punti e le rispettive caratteristiche da ottenere, in modo assolutamente trasparente all'utente, se non per quanto riguarda l'animazione finale della rete, sono:

- Single Point (SP), di cui bisogna poter conoscere valore e qualità; il primo è caratterizzato da una variabile booleana e quindi può assumere solamente i valori 0 oppure 1, mentre la qualità è definita con 0 quando è buona e con un intero diverso da 0 quando non è buona;
- Double Point (DP), di cui bisogna poter conoscere la qualità, definita come per il Single Point, e il valore, caratterizzato da una variabile booleana a 2 bit, che quindi può assumere 4 possibili valori: 00 (stato indeterminato), 01 (off), 10 (on), 11 (stato incongruente);
- Misura (ME), di cui bisogna poter conoscere la qualità, definita come per il Single Point, la natura (intero oppure float) e il valore (coerente con la natura e compreso tra  $-2^{31}$  e  $2^{31}$ );
- Step Point (ST), caratterizzato allo stesso modo del DP;
- BitString (BS), di cui bisogna poter conoscere la qualità, definita come per il Single Point e il valore (qualsiasi intero nel range tra 0 e  $2^{32}-1$ ).

### **3.3.1.2 Editor del proprio campo personalizzato**

Il sistema deve permettere all'utente la creazione di una propria rete dati alcuni widget, creati con grafica vettoriale, eventualmente ruotabili e posizionabili in una griglia di dimensioni massime 20x20; nei capitoli successivi verranno elencati e descritti i vari widget corrispondenti ai vari componenti della rete e le (eventuali) rotazioni che sarà possibile effettuare su di essi, oltre ai possibili stati che tali componenti potranno raggiungere una volta creata e salvata la rete.

### **3.3.1.3 Aggiornamento buffer degli eventi**

Ad intervallo stabilito dall'utente, e comunque sufficientemente piccolo, il sistema deve effettuare il refresh delle informazioni ottenibili dai punti e, nel caso si siano verificati cambiamenti dall'ultimo refresh, aggiornare il buffer degli eventi con indice di ordine cronologico, descrizione dell'evento e timestamp corrispondente alla data. Il buffer degli eventi è consultabile solamente quando l'attività di monitoraggio è in corso, viceversa il clic sull'icona corrispondente al buffer degli eventi non avrebbe alcun effetto.

### **3.3.1.4 Monitoraggio**

Alla pressione di un bottone da parte dell'utente, e in ogni caso dopo l'acquisizione degli stati dei vari punti, il sistema deve permettere l'avvio dell'operazione di monitoraggio, che consente di visualizzare eventuali errori e/o anomalie all'interno del sistema, assegnando colori diversi ad ogni tipologia di anomalia, come stabilito da legenda.

Come spiegato al punto 3.3.1.3 il sistema deve refresharsi in automatico, per cui è possibile che la colorazione dei vari widget possa cambiare a runtime, in seguito ad eventi esterni.

L'utente può arbitrariamente decidere di interrompere il monitoraggio, cliccando sull'apposito pulsante, salvo poi riprenderlo in qualsiasi momento.

### **3.3.1.5 Visualizzazione dettagli**

Passando con il mouse sul singolo componente della rete è possibile visualizzare in una finestra popup il dettaglio del componente e delle sue caratteristiche più importanti.

### **3.3.2 Requisiti non funzionali**

L'applicazione si basa su un'architettura di tipo client-server, che è divisa in due parti distinte ed indipendenti: il lato server è quello in cui viene fornito il servizio, mentre il lato client è la parte utilizzata dall'utente per ottenere il servizio richiesto.

Il linguaggio utilizzato è *JavaScript* <sup>[5]</sup>, in modo particolare la libreria *Jquery* <sup>[6]</sup> nella versione *jquery-2.2.3*, in combinazione con gli standard *HTML5* e *CSS3*, il database usato è quello proprietario di Selta SpA.

L'IDE utilizzato per lo sviluppo del software è *Eclipse Java Mars*, mentre l'application server utilizzato per lo sviluppo è *WildFly 8.1* <sup>[7]</sup>. Per l'utilizzo di questa interfaccia è richiesta una connessione internet stabile e un web browser recente, che possa supportare lo standard *HTML5* e con *Javascript* abilitato.

### 3.3.2.1 Interfacce di sistema, interfaccia utente e page sketches

Per poter utilizzare correttamente il sistema ed interagire con la GUI, sono necessarie le periferiche standard di input, ovvero il mouse e la tastiera. Non esistono attualmente estensioni specifiche che supportano il touchscreen, né sono in programma in un futuro a breve termine.

L'obiettivo del lavoro è quello di creare un'interfaccia semplice e di immediata comprensione; la nostra interfaccia utente deve prevedere la possibilità di soddisfare ognuno dei requisiti funzionali descritti in precedenza, tramite la generazione di pagine web dinamiche con funzionalità differenti. Si specifica che questa parte dell'elaborato è stata stilata in fase progettuale, e quindi non rappresenta la struttura definitiva del software in questione, che potrà pertanto subire cambiamenti, i quali verranno ovviamente documentati in corso d'opera.

Il software progettato prevede una pagina utente iniziale in cui l'utente riconosciuto dal sistema visualizza le proprie credenziali di accesso e ha a disposizione una prima possibilità di scelta, tra l'editing di una nuova rete e il monitoraggio di una rete preesistente, come si può vedere dal page sketch di seguito:

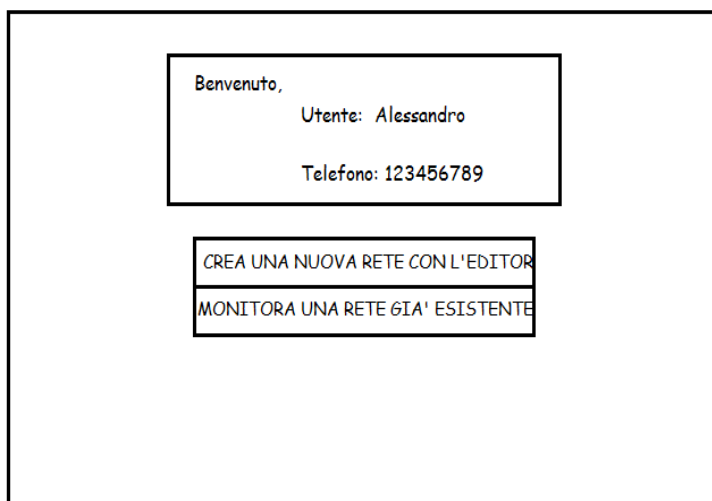


Figura 3.1: page sketch della pagina principale/ pagina utente



Un'altra pagina generabile dalla nostra interfaccia è la pagina di monitoraggio, che prevede una legenda, con la quale si possono associare i vari colori alle widget, i pulsanti di inizio e fine monitoraggio, il pulsante per accedere al buffer degli eventi e la rete da monitorare, come visibile dal page sketch in figura 3.2:

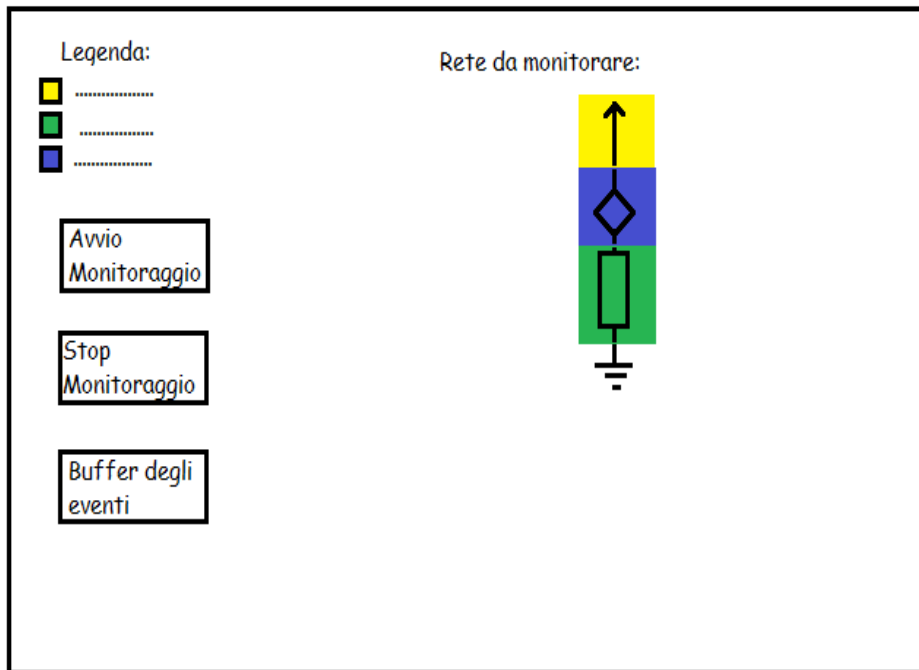


Figura 3.2: page sketch della pagina di monitoraggio

Infine si può generare, tramite popup dalla rete in fase di monitoraggio, una pagina di dettaglio, che permette la visualizzazione della widget del componente in questione e le caratteristiche salienti del punto del database corrispondente al componente stesso.

Nella figura 3.3 della pagina seguente è riportato un semplice page sketch della pagina di dettaglio:

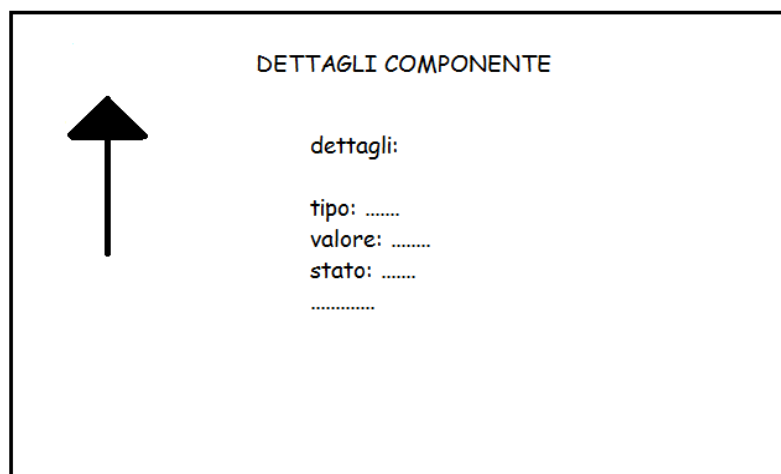


Figura 3.3: page sketch della pagina di dettaglio

### **3.3.2.2. Requisiti di database e di performance**

Il software si appoggia sui database propri di SELTA SpA e già in uso attualmente sulla macchina STCE/RTU; i requisiti di database sono quelli classici, posti a salvaguardia delle proprietà del paradigma ACID, ovvero:

- Assenza di duplicati in qualsiasi campo del database
- Assenza di dati inconsistenti, in qualsiasi campo del database
- Presenza, per ogni tupla contenuta nel database, di un identificativo numerico univoco, che funge da chiave primaria
- E' compito dell'application server, in maniera back-end, e quindi trasparente all'utente, il controllo della consistenza e della correttezza dei dati inseriti
- Qualsiasi operazione che viola uno qualsiasi o più dei vincoli sopra descritti deve essere abortita dal sistema prima di arrivare al commit

Per quanto riguarda nello specifico i requisiti di performance, non sussistono altre specifiche diverse da quanto già indicato.

### 3.3.3 Vincoli di design e conformità agli standard

Come già specificato in precedenza, nella progettazione del software ci si è rifatti ad alcuni standard, in particolare per quanto riguarda i linguaggi di markup :

- HTML5 - <http://www.w3.org/TR/html5/>
- XHTML – [www.w3.org/MarkUp/](http://www.w3.org/MarkUp/)
- CSS3 - <http://www.w3.org/Style/CSS/>

### 3.3.4 Assunzioni e limitazioni

In questa sezione vengono presentate tutte quelle caratteristiche del sistema e del software che, pur non rappresentando alcun tipo di requisito, rivestono una certa importanza nello sviluppo dell'interfaccia e delle quali è opportuno fare un breve cenno.

- Il fatto che l'utente, in fase di monitoraggio, rilevi e notifichi un'anomalia o uno stato di errore di un componente non ne implica la sua risoluzione: la gestione degli errori è delle anomalie è in capo all'utente che le gestirà in modo autonomo rispetto al software, quest'ultimo visualizzerà solamente il rientro dell'allarme o della situazione anomala tramite colorazione del componente sul monitor una volta che la situazione sarà ritornata di normale funzionamento;

- La creazione autonoma della rete non presenta vincoli di posizionamento tra i vari componenti; si lascia all'utente e al suo buonsenso la creazione di una rete formalmente corretta e quindi “monitorabile”, diversamente ne verrebbe meno il senso dell'utilizzo del software e quindi della macchina;

- La dimensione massima della rete editabile è di 20x20, per cui si possono monitorare un massimo di 400 punti contemporaneamente, nonostante la struttura della macchina, come si vedrà nei capitoli successivi, consenta l'acquisizione e il monitoraggio di 10000 punti, tra interni, esterni e calcolati.

### 3.3.5 Caratteristiche di funzionamento

Nella tabella seguente sono riportate le caratteristiche ambientali per cui è garantito il funzionamento del macchinario; all'esterno dei range specificati non si garantisce il corretto funzionamento <sup>[4]</sup>.

<b>Caratteristica</b>	<b>Range specificato</b>
Temperatura	-5°C / + 55°C
Massimo gradiente di variazione della temperatura	0,5 °C/minuto
Umidità relativa	5/95%
Umidità assoluta	1 g/m <sup>3</sup> / 29 g/m <sup>3</sup>
Pressione barometrica	66kPa / 108 kPa

*Tabella 3.1: Range di funzionamento della macchina*

### 3.4 Soluzione proposta e sua formalizzazione

In questa sezione si presenta la soluzione che si è pensato di sviluppare per far fronte al problema in oggetto e per apportare migliorie significative al sistema; considerando il fatto che si è ancora in fase pre-progettuale, è opportuno ricordare che la soluzione definitiva al problema potrà presentare

caratteristiche lievemente diverse da quelle di seguito specificate, ma non si discosterà comunque più di tanto da quella proposta in questa sezione; tali lievi discrepanze, nel caso, verranno ampiamente citate e documentate nei capitoli seguenti.

### **3.4.1 Identificazione e descrizione dei principali attori del sistema**

Per l'interfaccia in questione abbiamo un solo tipo di attore, ovvero l'utente che ha accesso a tutte le funzionalità previste dal sistema dopo aver effettuato l'accesso o il riconoscimento; ogni altro utente che non sarà stato riconosciuto dal sistema o per cui l'accesso non avrà avuto esito positivo non avrà alcuna possibilità di interazione con la macchina.

### **3.4.2 Analisi function point**

Questa tipologia di analisi dell'effort speso per la stesura del codice prevede di valutare cinque tipologie di interazione che presenta il sistema e, per ciascuna di esse, di calcolarne la complessità, facendo riferimento ad una scala con tre possibilità: S (Simple), M (Medium), C (Complex).

Le categorie che devono essere prendere in analisi sono le seguenti:

- #inputs: numero delle interazioni con l'utente;
- #outputs: numero di output forniti all'utente;
- #inquiries: numero delle informazioni esterne che devono essere necessariamente immagazzinate;
- #internal logical files: numero dati interni salvati direttamente sull'applicativo;
- #external logical files: numero di dati esterni che il sistema deve necessariamente acquisire.

Tutti questi dati vengono poi “*parametrizzati*”, seguendo una tabella predefinita:

Parametro / Tipo	Simple	Medium	Complex
# Inputs	3	4	6
# Outputs	4	5	7
# Inquiries	3	4	6
# Internal Logical Files	7	10	15
# External Logical Files	5	7	10

*Tabella 3.2: Tabella generica per l'analisi di tipo Function Point*

Per ogni categoria di parametri, si effettua quindi la somma pesata dei valori, moltiplicando la cardinalità dell'operazione per la sua complessità, dopodiché si calcola l'UFP, definito semplicemente come la somma delle somme pesate delle varie categorie.

Come suggerirebbe l'acronimo di UFP (Unadjusted Function Point), in linea teorica il valore ottenuto andrebbe “normalizzato”, moltiplicandolo per alcuni coefficienti che variano in base a 14 parametri propri del sistema, ma spesso tale soluzione, oltre a non risultare di immediata computazione, non è adatta e quindi in questo caso non verrà praticata. Concludendo, si può quindi fare una valutazione sul valore di UFP, che sarà ovviamente tanto maggiore, quanto saranno maggiori la dimensione e la complessità del sistema in analisi e che dovrà essere confrontato con altri sistemi di dimensioni simili per valutarne l'effettiva incidenza, ma presenterà sempre una criticità, ovvero quella di essere un tipo di analisi a carattere fortemente soggettivo.

Effettuiamo ora l'analisi relativa al caso del nostro sistema:

- Internal Logical Files (ILF): il sistema deve avere informazioni riguardo i punti e gli utenti; se per quest'ultima categoria la struttura è molto semplice, per quanto riguarda i punti è necessario avere informazioni riguardo diverse sottocategorie, per cui questa operazione può essere definita come complessa.

In conclusione, per questa sezione avremo  $1 \times 7 + 1 \times 15 = 22$  FPs

- External Logical Files (ELF): non esistono operazioni riconducibili a questa sezione, quindi il totale dei Fps per questa sezione è pari a 0
- Inquiries: non esistono operazioni riconducibili a questa sezione, quindi il totale dei Fps per questa sezione è pari a 0
- Inputs: il cliente deve poter interagire con il sistema per quanto riguarda le seguenti funzionalità: login e logout, scelta della modalità di funzionamento, avvio e interruzione del monitoraggio, consultazione del buffer degli eventi, visualizzazione dei dettagli del componente. Tutte queste sette operazioni possono essere considerate come semplici, in quanto non richiedono procedimenti particolarmente complessi. Inoltre, nel caso di rete editata dall'utente, è prevista anche l'operazione di creazione della rete, questa volta di media complessità. In totale, per questa sezione avremo  $7 \times 3 + 1 \times 4 = 25$  FPs
- Outputs: l'output del sistema è la visualizzazione grafica di ciò che viene monitorato; questa operazione è abbastanza complessa perché richiede diversi tipi di informazioni, e quindi possiamo considerare per questa sezione un totale di  $1 \times 7 = 7$  FPs

In totale abbiamo, quindi,  $22 + 0 + 0 + 25 + 7 = 54$ , che è il valore dell'UFP per l'intero sistema.

Senza un termine di paragone valido con il quale poter effettuare il confronto, il valore ottenuto non ha molto senso, ma come spiegato in precedenza non esistono elementi simili, in quanto il sistema progettato rappresenta un'assoluta novità.

### 3.4.3 Scenari

Di seguito vengono presentati gli scenari più significativi che possono configurarsi all'utente durante l'utilizzo del software; essi coprono pressoché la totalità delle funzionalità fruibili dall'utente con l'interfaccia progettata.

#### Scenario di creazione della rete:

L'utente, nella pagina di editor della rete, dispone le widget prescelte sulla griglia 20x20, in modo da formare una rete di senso compiuto, dopodiché, cliccando sul pulsante “salva” e confermando la scelta, effettua il salvataggio della propria rete, in modo da poterla monitorare in seguito.

#### Scenario di reset della rete in fase di creazione:

L'utente, mentre sta creando la propria rete, si accorge di avere sbagliato il posizionamento di un componente sulla griglia, per cui clicca sul pulsante “reset”, in modo da ottenere la griglia nuovamente vuota e poter così procedere con l'operazione descritta nello scenario di creazione della rete.

#### Scenario di avvio monitoraggio:

Nella pagina di monitoraggio, l'utente clicca sul pulsante “avvia monitoraggio” e dà avvio all'operazione di monitoraggio.

#### Scenario di stop del monitoraggio:

Sempre nella pagina di monitoraggio, e con la funzione di monitoraggio attiva, l'utente clicca sul pulsante “stop monitoraggio” e interrompe il monitoraggio che era in corso.



### Scenario di visualizzazione del cambio colore di un componente avente stato indeterminato:

Con la funzione di monitoraggio attiva, a runtime, e quindi senza interazioni da parte dell'utente, lo stato di un componente cambia in seguito ad un evento di campo, assumendo valore indeterminato, e quindi il corrispondente widget si colora di verde.

### Scenario di visualizzazione del cambio colore di un componente avente stato incongruente:

Con la funzione di monitoraggio attiva, a runtime, e quindi senza interazioni da parte dell'utente, lo stato di un componente cambia in seguito ad un evento di campo, assumendo valore incongruente, e quindi il corrispondente widget si colora di blu

### Scenario di visualizzazione del cambio colore di un componente avente qualità del dato non buona:

Con la funzione di monitoraggio attiva, a runtime, e quindi senza interazioni da parte dell'utente, la qualità di un componente cambia in seguito ad un evento esterno di campo, assumendo valore diverso da zero, e quindi il corrispondente widget si colora di giallo.

### Scenario di visualizzazione del dettaglio di un componente:

Passando il mouse su un componente della rete nella pagina di monitoraggio, il sistema fa aprire un popup nel quale vengono visualizzati i

dettagli del componente stesso, consistenti nell'immagine relativa al componente e in una breve descrizione testuale dello stesso.

#### Scenario di visualizzazione del buffer eventi:

Con la funzione di monitoraggio attiva, cliccando sull'icona raffigurante un calendario con la scritta "eventi", il sistema apre un popup nel quale viene visualizzato il buffer degli eventi, definito come spiegato nella sezione relativa ai requisiti funzionali.

#### Scenario di sovrascrittura della rete:

In seguito al cambiamento della rete da monitorare, l'utente può tornare alla pagina iniziale e, da lì, recarsi alla pagina di editing della rete e creare una nuova rete da monitorare, come descritto nello scenario di creazione di una nuova rete.

### **3.4.4 Use Cases e Sequence Diagrams**

In questa sezione vengono presentati i casi d'uso del sistema in corso di progettazione, ciascuno corredato da un sequence diagram, per ottenere una comprensione migliore delle varie interazioni tra l'utente e il sistema.

### 3.4.4.1 Visualizzazione informazioni nella pagina utente

Titolo	Visualizzazione informazioni nella pagina utente
Attori	Utente
Condizioni di input	L'utente si trova nella pagina personale
Flusso degli eventi	<ul style="list-style-type: none"><li>• L'utente naviga nella propria pagina personale</li><li>• L'utente visualizza le proprie credenziali</li></ul>
Condizioni di output	Nessuna
Eccezioni	Nessuna

Tabella 3.3: Tabella descrittiva dello use case "Visualizzazione informazioni nella pagina utente"

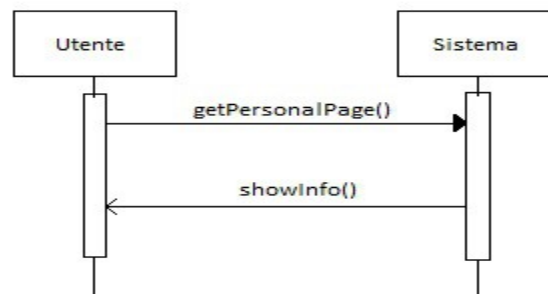


Figura 3.4: Sequence diagram relativo alla visualizzazione pagina personale dell'utente.

### 3.4.4.2 Creazione di una nuova rete

Titolo	Creazione di una nuova rete
Attori	Utente
Condizioni di input	L'utente si trova nella pagina personale
Flusso degli eventi	<ul style="list-style-type: none"><li>• L'utente clicca su “crea la tua rete”</li><li>• L'utente ruota i componenti che necessitano di una rotazione prima di essere piazzati sulla griglia</li><li>• L'utente piazza le widget dei componenti sulla griglia<ul style="list-style-type: none"><li>• L'utente clicca su “salva”</li></ul></li><li>• L'utente conferma il salvataggio cliccando su “ok”</li></ul>
Condizioni di output	<ul style="list-style-type: none"><li>• La rete viene creata</li><li>• L'utente viene reindirizzato in una nuova pagina, in cui può scegliere se tornare alla home o monitorare la rete appena creata</li></ul>
Eccezioni	Nessuna

Tabella 3.4: Tabella descrittiva dello use case “Creazione di una nuova rete”

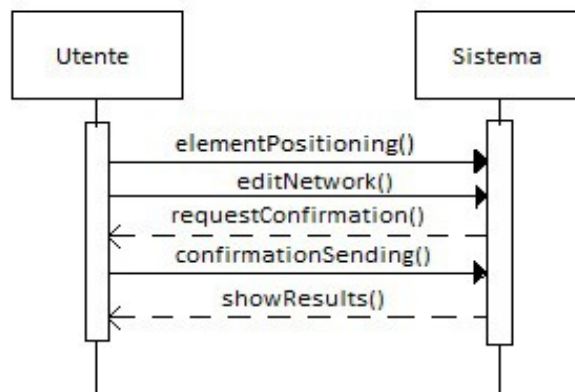


Figura 3.5: Sequence diagram relativo alla creazione di una nuova rete da parte dell'utente.

### 3.4.4.3 Avvio monitoraggio

Titolo	Avvio monitoraggio
Attori	Utente
Condizioni di input	L'utente ha creato/ acquisito i punti di una rete
Flusso degli eventi	<ul style="list-style-type: none"><li>• L'utente raggiunge la pagina di monitoraggio</li><li>• L'utente clicca sul pulsante “avvio monitoraggio”</li></ul>
Condizioni di output	<ul style="list-style-type: none"><li>• L'utente vede “colorarsi” la rete in base alle caratteristiche dei punti</li><li>• In seguito a refresh automatici dell'acquisizione del valore dei punti, il colore dei componenti può cambiare a runtime</li></ul>
Eccezioni	Nessuna

Tabella 3.5: Tabella descrittiva dello use case “Avvio monitoraggio”

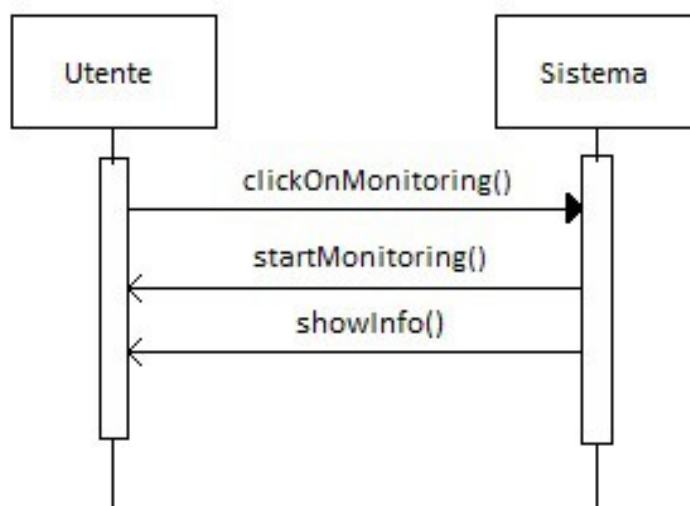


Figura 3.6: Sequence diagram relativo all'operazione di avvio monitoraggio.

#### 3.4.4.4 Stop monitoraggio

Titolo	Stop monitoraggio
Attori	Utente
Condizioni di input	<ul style="list-style-type: none"><li>• L'utente ha creato/ acquisito i punti di una rete</li><li>• L'attività di monitoraggio è in corso</li></ul>
Flusso degli eventi	L'utente clicca sul pulsante "stop monitoraggio"
Condizioni di output	La rete torna ad assumere i colori originali pre-monitoraggio
Eccezioni	

Tabella 3.6: Tabella descrittiva dello use case "Stop monitoraggio"

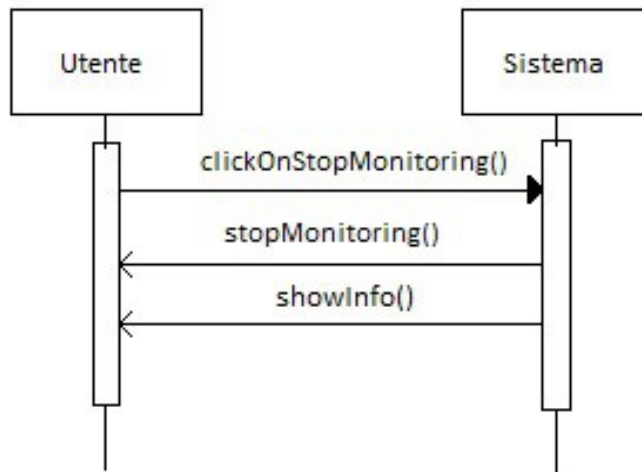


Figura 3.7: Sequence diagram relativo all'operazione di interruzione del monitoraggio.

### 3.4.4.5 Apertura pagina dettaglio

Titolo	Apertura pagina dettaglio
Attori	• Utente
Condizioni di input	<ul style="list-style-type: none"><li>• L'utente ha creato/ acquisito i punti di una rete</li><li>• L'utente si trova nella pagina di monitoraggio</li></ul>
Flusso degli eventi	L'utente passa con il puntatore del mouse sopra un componente della rete oggetto di monitoraggio
Condizioni di output	Si apre un popup in cui vengono visualizzati i dettagli del componente in questione
Eccezioni	Nessuna

Tabella 3.7: Tabella descrittiva dello use case "Apertura pagina dettaglio"

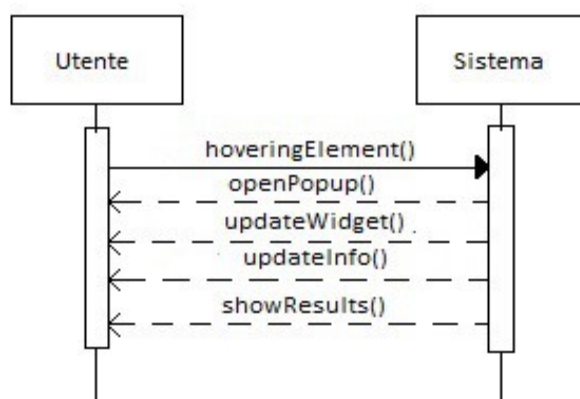


Figura 3.8: Sequence diagram relativo all'apertura, e conseguente visualizzazione, della pagina di dettaglio del componente selezionato.

### 3.4.4.6 Apertura buffer degli eventi

Titolo	Apertura buffer degli eventi
Attori	Utente
Condizioni di input	• L'utente è nella pagina di monitoraggio
Flusso degli eventi	L'utente clicca sull'icona del buffer degli eventi
Condizioni di output	Si apre una pagina contenente gli ultimi 2000 eventi relativi al campo oggetto del monitoraggio, ordinati cronologicamente a partire dal più recente
Eccezioni	L'attività di monitoraggio non è in corso e quindi la pagina del buffer degli eventi non si apre

Tabella 3.8: Tabella descrittiva dello use case "Apertura buffer degli eventi"

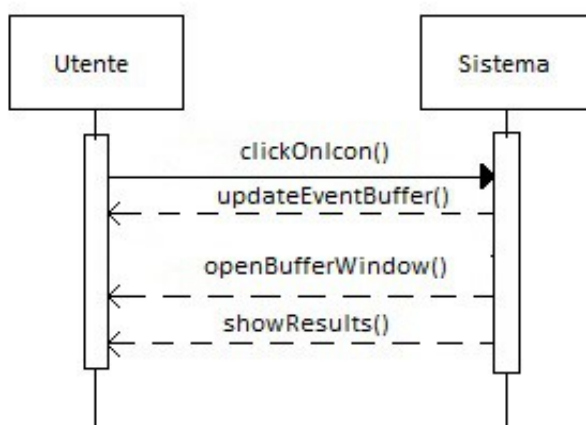


Figura 3.9: Sequence diagram relativo alla visualizzazione e consultazione del buffer degli eventi.



### 3.4.5 Attributi propri del software

Il software è stato progettato per avere, per quanto possibile, la massima affidabilità; per questo motivo il server centrale deve essere messo in un'area dotata di sistemi anti-incendio e anti-allagamento, e di un generatore di emergenza, per mantenere il server attivo anche in caso di blackout elettrico, per evitare crash del sistema dovuti a disastri o ad eventi accidentali.

Il sistema deve inoltre garantire che le transazioni che avvengono sul database rispettino le quattro proprietà ACID, ovvero l'atomicità, la coerenza, l'isolamento e la durabilità (o persistenza) nel tempo; il mancato rispetto di tali proprietà può creare conflitti tra le diverse transazioni e quindi portare al malfunzionamento o al crash del sistema.

Per quanto riguarda l'aspetto della disponibilità, è necessario che il sistema sia disponibile durante tutto il giorno senza alcuna interruzione. Il periodo di operatività del server deve essere il 100% per assicurare efficienza e sicurezza; ciò è reso possibile dalla dualizzazione di tutte le parti centralizzate della macchina per cui è stata progettata l'interfaccia

Un altro aspetto importante, infine, è quello della manutenibilità del sistema: in questo caso l'obiettivo da raggiungere è quello di creare un software che sia il più stabile possibile del tempo, e che quindi non debba essere oggetto di modifiche a breve termine. La scelta implementativa effettuata è stata quella di scrivere un codice che fosse il più semplice e lineare possibile, in modo da facilitare le operazioni di modifica del codice stesso, qualora esse si dovessero rendere necessarie.

## Capitolo 4: Specifiche e proprietà del software realizzato

Come ampiamente accennato in precedenza, lo scopo del nostro lavoro è quello di integrare e migliorare le funzionalità della macchina STCE/RTU <sup>[4]</sup> per il monitoraggio, e in particolare della sua interfaccia grafica, in modo da ottenere risultati significativamente migliori, soprattutto dal punto di vista dell'immediatezza e della località della visualizzazione, rispetto agli standard attuali.

Questo capitolo, in cui verrà presentato il software nel dettaglio, è strutturato in diverse sezioni:

Nella prima sezione, prima di procedere con la descrizione vera e propria del software è necessario discutere e motivare alcune scelte implementative, soprattutto a livello di tecnologie, linguaggi di programmazione e supporti usati.

Nella sezione successiva viene fatta una panoramica sui vari componenti che sono nella disponibilità dell'utente al momento di creare la rete, ognuno di essi corredato dal widget corrispondente, spiegando come funzionano i vari componenti e quali possono essere le interazioni tra loro.

Nella terza sezione si specifica qual è lo schema architetturale del software che verrà proposto, partendo da una descrizione più generale dello schema architetturale.

In seguito vengono definite la soluzione lato client, con le caratteristiche minime che deve avere un qualsiasi dispositivo client per poter utilizzare correttamente il client, e le specifiche ed i vincoli dell'architettura server della macchina.

Nella sesta sezione si specificano quali sono le modifiche effettuate in fase di sviluppo del codice di quelli che erano i requisiti stilati in fase preliminare, analizzandone le motivazioni e l'effettivo beneficio che tali modifiche portano all'utilizzo della macchina.

In seguito si forniscono all'utente le nozioni per poter avviare il software e si effettua una descrizione più dettagliata del sistema stesso.

Infine, nelle ultime due sezioni, sono riportati, al fine di una comprensione ancora maggiore della totalità del sistema e delle sue funzionalità, lo use case diagram e il diagramma di flusso relativi alle varie funzionalità proprie del software.

#### **4.1. Scelte implementative**

Nello sviluppo del codice proposto è stato utilizzato il linguaggio di scripting JavaScript<sup>[5]</sup>; questa scelta è stata effettuata, oltre che su consiglio del committente del software, anche perché il linguaggio in questione è in forte crescita per quanto riguarda la popolarità (come si può vedere dalla tabella dell'immagine 4.1), e soprattutto per l'integrazione che esso ha con lo standard e il linguaggio HTML. Quest'ultima caratteristica, sottolineata spesso negli anni '90 – celebre il nome DHTML (Dynamic HTML) per definire l'accoppiata HTML-JavaScript – consente di manipolare e modificare le pagine HTML in modo dinamico, in base all'interazione dell'utente con il browser lato client, e quindi senza per forza dover coinvolgere il server. Con l'avvento di HTML5 anche le possibilità applicative del linguaggio sono cresciute notevolmente, anche al di fuori del semplice Web Browser. Attualmente JavaScript non ha più il solo compito di collante tra codice HTML ed utente, ma può essere utilizzato anche lato server, ad esempio per applicazioni desktop e mobile. Un'altra ragione che ha portato all'utilizzo del linguaggio JavaScript è stata quella che ormai tutti gli ambienti di sviluppo integrati, in particolare Eclipse, quello utilizzato nel nostro caso con la versione Mars, prevedono un supporto specifico per il linguaggio.

Non avendo particolari preferenze, ma sempre nella necessità di mantenere il codice quanto più pulito e leggibile possibile, si è alternato il modo di inserimento del codice JavaScript all'interno delle pagine HTML, utilizzando sia codice *inline* per link e attributi di eventi, sia blocchi di codice all'interno dell'usuale tag `<script>` per codice di complessità maggiore o codice contenente definizioni di variabili e funzioni, sia codice proveniente da file JavaScript esterni, come nel caso dell'utilizzo di alcune librerie. Quest'ultimo approccio ha il vantaggio di collegare *script* e librerie alla pagina HTML in modo non intrusivo, mantenendo quindi una netta separazione tra la struttura del documento ed il codice.

Per la visualizzazione a livello grafico, sia per quanto riguarda i widget che per gli altri elementi grafici inseriti nelle pagine HTML, si è utilizzato un linguaggio di grafica vettoriale, nel quale un'immagine è descritta mediante un insieme di primitive geometriche, quali punti, linee, curve, poligoni, che possono essere colorate o sfumate a piacere.

Nel dettaglio, si è ritenuto di utilizzare il linguaggio di grafica vettoriale SVG <sup>[8]</sup> piuttosto che un qualunque formato grafico di tipo raster (ad esempio bitmap, gif, jpeg) per i seguenti motivi:

- Un file .svg è comodamente generabile e manipolabile tramite un comunissimo editor di testo, e inoltre un file testuale si può comprimere in modo molto efficiente favorendo l'utilizzo di SVG in ambito Web;
- SVG è un linguaggio grafico di tipo vettoriale; ciò implica la possibilità di zoommare o scalare a piacimento l'immagine senza perdere in qualità, a differenza di quanto accade in un qualsiasi formato grafico raster;
- SVG deriva da XML ed è una sua applicazione, per cui nello sviluppo di applicazioni che manipolano file SVG si possono facilmente ereditare gli svariati strumenti di sviluppo già esistenti per XML;
- SVG è interattivo, per cui tramite un linguaggio di scripting – come appunto può essere JavaScript – si può rendere l'immagine SVG interattiva, ovvero fare in modo che sia sensibile a certi eventi (ad esempio il click o il passaggio del mouse), e che in risposta a questi ultimi compia delle azioni predeterminate.

Il database utilizzato è quello già in uso sulla macchina, per cui non presenta vincoli particolari, se non quelli descritti nella sezione 3.3.2.2 del presente documento; nel corso di questo capitolo verrà spiegato in modo chiaro come il database viene interrogato e in che modo vengono associati i suoi punti con i widget della rete costruita dall'utente, oppure come tali punti vengano semplicemente acquisiti nel caso di una rete già esistente e non creata dall'utente.

## 4.2 Elenco dei widget utilizzati e loro significato

In questa sezione elenchiamo e diamo una breve descrizione dei componenti e dei 27 widget ad essi associati inseribili nelle reti da monitorare, nonché delle diverse posizioni che essi possono assumere.

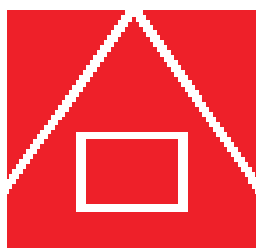


Figura 4.1: Widget componente di allarme

Simbolo di allarme: indica un componente preposto a mandare un segnale di allarme in caso di anomalia riscontrata. Non può essere ruotato.

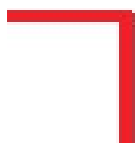


Figura 4.2: Widget componente di collegamento ad angolo retto

Simbolo di angolo: indica un componente della rete disposto ad angolo retto. La sua posizione iniziale è quella in figura (con l'angolo retto posto in basso a sinistra) e può essere ruotato di 90°, 180° o 270°.



Figura 4.3: Widget componente autotrasformatore

Simbolo di autotrasformatore: indica questo tipo di componente all'interno di una rete. Non può essere ruotato.



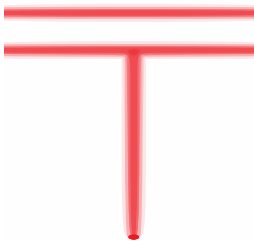
Simbolo di batteria condensatore: indica questo tipo di componente all'interno di una rete. Non può essere ruotato.

Figura 4.4: Widget componente batteria condensatore



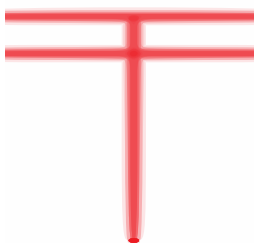
Simbolo di collegamento: indica un componente della rete con funzione di collegamento. La sua posizione originaria è quella orizzontale e può essere ruotato di 90° e quindi assumere anche posizione verticale.

Figura 4.5: Widget di componente collegamento



Simbolo di collegamento con sbarra: indica un componente della rete con funzione di collegamento con una sbarra posta ad un'estremità. La sua posizione originaria è quella con la sbarra sopra al collegamento, ma può essere ruotato di 180° e quindi ribaltato.

Figura 4.6: Widget di componente collegamento con sbarra 1



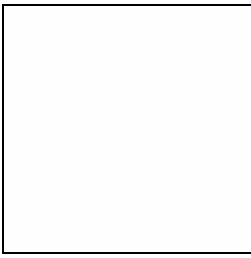
Simbolo di collegamento con sbarra: simile al componente precedente, si differenzia soltanto perché il collegamento raggiunge anche la sbarra superiore. Posizioni e rotazioni sono analoghe a quelle del componente precedente.

Figura 4.7: Widget di componente collegamento con sbarra 2



Simbolo di doppia sbarra: componente che rappresenta una doppia sbarra. La sua posizione iniziale è con la doppia sbarra rivolta verso l'alto, ma può essere ruotato di 180° e quindi ribaltato.

Figura 4.8: Widget di doppia sbarra



Simbolo di componente vuoto: rappresenta un componente vuoto che può essere inserito nella rete.

*Figura 4.9: Widget di componente vuoto a sfondo bianco*



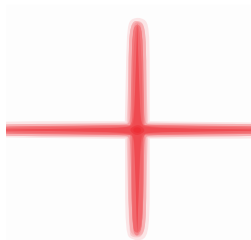
Simbolo di componente vuoto: a differenza del precedente ha sfondo grigio e rappresenta un componente vuoto che può essere riempito ed inserito nella rete.

*Figura 4.10: Widget di componente vuoto a sfondo grigio*



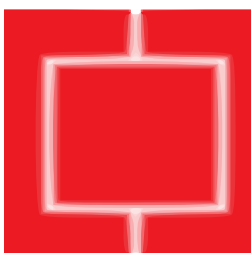
Simbolo di etichetta: rappresenta un etichetta da poter inserire nella rete. Non può essere ruotato.

*Figura 4.11: Widget di etichetta*



Simbolo di incrocio: rappresenta un punto sulla rete in cui due collegamenti provenienti da direzioni perpendicolari si incrociano. Non può essere ruotato.

*Figura 4.12: Widget di componente ad incrocio*



Simbolo di interruttore: rappresenta un interruttore posto sulla rete elettrica. La sua posizione iniziale è verticale, ma può essere ruotato di 90° e quindi assumere posizione orizzontale.

*Figura 4.13: Widget di interruttore*

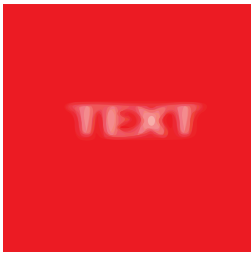


Figura 4.14: Widget di label con testo

Simbolo di label con testo: rappresenta un componente di tipo etichetta sul quale apporre del testo, che può essere posto sulla rete elettrica. Non può essere ruotato.

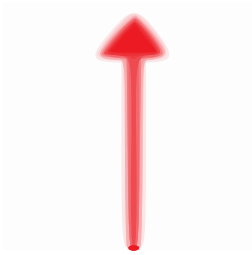


Figura 4.15: Widget di linea

Simbolo di linea: rappresenta un componente di tipo linea che si può sistemare sulla rete. Nella sua posizione iniziale la parte terminale è posta verso l'alto, ma può ruotare di  $180^\circ$  e quindi portarsi in una posizione avente la parte terminale verso il basso.

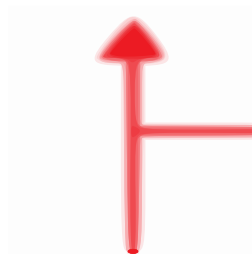


Figura 4.16: Widget di linea con incrocio a destra

Simbolo di linea con incrocio a destra: rappresenta un componente di tipo linea che riceve un collegamento sulla sua parte destra. Non può essere ruotato.



Figura 4.17: Widget di linea con incrocio a sinistra

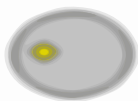
Simbolo di linea con incrocio a sinistra: componente simmetrico rispetto al precedente, rappresenta un componente di tipo linea che riceve un collegamento sulla sua parte sinistra. Non può essere ruotato.



Figura 4.18: Widget di divieto di accesso

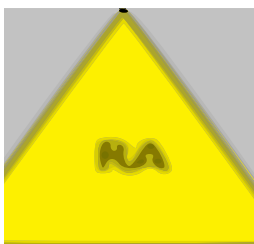
Simbolo di divieto di accesso: rappresenta un indicazione di divieto di accesso che può essere posta nei pressi della rete.





Simbolo di predispositore: rappresenta il componente di tipo predispositore da inserire in una rete. Non può essere ruotato.

Figura 4.19: Widget di predispositore



Simbolo di allarme di impianto, che notifica all'utente la presenza di un allarme. Non può essere ruotato.

Figura 4.20: Widget di allarme di impianto



Simbolo di allarme di impianto 2, ha lo stesso funzionamento del precedente. Anch'esso non può essere ruotato.

Figura 4.21: Widget di allarme di impianto 2



Simbolo di sezionatore: rappresenta un sezionatore all'interno della rete. La sua posizione iniziale è quella verticale, ma può essere ruotato di  $90^\circ$  e quindi posto in posizione orizzontale.

Figura 4.22: Widget di sezionatore



Simbolo di collegamento a "T": rappresenta un collegamento a forma di "T" sulla rete. La sua posizione iniziale è quella mostrata nella figura a lato, ma può essere ruotato di  $90^\circ$ ,  $180^\circ$  o  $270^\circ$ .

Figura 4.23: Widget di collegamento a "T"

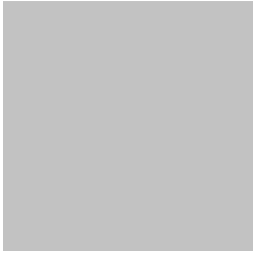


Figura 4.24: Widget di tag vuota

Simbolo di tag vuota: rappresenta una tag vuota da poter inserire nella rete. Non può essere ruotato.

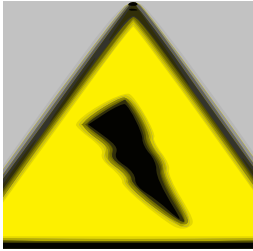


Figura 4.25: Widget con simbolo di tensione

Simbolo di tensione: rappresenta un avviso di tensione da inserire nella rete elettrica. Non può essere ruotato

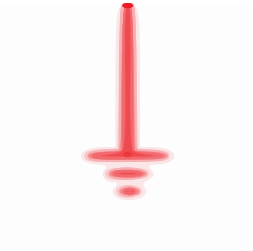


Figura 4.26: Widget con simbolo di massa a terra

Simbolo di massa a terra: rappresenta la massa a terra del sistema. La sua posizione iniziale è quella in figura, con la massa rivolta verso il basso, ma può essere ruotata di 90°, 180° e 270°.

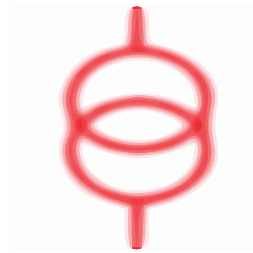


Figura 4.27: Widget di variatore

Simbolo di variatore: rappresenta tale componente all'interno della rete. Non può essere ruotato.

### 4.3 Schema architetturale del sistema proposto

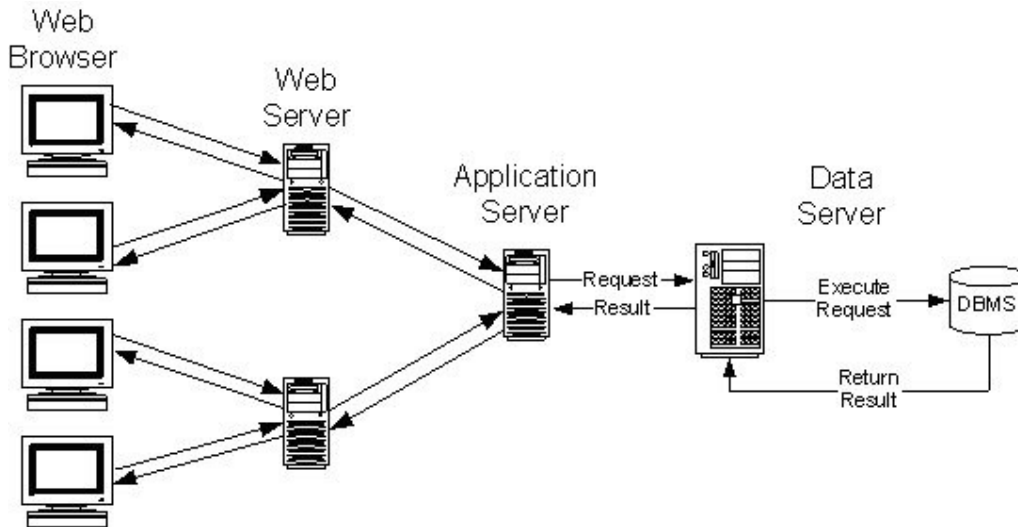


Figura 4.28: Schema dell'architettura client/server del sistema

Nella figura 4.28 è rappresentata l'architettura del sistema proposto, con la migliorata descritta nei capitoli precedenti; la figura definisce solamente il flusso dello scambio di informazioni, il numero dei componenti in figura è puramente indicativo, e nell'applicazione reale dipende dai vincoli che ha la macchina. Si può vedere che dal Web Browser, che nel concreto è ciò che mostra tramite schermo all'utente l'interfaccia grafica oggetto di questo lavoro, ci si interfaccia ad un Web Server, che gestisce le richieste di trasferimento di pagine web del client, in modo da fornirgli, sottoforma di documenti, immagini e qualsiasi altro tipo di informazioni, tutto quanto necessario alla corretta e completa visualizzazione dell'interfaccia.

Il compito del Web Server <sup>[9]</sup> è molto semplice: esso riceve le richieste dal browser, le decifra in modo da determinare quali informazioni sono richieste, verifica se tali informazioni sono disponibili e, in caso affermativo, le inoltra al Web Browser tramite connessione internet. Il programma del Web Server funziona in modo continuo, e si pone in attesa delle richieste degli utenti tramite Internet, creando una connessione per mezzo

della quale riceve le richieste dell'utente.  
La connessione tra browser e Web Server può avvenire in due modi differenti:

- Tramite una rete LAN o WAN, cui l'apparato è connesso mediante la porta ETH dell'unità centrale;
- Mediante un collegamento diretto, dalla porta seriale del dispositivo client alla porta seriale PC dell'Unità Centrale, utilizzando il protocollo SLIP.

Il protocollo SLIP <sup>[10]</sup> (*Serial Line Internet Protocol*) rappresenta il metodo originale sviluppato per le comunicazioni TCP/IP su linee telefoniche seriali, mettendo a disposizione una connessione punto a punto che utilizza il protocollo TCP/IP. Questo protocollo viene implementato come protocollo packet-framing, ovvero con trame che incapsulano pacchetti, comunemente di 1,006 o 1,500 byte, ma con dimensione massima non definita.

Il Web Server è in comunicazione con l'Application Server, sul quale risiede e viene eseguita la *business logic* del software. L'Application Server si occupa anche dell'interfacciamento con il Data Server, comprendente il database che, come spiegato in precedenza, contiene tutte le informazioni necessarie e deve essere interrogato per estrarre e fornire queste informazioni al Web Server e di seguito al client, passando prima tramite l'Application Server.

#### **4.4 Soluzione client-side e caratteristiche minime del sistema client**

Come già detto in precedenza, il lato client dell'applicazione fa in modo che la rappresentazione dell'interfaccia grafica venga delegata ad un comune browser Web, dotato di opportuni plug-in, che deve rendere l'interfaccia HMI in grafica vettoriale, utilizzando il formato di grafica vettoriale SVG.

Per poter utilizzare correttamente il Web Server di apparato, è necessario un personal computer dotato di un Web Browser (Internet Explorer o equivalenti) e avente le seguenti caratteristiche minime:

- CPU Pentium III 250 Mhz;
- Scheda grafica SVGA con risoluzione minima di 1024x768 pixel;
- Memoria RAM 64Mbyte;
- Hard Disk con almeno 100 Mbyte di spazio libero;
- Lettore CD-Rom;
- Porta Seriale RS232;
- Monitor a colori;
- Sistema operativo Microsoft Windows NT 4.0/2000/XP;
- Scheda di rete 10/100 BaseT (in caso di connessione remota)

#### **4.5 Vincoli tecnici e strutturali della macchina**

Prima di di proseguire nella descrizione della soluzione adottata, è necessario specificare i vincoli che possiede il macchinario utilizzato; tali vincoli si definiscono sia a livello di struttura fisica, sia in relazione alla capacità di calcolo o di memorizzazione, e altresì fissandone la possibilità di comunicazione con altri strumenti, quali ad esempio la stampante locale.

Tutti questi vincoli sono riassunti nella tabella di seguito <sup>[4]</sup>.

<b>Vincolo</b>	<b>Valore</b>
Numero massimo di Subtelai (compreso il Subtelaio Base)	8
Numero massimo di gruppi ripetitivi (unità periferiche) equipaggiabili nel Subtelaio Base	7
Numero massimo di gruppi ripetitivi (unità periferiche) equipaggiabili nel Subtelaio Estensione	14
Numero massimo totale di gruppi ripetitivi (unità periferiche) gestiti	98
Modularità massima gruppi ripetitivi	Dipende dalla tipologia di gruppo equipaggiato
Numero massimo di punti di I/O gestiti	6272
Numero massimo di punti seriali gestiti	8000 (se l'apparato è equipaggiato con 4 unità seriali, ho 2000 punti per ciascuna unità seriale)
Numero massimo di punti gestiti (interni, esterni, calcolati)	10000
Eventi inseribili nel buffer di Log locale	2000
Eventi di diagnostica inseribili nel buffer di Log locale	200
Numero vie di comunicazione con centri indipendenti	Fino ad un massimo di 3
Numero stampanti locali	1
Dimensione buffer eventi (uno per centro)	2000

Tabella 4.1: Vincoli della macchina

## **4.6 Modifiche alle specifiche della sezione dei requisiti**

Considerando che la sezione relativa alla definizione dei requisiti è stata formalizzata in una fase preliminare rispetto a quella in cui si è sviluppato il software vero e proprio, in fase di implementazione si è deciso di modificare lievemente quanto definito in precedenza, e più precisamente nei page sketch definiti in quella sezione: passando con il mouse sulla rete in fase di monitoraggio non vengono visualizzati tipo, stato e valore del punto in questione, come specificato nel capitolo 2, ma tali informazioni vengono sostituite da una breve descrizione testuale del componente. Inoltre, il pulsante che permette di aprire il buffer degli eventi è stato sostituito, in fase progettuale, da un'icona a forma di calendario con la dicitura "eventi", per differenziare questa funzionalità da quelle di avvio e stop monitoraggio e per rendere maggiormente *user-friendly* l'interfaccia.

## **4.7 Operazione di avvio e descrizione del sistema**

E' possibile avviare il software semplicemente tramite il click su un pulsante visibile sullo schermo della macchina, che porterà l'utente alla pagina principale dell'applicazione in questione.

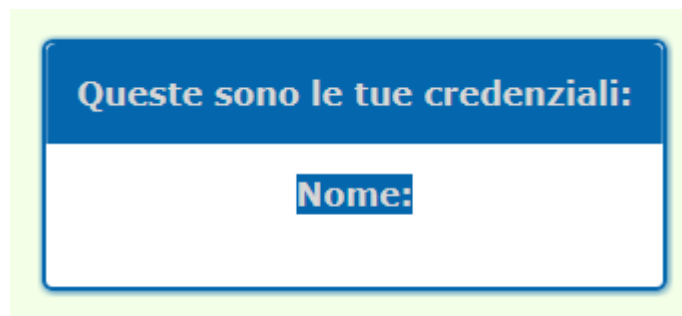
Il sistema sviluppato risulta essere un miglioramento dell'interfaccia utente/macchina per il macchinario di telecontrollo STCE/RTU. Nello sviluppo dell'interfaccia è stata prestata particolare attenzione alla facilità di uso che deve avere il sistema stesso: uno degli obiettivi principali è quello di dare all'utente un servizio che garantisca la semplicità massima possibile dell'interazione con l'utente, in relazione al grado di complessità delle funzionalità offerte dal sistema. L'interfaccia appare quindi decisamente minimale, ma soddisfa in ogni caso tutti i requisiti stilati in fase preliminare.

Di seguito sono elencate e descritte le funzioni e le pagine componenti l'interfaccia in questione, accompagnate da immagini che hanno il compito di chiarificare ulteriormente la descrizione.

## Funzione 1: Scelta della funzionalità

Nella pagina principale della nostra interfaccia, la pagina dell'utente, che viene raggiunta quando le credenziali di accesso dell'utente sono verificate e rese valide, si hanno due blocchi principali:

Nel primo blocco vengono mostrate le credenziali di accesso dell'utente attuale, in particolare lo username, mentre la password non viene mostrata per ovvi motivi di privacy e sicurezza



*Figura 4.29: Parte della schermata della pagina utente che mostra le credenziali dell'utente attivo*

Il secondo blocco invece prevede la presenza di un pulsante che fornisce all'utente la scelta tra le due possibilità descritte ampiamente in fase pre-implementativa: la creazione, e il conseguente monitoraggio, di una nuova rete, oppure l'attività di gestione e monitoraggio di una rete esistente.



*Figura 4.30 Parte della schermata della pagina utente che mostra il bottone di scelta della funzionalità*



## Funzione 2: Editing della rete

Nella pagina di editing della rete si ha la possibilità di editare una nuova rete, a partire dalle 27 tipologie di componenti descritti nei paragrafi precedenti di questo capitolo. Di seguito viene mostrata la struttura della sua pagina e delle sue componenti principali.

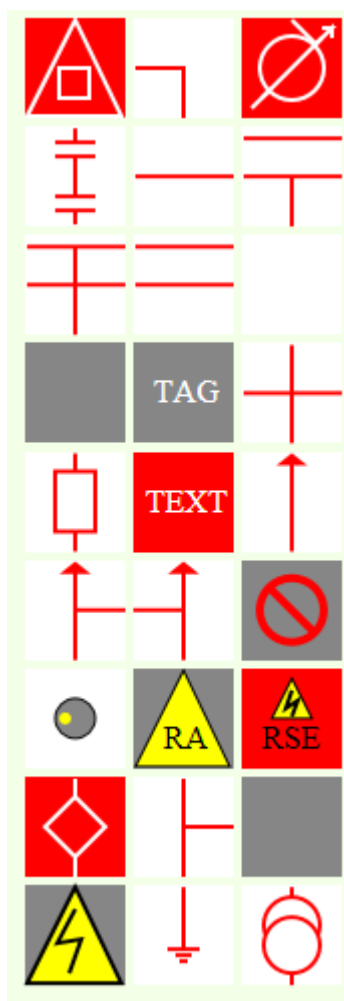


Figura 4.31: Griglia con widget dei componenti

Nell'immagine in figura 4.31 è mostrata la struttura dei widget ai quali si può attingere per creare la rete; questa struttura è posta nella parte sinistra della pagina.

Durante la fase di selezione del componente, al passaggio del mouse sopra il widget, quest'ultimo assumerà uno sfondo di colore nero, come si vede nell'esempio di figura 4.32.

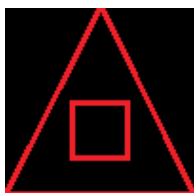


Figura 4.32: Esempio di componente selezionato

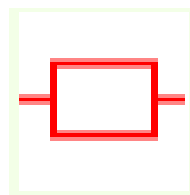


Figura 4.33: Esempio di componente ruotato

In figura 4.33 si può vedere l'esempio di un componente ruotato, azione che accade a seguito di un click del mouse sul componente che tra le sue caratteristiche (si veda il paragrafo 4.2) ha la possibilità di essere ruotato.

L'operazione di composizione della rete si configura con un semplice drag'n'drop del componente selezionato, trasportandolo sulla cella prescelta della griglia 20x20 posta sulla parte destra della pagina e di cui, per motivi di spazio, se ne può vedere solamente una parte in figura 4.34.

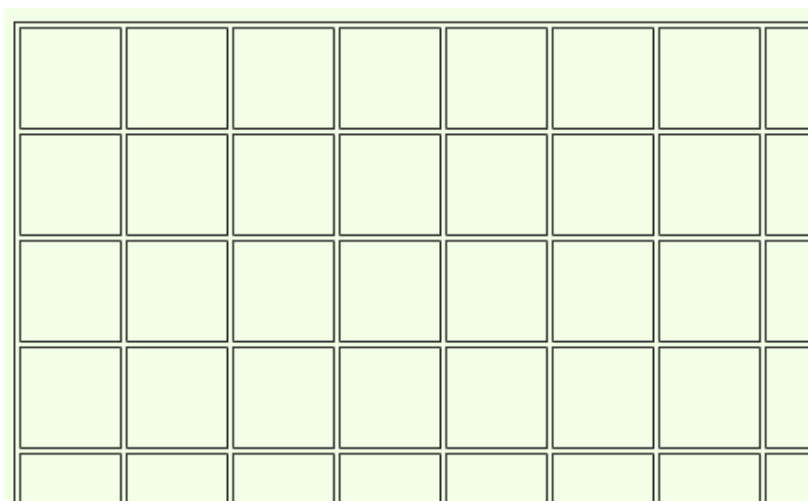


Figura 4.34: Parte della griglia 20x20 sulla quale comporre la rete

Sotto ai widget ci sono inoltre due pulsanti: quello di reset consente di riportare la griglia allo stato originale, togliendo i widget posizionati su di essa fino al momento, mentre quello di salvataggio consente di salvare la rete, una volta ultimata l'operazione di composizione.



Figura 4.35: Pulsanti di reset e di salvataggio della rete

Infine, a titolo di completezza, occorre citare la presenza di una freccia in prossimità del vertice in alto a sinistra della pagina, con funzione di ritorno alla pagina precedente.

### Funzione 3: Acquisizione dei punti

Nella pagina deputata all'acquisizione dei punti sono presenti soltanto due icone: una permette la redirectione dell'utente alla pagina di monitoraggio, l'altra consente il ritorno alla pagina iniziale per operare una scelta differente in fatto di attività da svolgere. L'attività di acquisizione dei punti deve essere svolta dal sistema in modo completamente trasparente all'utente, che deve soltanto rendersi conto della buona riuscita dell'acquisizione ad operazione avvenuta.



*Figura 4.36: Icona di redirectione alla pagina di monitoraggio*



*Figura 4.37: Icona di redirectione alla pagina iniziale*

Nel caso in cui si voglia monitorare una rete già esistente, dalla pagina iniziale si raggiungerà direttamente la pagina di acquisizione punti, senza ovviamente passare dalla pagina di editing della rete.

#### Funzione 4: Avvio ed interruzione del monitoraggio

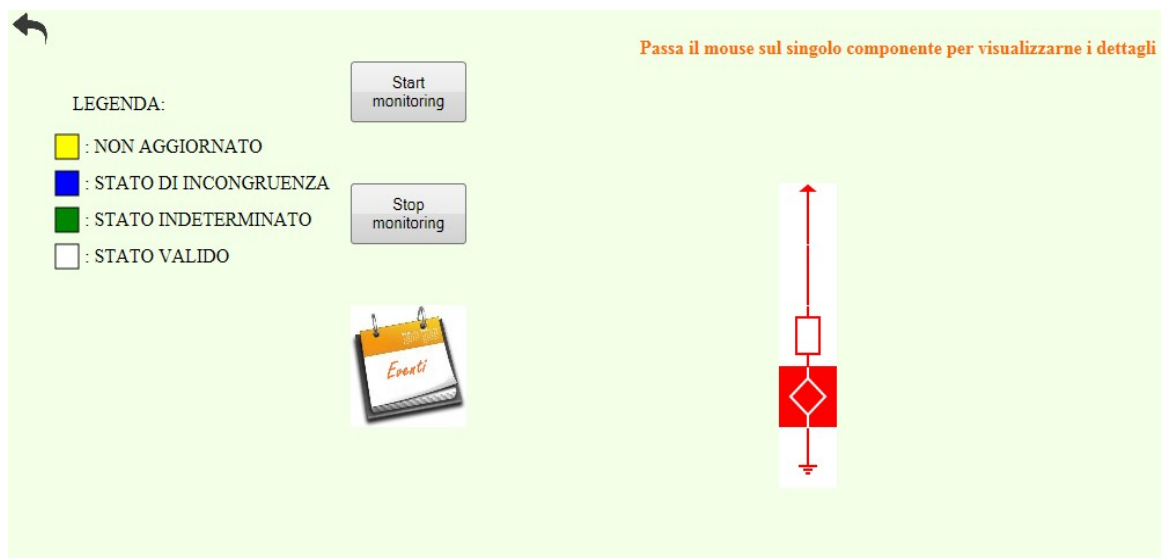


Figura 4.38: Pagina di gestione attività di monitoraggio

In figura 4.38 abbiamo una panoramica generale sulla pagina nella quale vengono compiute le operazioni di monitoraggio; come si può vedere nella schermata sono presenti:

- una freccia in alto a destra che permette di tornare alla pagina precedente;
- la legenda che deve guidare l'utente una volta che il monitoraggio è attivo;
- due pulsanti, uno di avvio e uno di interruzione dell'attività di monitoraggio;
- un'icona che permetterà all'utente di accedere al buffer degli eventi, come descritto nel paragrafo seguente;
- la rete vera e propria da controllare e monitorare.

Cliccando sul pulsante di avvio monitoraggio, i widget relativi ai componenti della rete si “colorano”, in base a quanto definito dalla legenda, a

seconda dello stato del punto e del componente a cui sono associati, come nell'esempio in figura 4.39.

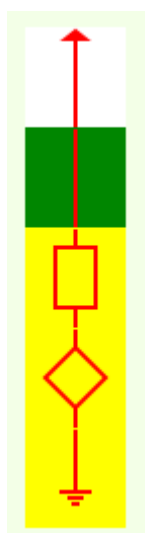
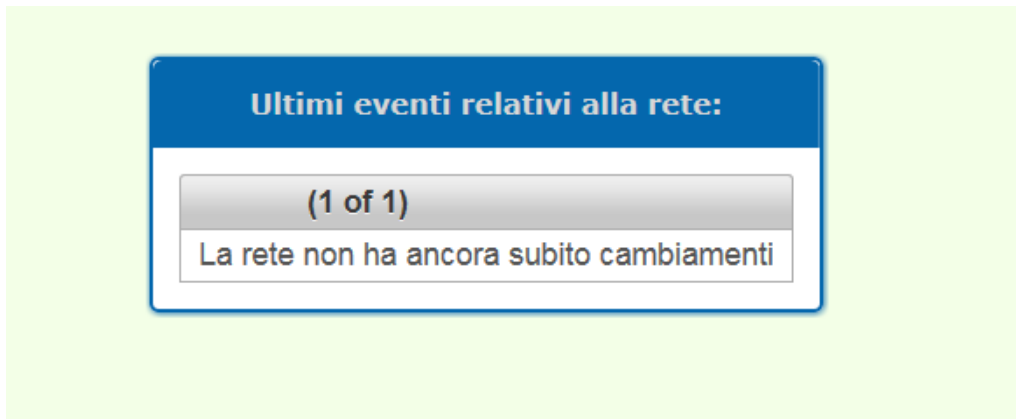


Figura 4.39: Rete in fase di monitoraggio

Una volta avviata la procedura di monitoraggio, è probabile che il colore dei componenti cambi a runtime e quindi senza ulteriori interazioni dell'utente; ciò è dovuto a cambiamenti nel valore degli stati dei punti in campo, che vengono ricalcolati ad ogni refresh periodico effettuato dalla macchina. All'atto di interruzione del monitoraggio, la colorazione dei widget tornerà ad essere quella iniziale. Occorre inoltre precisare che, mentre è impossibile avere un componente nello stesso momento sia in uno stato incongruente che in uno stato indeterminato, è possibile che un componente si trovi in uno dei due suddetti stati, oltre a non essere aggiornato; in questi casi la precedenza spetta alla qualità del componente, per cui la colorazione gialla avrà sempre precedenza su quelle blu e verde.

## Funzione 5: Visualizzazione del buffer eventi

Il click dell'utente sull'icona relativa al buffer degli eventi comporta l'apertura, da parte del sistema, di una finestra popup contenente il buffer degli eventi, descritto come da specifiche della macchina nel capitolo 2 e da specifiche del documento dei requisiti nel capitolo 3, solamente nel caso in cui l'attività di monitoraggio sia in corso.



*Figura 4.40: Schermata buffer degli eventi*

## Funzione 6: Visualizzazione dettaglio del componente

Scorrendo con il mouse sopra un widget, il sistema apre un popup in cui viene mostrata una breve descrizione del componente corrispondente al widget stesso, oltre ad un'immagine ingrandita del widget.

## **4.8 Use Case diagram**

In questa sezione viene mostrato ed adeguatamente commentato lo Use Case diagram relativo alla soluzione proposta per il software in oggetto.

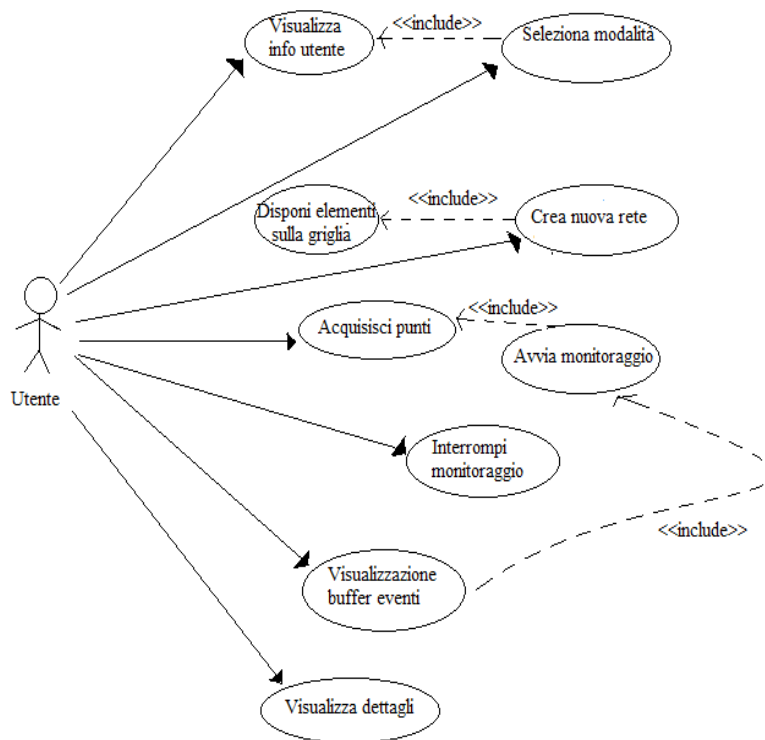


Figura 4.41: Use Case Diagram della soluzione proposta

Prima di iniziare la descrizione del diagramma, e solamente a titolo di precisazione, occorre notare come i casi d'uso proposti in fase preliminare siano stati scomposti in più attività che seppur tra loro collegate, possono essere effettuate in autonomia, a meno di alcuni vincoli preesistenti.

Dallo Use Case diagram proposto, e rappresentato in figura 4.40, si possono riconoscere i vincoli sopracitati, evidenziati con una relazione di inclusione dell'attività posta alla punta della freccia di relazione in quella posta alla base della freccia stessa; ad esempio:

- Per selezionare la modalità d'uso della macchina, e quindi la tipologia di monitoraggio da effettuare, è necessario passare prima dalla pagina principale, e quindi visualizzare necessariamente le proprie credenziali;
- La disposizione degli elementi sulla griglia è attività vincolante e necessaria per la creazione di una nuova rete;
- L'avvio del monitoraggio implica necessariamente l'acquisizione dei punti da monitorare;

- Per rendere possibile la consultazione del buffer degli eventi è necessario che l'attività di monitoraggio sia attiva.

Alcuni vincoli e relazioni triviali, come ad esempio il fatto che il monitoraggio si può interrompere solamente quando esso è attivo, sono stati volutamente omessi dal diagramma a causa della loro banalità.

#### ***4.9 Flowchart del processo del software***

Per un'ancora migliore comprensione della funzionalità del software, di seguito, in figura 4.41, è riportato un diagramma di flusso di ciò che accade dall'inizio alla fine dell'attività del software stesso; in questo diagramma vengono mostrati tutti i possibili percorsi effettuabili dall'inizio alla fine dell'attività del software, inoltre il blocco corrispondente all'attività di acquisizione di punti è rappresentato come un parallelogramma, in quanto riceve in input dei valori da esterno, e quindi si configura come un blocco di I/O



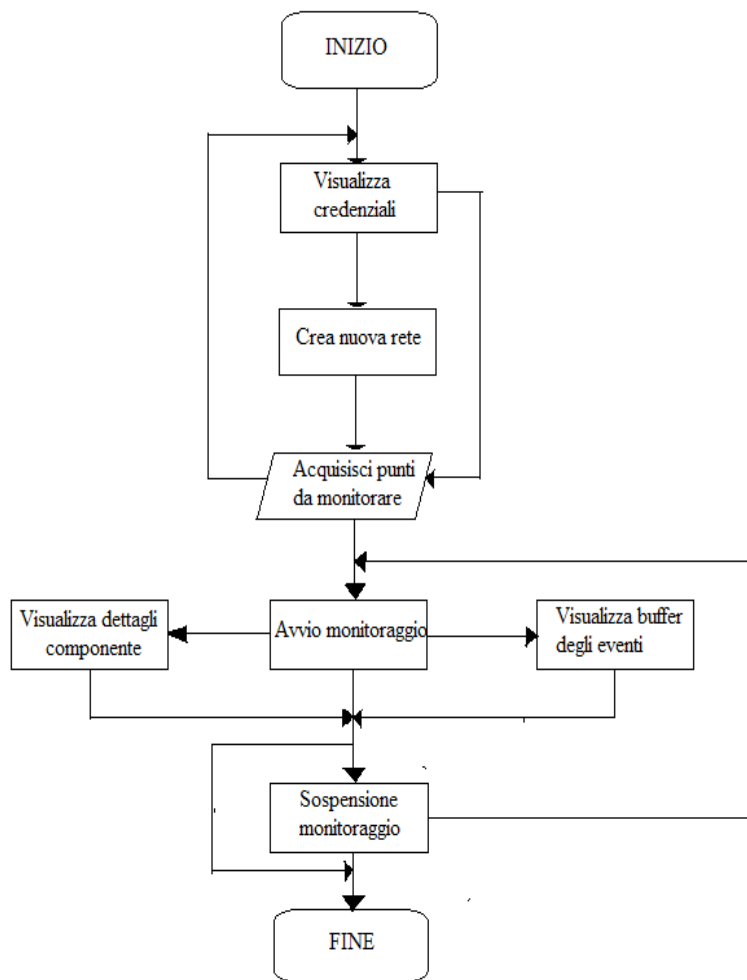


Figura 4.42: Diagramma di flusso delle funzionalità del software

## 4.10 Schema di raggiungibilità tra le varie pagine

Sempre allo scopo di rendere più chiara la visione d'insieme del software all'utente, di seguito è riportato lo schema comprendente il grafo di raggiungibilità tra le varie pagine.

E' necessario fare due brevi precisazioni riguardo al grafo:

- La pagina iniziale è la pagina utente;
- Le pagine relative a buffer degli eventi e dettagli del componente sono rappresentate con gli angoli smussati, in quanto rappresentano pagine “popup”, che quindi non si aprono a schermo intero, ma ne occupano solamente una porzione.

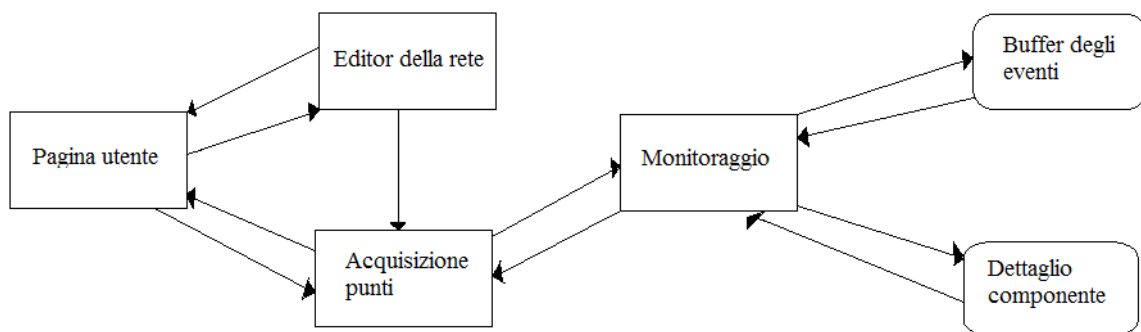


Figura 4.43: Grafo di raggiungibilità tra le varie pagine del sistema.

## Capitolo 5: Sperimentazione e testing

### 5.1 Introduzione

Questo capitolo presenta una prima sezione in cui si specifica la necessità di effettuare un'attività di testing <sup>[1]</sup> estensiva e di qualità, descrivendo come avviene generalmente il testing e quali parti del sistema dovrebbero esserne oggetto, passando poi ad una descrizione delle tipologie di testing “*black box*” e “*white box*” in seguito alla quale viene specificato per quale motivo ci siamo concentrati su un testing di tipo “*black box*”; in seguito descriveremo una simulazione di testing effettuata sull'attività svolta dalla macchina dopo l'installazione del nuovo software su di essa, specificandone dettagli, esiti, caratteristiche e modalità di risposta ai requisiti funzionali, indicando per ciascuno di essi il raggiungimento o meno dell'obiettivo prefissato e gli eventuali errori che sono stati incontrati, con le correzioni volte ad eliminarli.

### 5.2 Il software testing

Al giorno d'oggi, praticamente chiunque utilizza software, dal banale uso della posta elettronica e di Internet per i più svariati scopi, fino ad utilizzi di software prettamente tecnici, quali linguaggi di programmazione, tools di modellazione e di simulazione, sistemi di gestione del database, la tecnologia ci viene in aiuto per necessità che fino a qualche anno fa venivano eseguite manualmente. Quelli appena citati sono esempi in cui l'utilizzo del software è “ovvio” per gli utenti, ma in generale il software è molto più diffuso rispetto a quanto visto, addirittura si dice che esso sia comune almeno quanto la corrente elettrica agli inizi del secolo scorso: quasi tutti gli oggetti che abbiamo all'interno della nostra casa e al lavoro, ad esempio televisori, smartphone, frigoriferi, utensili da cucina, orologi da polso, sono frutto dell'integrazione di una grande quantità di software. Tutta questa diffusione del software ci fa necessariamente porre alcune domande sulla maniera in cui il software è sviluppato e distribuito; per chiunque sviluppi software è necessario mettere in commercio un software

che non abbia, per quanto possibile, “problemi” evidenti, in quanto risulterebbe poi complicato fornire una “patch” a tutti i clienti di questo software.

Inoltre, sfortunatamente, la predicibilità del difetto di un software aumenta all'aumentare del numero di utenti, e quindi è probabile che alc difetti rimangano “nell'ombra” per parecchio tempo. Ancora, è impossibile per chi mette in commercio un software, controllare l'utilizzo che tutti i clienti ne fanno, e quindi è possibile che a causa di un utilizzo non standardizzato del software da parte di un cliente, esso possa manifestare malfunzionamenti.

Infine, l'impatto di ogni singolo difetto richiede un'analisi, specialmente se si sta testando un'applicazione di tipo critico, perché solamente lo 0,1% di possibilità di errore è una percentuale troppo elevata per un software posto in commercio.

Generalizzando la situazione, esiste un paradigma del testing, da seguire quando si vuole effettuare attività di testing su un software:

- 1) L'obiettivo del testing è quello di trovare difetti prima che lo facciano i clienti;
- 2) Il testing esaustivo non è possibile: il testing può mostrare soltanto la presenza di difetti, non la loro assenza;
- 3) Il testing è un'attività valida per tutto il ciclo di vita del software, e non un'attività di fine ciclo;
- 4) Occorre capire pienamente le ragioni che stanno dietro ai test;
- 5) Occorre innanzitutto testare i test, per verificarne validità e attendibilità;
- 6) I test sviluppano immunità e devono essere rivisti periodicamente e costantemente;
- 7) I difetti si verificano prevalentemente nei cluster, e i test dovrebbero concentrarsi in tali cluster;
- 8) Il testing comprende la prevenzione dei difetti;
- 9) Il testing rappresenta un equilibrio delicato tra riconoscimento e prevenzione dei difetti;
- 10) L'automazione intelligente e ben pianificata è la chiave per ottenere benefici dal testing;
- 11) Il testing richiede talento, lavoro di gruppo e fiducia da parte dei committenti.

Nella figura 5.1 di seguito si può vedere come si articolano le varie fasi di sviluppo e come i difetti non riconosciuti nelle fasi a monte dello sviluppo del software vengono propagati nelle fasi a valle, causando un aumento dei costi non necessari e un potenziale downtime del sistema una volta che si arriva a concludere la fase di testing e a mettere in commercio il software con alcuni difetti non trovati.

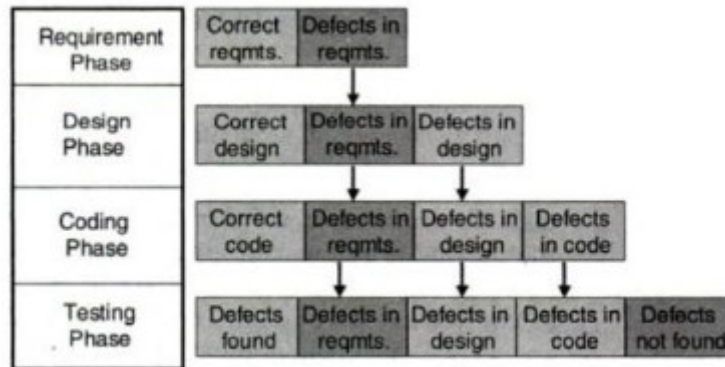


Figura 5.1: Propagazione dei difetti nelle varie fasi del ciclo di vita del software

La definizione stretta di “*testing*” indica la fase che segue il *coding* e segue il *deployment*, e rappresenta quella fase in cui si testa il codice del programma. Come detto, l'obiettivo del testing è quello di far emergere, minimizzare e prevenire i difetti del software, ma il solo testing del codice non è sufficiente a ciò, per cui occorre testare separatamente ogni singola fase dello sviluppo, in modo da poter riconoscere prima possibile eventuali difetti e ridurre quindi i costi.

Esistono svariate tipologie di testing, ma la scelta effettuata per testare questo software è stata principalmente tra due di queste tipologie: il *white box testing* e il *black box testing* :

- Il *white box testing* prevede la conoscenza completa del codice sorgente del software, e quindi chi esegue il testing deve trovare dati di test, ovvero assegnare determinati valori ad alcune variabili, in modo da percorrere tutto il programma, e poter identificare più cammini (sequenze di istruzioni

attraversate durante una singola esecuzione del programma) diversi che, combinati tra loro, possano dare una copertura delle istruzioni quanto più vicina possibile al 100%.

- Il black box testing, invece, non prevede necessariamente la conoscenza dell'intero codice, ma si basa principalmente sui requisiti forniti in fase preliminare, confrontando per ciascuno di essi quale dovrebbe essere l'output espresso dal requisito e l'effettivo output prodotto.

Avendo definito in precedenza, come si può vedere nel capitolo 3 di questo documento, quali sono tutti i requisiti, sia funzionali che non funzionali, che il software deve soddisfare, si è deciso di concentrarsi su una tipologia di testing di tipo black box.

### ***5.3 Demo e testing del software realizzato***

In questa sezione del documento si specificano le caratteristiche del device lato client utilizzato per effettuare la simulazione di utilizzo del software, per poi passare alla definizione della rete (o delle reti) utilizzata (utilizzate) per la simulazione e infine alla descrizione vera e propria della prova effettuata, nella quale vengono inoltre definite modalità e tempistiche delle corrispondenze ai requisiti.

#### **5.3.1 Caratteristiche delle risorse utilizzate**

Per fornire un'idea di massima agli utilizzatori del software, si specificano le caratteristiche della macchina utilizzata dal client per la simulazione effettuata, in modo che chiunque utilizzi il software possa avere un termine di paragone concreto e valido.

Device lato client: è stato utilizzato un normale personal computer con le seguenti caratteristiche:

- Sistema operativo: Window 7 Professional;
- Frequenza CPU: 2,53GHz;
- RAM:4,00 GB.

Tali caratteristiche sono ampiamente accettabili, in quanto rientrano ampiamente nei requisiti minimi del device lato client, descritti nella sezione 4.4

### **5.3.2 Definizione della modalità di testing effettuata**

Per poter trasferire il codice dell'interfaccia progettata dall'ambiente di sviluppo direttamente sulla macchina è stato utilizzato FileZilla, un comune FTP client: trasferendo tutti i file necessari verso l'host indicato dall'indirizzo IP della macchina è risultato semplice caricare sulla RTU l'interfaccia progettata.

La simulazione sulla macchina è iniziata acquisendo il valore di un semplice punto del campo, di tipo Single Point, per poi estendere man mano la simulazione, sia a livello di tipologia che di cardinalità dei punti.

### **5.3.3 Test dei requisiti e delle funzionalità**

Si è deciso di testare, oltre ai requisiti funzionali, anche quelle funzioni non facenti parte della cerchia dei requisiti, ma che comunque hanno una rilevanza notevole nel funzionamento del sistema stesso. Per ogni funzionalità testata è riportato un breve specchietto descrivente il requisito da testare e il risultato ottenuto.

- Test n.1: Avvio corretto del sistema sulla macchina STCE/RTU

Requisito: Dopo aver collegato il proprio device alla macchina, al click del pulsante relativo alla directory scelta, il sistema dovrebbe mostrare la pagina principale dell'interfaccia progettata.

Risultato: Il sistema mostra effettivamente la pagina principale del sistema, comprendente le credenziali di accesso dell'utente e la scelta della funzionalità.

- Test n.2: Funzionamento della modalità di editor di una nuova rete

Requisito: Scegliendo la funzionalità “*crea la tua rete*” dalla pagina principale, l'utente deve poter creare una propria rete. In particolare occorre verificare alcune particolarità della funzionalità in questione, tra cui:

-la possibilità di resettare la propria rete personalizzata, e quindi di rimuovere tutti i componenti posizionati sulla rete fino al momento;

-la possibilità di ruotare i widget che lo consentono prima di posizionarli sulla rete;

-la possibilità di non confermare il salvataggio, e quindi di poter modificare ulteriormente la rete dopo averla composta.

Risultato: il sistema permette di ruotare tutti i componenti per i quali è prevista questa possibilità, inoltre cliccando su “reset” la rete viene effettivamente resettata e riportata al suo stato originale, senza alcun widget posizionato ed infine, cliccando su “salva” si può effettivamente salvare la rete, dando la conferma con “ok”, oppure rendere disponibile la rete per ulteriori modifiche, cliccando “annulla”.

- Test n.3: Acquisizione dei punti

Requisito: Il sistema deve poter acquisire informazioni su tutti i punti oggetto di monitoraggio; in particolare è necessario conoscere per ogni punto il tipo, la qualità, il valore, l'eventuale natura (solo in caso di misura) e la descrizione, che sarà poi utile in seguito.

Risultato: Interfacendosi con la macchina è possibile ottenere in tempo reale lo stato di ogni singolo punto monitorato; nel caso specifico le caratteristiche



peculiari di ogni punto sono state salvate in un array, in modo che esse possano essere visualizzate a video nel corso del monitoraggio, o comunque gestite dall'utente al bisogno.

- Test n.4: Visualizzazione del buffer degli eventi

Requisito: cliccando sulla corrispondente icona, il sistema deve mostrare all'utente il buffer degli eventi, aggiornato in tempo reale, relativo al campo.

Risultato: l'utente, cliccando sull'icona, ha effettivamente accesso al buffer degli eventi e può consultarlo. Inoltre, si è provato ad accedere al buffer quando l'attività di monitoraggio era sospesa e, come ci si attendeva, non è stato possibile consultare il buffer degli eventi.

- Test n.5: Attività di monitoraggio

Requisito: Dopo l'acquisizione dei punti, viene eseguita la funzionalità del monitoraggio vero e proprio, nella quale, in base ad eventi di campo, i widget relativi ai componenti interessati da tali eventi dovrebbero cambiare colore a runtime.

Risultato: Effettivamente, dopo l'interfacciamento con la macchina l'utente riesce ad avere tutte le informazioni relative ad ogni singolo punto, per cui può vedere il cambiamento dei colori dei widget i cui componenti associati sono interessati da eventi.

- Test n.6: Visualizzazione dettagli

Requisito: Passando con il mouse sul widget, l'utente deve poter visualizzare i dettagli del componente associato, in particolare l'immagine ingrandita del componente e la sua descrizione.

Risultato: Dopo l'interfacciamento con la macchina e l'acquisizione delle caratteristiche dei punti, in particolare della descrizione, come ci si poteva aspettare, al passaggio del mouse sul widget vengono visualizzati l'immagine ingrandita e la descrizione del componente che esso rappresenta.

## Capitolo 6: Conclusioni e sviluppi futuri

Grazie all'aumento delle potenzialità hardware, che hanno permesso di rilassare i vincoli presenti finora, sia in fatto di costo della memoria che di potenza del microprocessore, è stato possibile realizzare un'HMI che potesse gestire autonomamente sulla macchina per la quale è stata progettata la parte di visualizzazione grafica dello stato di un set di punti tra loro correlati, senza necessariamente richiedere l'apporto di un centro remoto esterno.

Ovviamente l'architettura di tipo client-server è risultata la più adatta all'implementazione di questo tipo di interfaccia, e inoltre lo sviluppo dell'interfaccia stessa si è evoluto seguendo il fenomeno della convergenza digitale, il quale prevede:

- aumento della potenza e della flessibilità dei dispositivi utilizzati;
- numerose specifiche di standard tecnologici, quali ad esempio XML e HTML;
- successo di interfacce utente e architetture standard (web browser, grafica vettoriale, architetture a tre livelli)
- capacità di gestione delle informazioni in maniera coordinata ed integrata

Attualmente, questa interfaccia è stata progettata unicamente per attività di rilevamento, e quindi per supportare un flusso di dati unidirezionale che va dalla macchina fino all'utente; una sua possibile e logica estensione potrà sicuramente essere quella di implementare anche la possibilità di settare da remoto alcuni parametri relativi ai punti del campo, e quindi di non poter solamente rilevare, ma anche modificare certi valori.

In generale, l'ambito del telecontrollo è abbastanza chiuso in fatto di sviluppo, in quanto totalmente già esplorato, e quindi da questo punto di vista è molto difficile pensare ad ulteriori sviluppi significativi che seguano questo ambito.

Una possibile evoluzione di quanto sviluppato, potenzialmente realizzabile, benché parecchio “*visionaria*”, potrebbe essere quella di vedere in futuro la singola RTU come un punto terminale di una rete di Internet delle cose (IoT), e sfruttando le notevoli potenzialità del cloud, utilizzare quest'ultimo come tramite per la comunicazione delle varie macchine, fornendo così a tali macchine una potenzialità pressoché illimitata.

A livello personale, ho senza dubbio ritenuto molto accattivante il fatto di poter eseguire un lavoro di tesi su un argomento non astratto, e quindi di poterne vedere “concretamente” gli effetti, una volta installata l'HMI sulla macchina. Ciò ha notevolmente solleticato la mia curiosità è stato sicuramente motivo di grande stimolo per me.

Un fattore di stimolo ulteriore è stato il fatto di dover implementare una miglioria non ancora esistente nel concreto e quindi per cui non si avevano aspettative: il confrontarsi con tutte le variabili e i problemi che emergevano “*work in progress*”, e il dover pensare in tempo reale a modifiche e varianti alla soluzione iniziale, ha reso questo lavoro più imprevedibile, ma allo stesso tempo sicuramente anche più accattivante.

Infine, lo svolgimento di questa tesi ha segnato la mia prima esperienza nel mondo lavorativo, e nonostante il background e le conoscenze ottenute durante il percorso degli studi universitari, ciò ha rappresentato senza dubbio una grossa novità per me, sia a livello di tempistiche, che di conoscenze, che soprattutto di metodo di lavoro; sicuramente queste novità hanno portato al palesarsi di alcune difficoltà in questi mesi di lavoro, ma gli stimoli descritti in precedenza mi hanno fornito un grosso aiuto per superarle.

## Bibliografia e Sitografia

[1] Il telecontrollo nei servizi di pubblica utilità – Rocco Cutuli - Franco Angeli Editore, 2005.

[2] Sistemi di telecontrollo di reti di pubblica utilità. Reti idriche – Franco Angeli Editore, 2003.

[3] Trazione elettrica. Linee primarie e sottostazioni – Finzi Vittorio Editore, 2002.

[4] STCE/RTU, Monografia tecnica di apparato – Volume proprietario di SELTA SpA

[5] Introduzione a JavaScript [Online] <http://www.html.it/pag/45343/introduzione-a-javascript/>

[6] API della libreria JQuery, di JavaScript [Online] <http://api.jquery.com/>

[7] Application Server Wildfly [Online] <http://wildfly.org/>

[8] Introduzione a SVG [Online] <http://www.html.it/pag/31773/introduzione-a-svg/>

[9] Web server technology: The advanced guide for World Wide Web Information Providers – N.J. Yeager, R.E. McGrath, ed. Krol

[10] TCP/IP guida completa – Brian Komar

[11] Software testing, principle and praticses – S. Desikan, G. Ramesh, Pearson Education.