

SOMMARIO

Introduzione	1
1 Requisiti Software Generali.....	2
1.1 Tipologia di software e caratteristiche del fornitore	2
1.2 Modellazione orientata agli oggetti (Templates) e "Modello d'Impianto"	2
1.2.1 Creazione di Templates	2
1.2.2 Creazione di Istanze, ovvero Oggetti applicativi	3
1.2.3 Semantica dei Templates	3
1.2.4 Istanze e "Modello d'Impianto"	3
1.3 Logica Client-Server nativa	3
1.4 Allarmi	4
1.5 Formazione e Certificazioni per i System Integrators designati.....	4
1.6 Architettura flessibile: scalabilità ed estensibilità.....	4
1.7 Sistemi operativi supportati	5
1.8 Virtualizzazione e ridondanza, alta disponibilità e recupero del disastro.....	5
2 Requisiti software per l'Ambiente di Sviluppo	6
2.1 Ambiente di Sviluppo integrato (IDE).....	6
2.2 Ambiente di Sviluppo multi-utente	6
2.3 Organizzazione del progetto dall'Ambiente di Sviluppo	6
2.4 Sicurezza dell'utente	6
2.5 Tracciabilità dello sviluppo	6
2.6 Configurazione centralizzata delle impostazioni a livello di sistema	6
2.7 Archivio centralizzato di Templates e Oggetti applicativi	7
2.8 Integrità della configurazione (check in – check out)	7
2.9 Archivio di oggetti e relazione con l'ambiente di Runtime	7
2.10 Template di oggetti standard (base): estensibilità	7
2.11 Security (integrazione con Microsoft "Active directory")	7
2.11.1 Comunicazioni inter-nodo e inter-processo nel sistema di Telecontrollo	8
2.12 Protezione dei Templates.....	8

2.13	Utilizzo dei Templates da parte di altri Teams di sviluppo o di Teams di Messa In Servizio.....	9
2.13.1	Sviluppo	9
2.13.2	Messa In Servizio	9
2.14	Configurazione integrata dei dati storici.....	9
2.15	Configurazione integrata dei dati di allarme.....	9
2.16	Grafica integrata.....	9
2.17	Script logici per l'applicazione di Template di Oggetti	10
2.18	Assegnazione delle Referenze di I/O per ciascun attributo di Oggetto, verso il PLC	10
2.19	Modifiche ai Templates e propagazione ai Templates-figli e alle Istanze (Oggetti applicativi) .	11
2.20	Utility di importazione, esportazione, bulk import	11
3	Requisiti per il software di sviluppo dell'interfaccia HMI.....	12
3.1	Repository della configurazione HMI	12
3.2	HMI e sviluppo Object-Oriented	12
3.3	Architetture per HMI.....	12
3.4	Aspetti di internazionalizzazione.....	13
3.5	Tool di sviluppo della Grafica	13
3.6	Librerie di Stili Grafici	14
3.7	Librerie grafiche di Simboli e <i>Situational Awareness</i>	14
3.8	Gestione di periferiche video multi-monitor	14
3.9	Gestione di Schermi applicativi (Screens) come insiemi di Display	15
3.10	Gestione di Layout Applicativi (Application Layouts).....	15
3.11	Multi touch	15
3.12	Pan & Zoom	15
3.13	Ridimensionamento della Grafica e occultamento di particolari (cluttering / decluttering).....	15
3.14	Funzione di Alarm border.....	16
3.15	Navigazione: configurazione automatica basata sul Modello d'Impianto	16
3.16	Integrazione con Providers di Mappe e con Sistemi GIS.....	16
3.17	Controllo grafico di visualizzazione allarmi d'impianto.....	17
3.18	Funzione di Replay (visualizzazione a time lapse dei dati storici nella HMI)	17
3.19	Multi-istanza della HMI	18

3.20	Gestione di applicazioni HMI distribuite in rete.....	18
3.21	Notifica di modifiche dell'applicazione HMI al client.....	18
3.22	File di Logs (per applicazione HMI/Front End e per nodi Server).....	18
4	Ambiente <i>Runtime</i>	20
4.1	Gestione allarmi (acquisizione, classificazione, storicizzazione, Alarm Analytics).....	20
4.1.1	Classificazione degli Allarmi (Priorità e Severità: Standard EEMUA 191)	20
4.1.2	Alarm Analytics.....	21
4.1.3	Alarm Alerting	21
4.2	Architetture di comunicazione con Periferiche di Campo e sorgenti Dati (disponibilità di Drivers specifici).....	22
4.3	Disponibilità di Protocol Converters / Gateway.....	23
4.4	Internet of Things (IoT).....	23
4.5	Driver multi-istanza, comunicazioni Driver-DI TELECONTROLLO, comunicazioni in MQTT).....	23
4.6	Analizzatore (visualizzatore) di dati in runtime.....	24
4.7	Ridondanza nativa e Failover del runtime del Telecontrollo	24
4.8	Eventi di guasto: acquisizione e segnalazione.....	24
4.9	Tracciabilità in runtime (autenticazione utente in modalità Singola / Doppia).....	24
4.10	Registro delle azioni degli operatori.....	25
4.11	Reporting.....	25
4.12	Dashboarding, Analytics e Data Collection	26
5	Storico dei dati (Servizio "Historian"): requisiti.....	27
5.1	Grande velocità di archiviazione dei samples	27
5.2	Gestione efficiente dell'occupazione di spazio su disco	27
5.3	Gestione del dato archiviato per circolarità.....	28
5.4	Architettura nativamente fault tolerant della comunicazione "real time vs. Historian":.....	28
5.5	Interfacce standard per il consumo del dato storico	28
5.6	Interfacce custom per l'accesso al dato storico (SDK e Toolkits).....	28
5.7	Versionamento del dato storico.....	29
5.8	Disponibilità di Clients desktop-based o Web-based (HTML5).....	29
5.9	Caratteristiche specifiche dei Clients	29

5.10	Ridondanza	30
5.11	Cloud readiness	30
5.12	Architetture Tiered	30
5.13	Aggregazione spontanea del dato	31
5.14	Motore ad Eventi	31
5.15	Integrazione con la suite MS Office	31
5.16	Integrazione con MS "Reporting Services"	31
6	Servizi in supporto alla Soluzione	32
6.1	Garanzia, manutenzione e assistenza del software	32
6.1.1	Assistenza in garanzia	32
6.1.2	Assistenza estesa e manutenzione del software	32
6.2	Aggiornamenti software	32
6.3	Supporto delle patch per il sistema operativo	32
6.4	Assistenza telefonica	32
6.5	Assistenza via posta elettronica	33
6.6	Scaricamento di file	33
6.7	Assistenza via Web	33
6.8	Newsletter e CD dell'assistenza tecnica	33
6.9	Retrocompatibilità del software	33
7	Soluzione	34
7.1	Sviluppo Enti Impianto, sue sezioni e Asset di automazione	34
7.2	Templates di esempio	36

INTRODUZIONE

Questo documento intende fornire le specifiche per l'implementazione di una piattaforma operativa in tempo reale, al fine di garantire la gestione e il controllo efficaci dell'infrastruttura e di apparecchiature quali dispositivi/sensori, con un ambiente pienamente automatizzato per ottimizzare il monitoraggio, il controllo e l'esecuzione dei servizi, in modo intelligente e integrato.

Tale piattaforma consentirà di ottenere miglioramenti in 4 aree principali:

1. operatività
2. sicurezza
3. riduzione dei costi
4. esperienza degli utenti

1 REQUISITI SOFTWARE GENERALI

1.1 Tipologia di software e caratteristiche del fornitore

La piattaforma di sviluppo della soluzione di telecontrollo dovrà essere un prodotto software di mercato (non una soluzione basata su codice custom), distribuita da un fornitore di software che soddisfi i seguenti requisiti:

- oltre 100.000 implementazioni in tutto il mondo
- rete di System Integrator composta da oltre 3.000 aziende nel mondo
- R&D (roadmap di prodotto per i prossimi due anni)
- garanzia di aggiornamento delle versioni
- diversi livelli di assistenza e supporto tecnico, anche in lingua italiana

1.2 Modellazione orientata agli oggetti (Templates) e "Modello d'Impianto"

La piattaforma di sviluppo dovrà esprimere un concetto di modellazione degli Asset e della loro supervisione totalmente Orientato agli Oggetti (approccio *Object Oriented*).

Gli Asset da monitorare dovranno essere modellati da oggetti software con la funzione di Templates (Modelli), che incapsuleranno tutte le caratteristiche salienti quali:

- anagrafica degli attributi di I/O (comandi, stati, misure, setpoints ecc.)
- logica (codice, nel linguaggio di scripting della Piattaforma) per implementare funzioni da eseguirsi lato-Server quando in Esercizio
- grafica (uno o più simboli grafici per molteplici rappresentazioni dello stesso Asset in contesti diversi)
- configurazione della storicizzazione (quali attributi di I/O sono storicizzati e con che *policy*)
- configurazione della *security* (quali profili-operatore possono accedere quali attributi, ed in che modo)
- configurazione della comunicazione (indipendente dal protocollo di comunicazione di Campo che sarà effettivamente usato in Esercizio)

1.2.1 Creazione di Templates

A partire da librerie di Templates-base fornite dal Produttore del software, si potranno definire nuovi Templates-utente secondo le modalità tipiche dello sviluppo *Object-Oriented*, ovvero per Derivazione padre-figlio (definire Templates "figli" più specifici – con più attributi e caratteristiche -a partire da Templates "padri" più generici. I "figli" ereditano tutte le caratteristiche dei "padri" aggiungendo maggiori dettagli in più al loro livello), ma anche per Inclusione (si generano nuovi Templates includendo dentro a un Template "contenitore" uno o più altri Templates "contenuti", che non hanno nessuna parentela padre-figlio con il Container).

1.2.2 Creazione di Istanze, ovvero Oggetti applicativi

In seguito, a partire da un Template di asset, la piattaforma darà la possibilità di generare Istanze del Template (altrimenti dette Oggetti applicativi) in numero corrispondente agli Equipment in campo che sono tecnicamente descritti/rappresentati da quel Template (per esempio, dato un Template "Pompa", se un impianto di sollevamento fognario sotto supervisione possiede tre equipment di quella classe/tipo, si procederà a derivare da "Pompa" le istanze "Pompa_01", "Pompa_02", "Pompa_03" che corrisponderanno a processi da mettere in esercizio sulla parte Server del sistema di Telecontrollo).

1.2.3 Semantica dei Templates

I Templates potranno anche rappresentare funzioni logiche o astratte del sistema (per esempio, comunicazioni con processi esterni, come Databases di sistemi gestionali/amministrativi), dunque non solo per forza di cose degli Equipment o degli Asset fisici.

1.2.4 Istanze e "Modello d'Impianto"

Le Istanze create dai Templates, a livello "topologico", andranno a popolare una descrizione gerarchica, a "Modello d'Impianto", del patrimonio di Asset dell'utente finale.

La terminologia "Modello d'Impianto" viene usata in senso generale. Il termine "impianto" qui non è relazionato al concetto di *manufacturing*, ma viene utilizzato nel senso di un parco-assets generico, organizzato per gerarchia, ovvero:

- Sito di Esercizio
 - Aree ...
 - Sotto-Aree ...
 - Equipment complessi ...
 - Equipment elementari (parti costituenti) ...
 - Singoli segnali (I/O)

1.3 Logica Client-Server nativa

Il sistema di Telecontrollo deve supportare una architettura e una logica operativa Client-Server, che permetta di distribuire su più server l'intera applicazione.

L'insieme dei Server su cui è distribuita l'applicazione deve risultare come un unico cluster a cui puntano tutti i client.

I Client devono vedere un unico namespace di variabili, a prescindere dalla distribuzione dell'applicazione su qualsivoglia numero di Server. Deve essere possibile il trasferimento mirato di parti dell'applicazione da un server ad un altro (es. a scopo manutentivo o di load balance) senza dover riconfigurare nulla lato-Client.

I meccanismi con cui i Server si allineano deve essere nativo del prodotto e non deve richiedere sviluppo applicativo dedicato.

Al fine di mantenere l'integrità del progetto e ridurre al minimo i rischi di errore dovute a operazioni manuali, la distribuzione dell'applicazione dal repository unico di progetto verso i server e client deve essere nativo di prodotto e non deve richiedere allineamenti manuali (es. copie di file o cartelle).

1.4 Allarmi

Al fine di identificare gli allarmi in maniera efficace, deve essere possibile dare ad ogni allarme una priorità almeno fino a 999 livelli. Deve poi essere possibile raggruppare gli allarmi in 4 categorie di severità (cfr. standard EEMUA). Le severità devono poter essere associate a icone grafiche e colori che siano automaticamente richiamati sui sinottici, al fine di rendere più agevole agli operatori il rilevamento degli allarmi in base alla severità.

La storicizzazione degli allarmi deve permettere di sostenere fino a 1 milione di eventi al secondo. La storicizzazione degli allarmi, come per la parte di dati di campo, deve basarsi su meccanismi non-SQL al fine di efficientare lo storage.

1.5 Formazione e Certificazioni per i System Integrators designati

Il produttore della Piattaforma di Telecontrollo deve avere un "Programma di Certificazione" per valutare la qualità dei diversi System Integrator e ingegneri che implementeranno l'applicazione.

Al tempo stesso deve essere fornito un "Programma di Formazione", sia per i System Integrator sia per il personale dell'autostrada. Tale formazione riguarderà la tecnologia e non le modalità di utilizzo dell'applicazione sviluppata.

1.6 Architettura flessibile: scalabilità ed estensibilità

L'architettura software deve poter supportare potenzialmente un elevato numero I/O *real time* (dell'ordine del milione) e di centinaia di nodi-Server in una rete distribuita, anche geograficamente (WAN).

Per quanto riguarda i dati storici, deve essere possibile gestire fino a 500.000 historical tag su di un unico Historian Server. Se necessario, devono poter essere implementati (aggiunti) altri Server (possibilità di *scaling out* della soluzione, sia per la parte *real time* che storica, senza re-ingegnerizzare quanto già in Esercizio).

La piattaforma dovrà consentire - in generale - modifiche dell'architettura *hardware*. Si potranno includere, escludere o separare nuovi server senza che sia necessario cambiare o modificare le applicazioni installate su di essi.

Questa scalabilità verrà applicata non solo alle licenze ma anche all'esecuzione delle applicazioni. Quindi, progetti, espansioni o modifiche futuri devono poter essere realizzati senza apportare cambiamenti all'applicazione o alle applicazioni già sviluppate.

1.7 Sistemi operativi supportati

La Piattaforma di telecontrollo dovrà supportare la tecnologia Microsoft per ciò che riguarda l'utilizzo di Sistemi Operativi di Classe "Server" e "Client", come anche le piattaforme DBMS Microsoft "SQL Server", alle loro ultime versioni disponibili.

Devono potere essere supportati ambienti hardware con processori multicore (o ambienti multi-processor in generale), e supporti di archiviazione di massa di dati basati su tecnologie a Stato Solido (SSD).

1.8 Virtualizzazione e ridondanza, alta disponibilità e recupero del disastro

La parte Back End della Piattaforma di Telecontrollo (lato-Server) deve potere garantire la ridondanza come strumento nativo per rendere sia la comunicazione con le periferiche, sia l'esecuzione di logica lato-Server robuste e tolleranti ai guasti (a meno di tempi tecnici di failover tra elementi ridondati dell'architettura definitiva dell'applicazione).

La Piattaforma di Telecontrollo deve essere altresì compatibile con ambienti di virtualizzazione come Microsoft "Hyper-V" e "VMWARE VSphere", alle loro versioni più recenti, includendo anche compatibilità con loro caratteristiche avanzate quali funzioni di High Availability (HR), Fault tolerance (FT) e Disaster Recovery (DR).

2 REQUISITI SOFTWARE PER L'AMBIENTE DI SVILUPPO

2.1 Ambiente di Sviluppo integrato (IDE)

L'Ambiente di Sviluppo è costituito da un'unica applicazione integrata (Integrated Developing Environment) in grado di gestire tutti gli aspetti di sviluppo e test dell'applicazione di Telecontrollo.

2.2 Ambiente di Sviluppo multi-utente

L'Ambiente di Sviluppo deve offrire funzionalità multi-utente simultanee, con permessi di sicurezza e autorizzazioni in base al ruolo di ciascun utente.

2.3 Organizzazione del progetto dall'Ambiente di Sviluppo

L'Ambiente di Sviluppo deve offrire la possibilità di:

- organizzare i Template di Oggetti in Librerie-utente
- organizzare i simboli grafici utilizzati nei Templates in Librerie-utente
- visualizzare e configurare l'applicazione dal punto di vista del Modello d'Impianto (distribuzione delle Istanze, ovvero gli Oggetti applicativi distribuiti per Aree di appartenenza)
- visualizzare e configurare gli Oggetti mostrando la genealogia all'indietro dell'Oggetto stesso, fino al suo Template-genitore e ritornando in avanti all'Oggetto-base, indipendentemente dalla lunghezza della relazione genitore-figlio
- visualizzare e configurare l'applicazione dal punto di vista delle comunicazioni con il Campo

2.4 Sicurezza dell'utente

L'Ambiente di Sviluppo deve essere in grado di utilizzare la sicurezza del sistema operativo Microsoft, ad esempio i servizi di dominio Active Directory per autorizzare gli utenti a visualizzare, configurare e/o modificare i template e gli Oggetti applicativi.

2.5 Tracciabilità dello sviluppo

L'Ambiente di Sviluppo deve registrare un "tracciato" delle operazioni di check-out, check-in e revisione effettuate per ciascun template e oggetto applicativo, riportando l'ID dell'utente che ha effettuato le modifiche, un timbro con data e ora e un riepilogo dettagliato delle modifiche apportate.

2.6 Configurazione centralizzata delle impostazioni a livello di sistema

L'Ambiente di Sviluppo deve consentire la gestione e la configurazione centralizzata delle impostazioni a tutti i livelli del sistema, con la possibilità di:

- configurare la rappresentazione grafica standard (icone standard valide per tutta l'applicazione) della qualità della comunicazione e dello stato di lettura (valido/invalido) di ogni singolo attributo di campo.
- configurare gli stili standard per la formattazione numerica in tutta l'applicazione

- organizzare gli allarmi su quattro livelli di gravità: critico, alto, medio e basso, per agevolare l'interpretazione da parte degli operatori.
- assegnare priorità numeriche agli allarmi in base alla gravità
- selezionare le icone-standard per rappresentare i vari livelli di gravità di allarme previsti
- aggregazione di allarmi: capacità di creare Risolutivi di allarme, ovvero di contare e totalizzare tutti gli allarmi di ciascuna Area del Modello, e fornire informazioni per esempio su quale è l'allarme più grave o più prioritario.

2.7 Archivio centralizzato di Templates e Oggetti applicativi

L'Ambiente di Sviluppo deve utilizzare un archivio centrale (repository) di template e oggetti applicativi, gerarchia degli oggetti, configurazione dell'implementazione e genealogia. Deve inoltre offrire la possibilità di utilizzare lo stesso archivio per conservare e gestire template grafici e l'applicazione di visualizzazione.

Il repository deve essere implementato come Database relazionale su tecnologia Microsoft SQL Server.

2.8 Integrità della configurazione (check in – check out)

L'Ambiente di Sviluppo deve consentire a tutti gli utenti che dispongono dei diritti di configurazione di visualizzare gli oggetti e deve garantire che una sola persona alla volta possa mettere in check-out uno specifico template o un oggetto applicativo a scopo di modifica.

2.9 Archivio di oggetti e relazione con l'ambiente di Runtime

L'archivio (repository) deve essere utilizzato a solo scopo di configurazione, pertanto deve poter essere scollegato da un sistema in esecuzione senza compromettere il funzionamento runtime del sistema stesso. Deve essere possibile utilizzare il repository nella fase di runtime, ma non deve essere necessario.

2.10 Template di oggetti standard (base): estendibilità

L'Ambiente di Sviluppo deve rendere disponibili Librerie di Template-Base, cioè i template forniti dal produttore di software, sia rendere possibile la definizione di Template-Utente con le modalità di sviluppo Object-Oriented, per come descritte al Paragrafo della "Modellazione orientata agli oggetti".

Deve comunque essere disponibile un kit di strumenti di programmazione (SDK o Toolkit) con il quale l'utente possa creare nuovi oggetti base utilizzando linguaggi Microsoft Visual Studio come Visual C++, Visual C#, Visual Basic .NET.

2.11 Security (integrazione con Microsoft "Active directory")

L'ambiente di sviluppo deve potere permettere di configurare Ruoli e Utenti associati ai Ruoli sia in modo nativo (indipendente) sia integrato con i Servizi di Microsoft "Active Directory" (ovvero importare nella security dell'applicazione i Gruppi e gli utenti Windows membri di tali Gruppi, da dei nodi con qualifica di "Domain Controller", reperibili in rete).

Deve essere possibile definire una serie di privilegi tipici della Piattaforma di Telecontrollo, privilegi sia operativi (di runtime) sia di sviluppo e di configurazione (circa l'utilizzo dell'IDE o di strumenti affini) da associare ai Ruoli precedentemente definiti.

L'utente membro di un Ruolo deve ereditare tutti i privilegi del Ruolo.

Per privilegi operativi si devono intendere ad esempio:

- possibilità di tacitare allarmi
- possibilità di cambiare setpoints e in che modo (liberamente, con firma/autenticazione singola, o in doppia autenticazione)
- possibilità di inviare comandi critici (liberamente, con firma/autenticazione singola, o in doppia autenticazione)
- possibilità di interagire con Oggetti applicative in certe Aree ma non con Oggetti in altre aree differenti
- Possibilità di silenziare allarmi, o di mettere alcuni attributi allarmati in "shelving" (silenziamento limitato nel tempo)

Per privilegi di configurazione/sviluppo si deve intendere generalmente la possibilità di usare l'IDE o strumenti affini a piena capacità oppure no, profilando gli Utenti a seconda del Ruolo, e inibendo per esempio per alcuni Ruoli la possibilità di fare operazioni critiche sul progetto, quali:

- mettere fuori-scansione alcuni oggetti applicativi
- ripristinare backup di progetti al di sopra del progetto attualmente in Esercizio

2.11.1 Comunicazioni inter-nodo e inter-processo nel sistema di Telecontrollo

Per le comunicazioni interne alla piattaforma di Telecontrollo (tra nodi di architettura distribuita o più in generale tra processi della Piattaforma a prescindere dall'architettura dell'applicazione) deve essere possibile l'utilizzo di protocolli di crittografia di famiglia TLS (Transport Layer Security).

La soluzione deve essere in grado di implementare comunicazioni su Protocollo TLS a versione 1.2, utilizzando canali di comunicazione sicuri sopra i quali operare con le primitive tipiche del protocollo quali: TLS handshake, Key Exchange e Validazione di Certificati.

2.12 Protezione dei Templates

Per favorire gli standard di progettazione, distribuzione e protezione, il sistema di sviluppo deve consentire l'esportazione dei template di oggetti in modalità protetta, in modo che possano essere utilizzati da altri senza la possibilità di modificarli. Lo stesso vale per i template grafici.

2.13 Utilizzo dei Templates da parte di altri Teams di sviluppo o di Teams di Messa In Servizio

2.13.1 Sviluppo

Altri sviluppatori possono ricevere Templates, protetti oppure no, esportati dal progetto originario e trapiantati in un nuovo progetto. Nel nuovo ambiente, non sarà possibile modificare il Template se questo è protetto, ma lo si potrà sempre sotto-classare (derivare Templates-figli) e implementare relazioni di Contenimento tra Templates-figli per modellare asset complessi. In pratica, il Template protetto nel nuovo progetto viene assimilato a un Template-base.

2.13.2 Messa In Servizio

In una situazione di Messa in Servizio, i tecnici potranno creare Istanze a piacimento (oggetti applicativi) del Template, anche se protetto: la protezione non deve essere di ostacolo alla creazione di Istanze. Se il Template è stato sviluppato con determinate opzioni di configurazione, i Tecnici potranno scegliere diverse versioni del medesimo Template da istanziare, per creare Oggetti applicativi con più attributi o meno attributi, strutturalmente tra loro diversi, ma sempre direttamente derivati da quell'unico Template.

Semplificando, si vuole che i Tecnici di messa in servizio creino istanze anche strutturalmente diverse tra loro, maneggiando uno o pochi Templates, invece di utilizzare tanti Templates differenti ogni volta che debbano istanziare Oggetti applicativi anche leggermente / di poco diversi tra loro.

Questo meccanismo deve rendere la Messa In Servizio più semplice per i Tecnici, perché fa in modo che debbano maneggiare meno Templates, e contemporaneamente deve limitare l'eccessiva proliferazione del numero di Templates nelle Librerie, rendendole più comprensibili e leggibili a una loro navigazione (browsing).

2.14 Configurazione integrata dei dati storici

I Template di Oggetti devono consentire di configurare la memorizzazione dei dati storici degli attributi all'interno dell'oggetto con una semplice casella di spunta (checkbox). Non deve essere richiesto uno strumento apposito.

2.15 Configurazione integrata dei dati di allarme

I Template di Oggetti devono consentire di configurare il collegamento a un sottosistema di allarme che supporti allarmi condizionali (LoLo, Lo, Hi, HiHi, Rate-of-Change Deviation, ecc.), allarmi basati su eventi (True/False, Fail to Open, Fail to Close, Command Disagree, ecc.) con strumenti predefiniti che guidino lo sviluppatore lungo il processo di definizione della configurazione.

2.16 Grafica integrata

I Template di Oggetti devono consentire l'incapsulamento di una o più rappresentazioni grafiche dell'apparato modellato, a scopo di visualizzazione in diversi contesti di HMI (esempio: un simbolo per la

rappresentazione a "faceplate": pannello di controllo. Un differente simbolo per la rappresentazione dentro un Ainottico di Area, eccetera).

2.17 Script logici per l'applicazione di Template di Oggetti

I Template di Oggetti devono consentire l'associazione e la configurazione di uno o più script logici (logica lato-Server che l'Oggetto istanziato a partire dal Template eseguirà, se certe condizioni in esercizio – trigger – si verificano).

Gli script contenuti in un Template di Oggetti devono implementare ed essere aperti alla tecnologia Microsoft .NET (dot NET) e la compilazione su .NET Common Language Runtime. Il linguaggio di scripting della Piattaforma deve essere di facile comprensione, possibilmente somigliante a (o essere anche) Microsoft C++, C#, Visual Basic .NET.

L'utente deve essere in grado di modificare gli script logici mentre il sistema sta monitorando il processo, e decidere poi se e quando propagare la modifica (nuovo script o script modificato) al sistema in runtime.

Il linguaggio di scripting deve essere di tipo Imperativo e supportare costrutti e operatori quali IF, THEN, ELSE, ELSE IF, FOR, AND, OR, NOT, ADD, SUBTRACT, MULTIPLY, DIVIDE, EQUAL TO, NOT EQUAL TO, GREATER THAN e LESS THAN.

L'editor di codice deve offrire: funzionalità di completamento automatico (Intellisense-like); sintassi colorata per distinguere a colpo d'occhio parole-chiave, variabili, operatori; indice di riga/linea; controllo statico della sintassi nel codice (consapevolezza di possibili errori statici, senza dovere mettere in runtime il codice). Devono potere essere disponibili delle capacità almeno basiche di debug, come un log di sistema o un tool di osservazione delle variabili nell'applicazione di Telecontrollo (attributi di Oggetti applicativi) impattate dallo script.

Deve essere disponibile una libreria completa di funzioni matematiche e script più complessi.

Deve essere possibile includere nel codice dello Script chiamate a metodi esposti da Class Libraries .NET esterne (previa importazione e registrazione delle .DLL che li espongono, nel sistema di sviluppo del Telecontrollo).

La guida allo scripting (contestuale allo strumento di sviluppo – IDE – o esterna, su manualistica) deve comprendere esempi reali di codice funzionante che possano essere copiati e incollati nell'editor, per soddisfare tipiche esigenze applicative come: manipolazione di attributi dell'oggetto locale o di oggetti remoti, interazione con files o con databases esterni, ecc.).

2.18 Assegnazione delle Referenze di I/O per ciascun attributo di Oggetto, verso il PLC

L'Ambiente di Sviluppo deve offrire la possibilità di assegnare automaticamente referenze di Campo agli attributi degli oggetti applicativi (regole di autobinding che sfruttano regole basate su mappature e naming convention predefinite per i Topici e gli Items dei PLC disponibili).

2.19 Modifiche ai Templates e propagazione ai Templates-figli e alle Istanze (Oggetti applicativi)

La propagazione delle modifiche di un Template ai Templates-figli e agli Oggetti applicativi da loro derivati per Istanziamento, deve essere governabile:

se la propagazione debba avvenire e quando, è lasciato alla decisione dello sviluppatore.

2.20 Utility di importazione, esportazione, bulk import

L'intero progetto dell'applicazione di Telecontrollo (configurazione completa) o – selettivamente – gruppi di Templates, di Oggetti applicativi, di Simboli grafici devono potere essere esportati e reimportati su altri ambienti di sviluppo, a mezzo di trasferimento di files (anche di formato proprietario), per replicare un progetto esistente o dividerne elementi con altri Team di sviluppo.

L'Ambiente di Sviluppo deve anche supportare l'importazione e l'esportazione degli Oggetti applicativi in un formato di file leggibile e standard, ad esempio .CSV (separato da virgole), per la praticabilità di modifiche a mezzo di editor di testo o di un foglio elettronico come Microsoft Excel.

Deve essere possibile istanziare massivamente Oggetti applicativi in grande numero (funzione di bulk import), a mezzo di importazione di files .CSV che li descrivono.

3 REQUISITI PER IL SOFTWARE DI SVILUPPO DELL'INTERFACCIA HMI

Questo paragrafo illustra i requisiti per il software di sviluppo e di esercizio della HMI dell'applicazione di Telecontrollo.

3.1 Repository della configurazione HMI

Tutto il lavoro di sviluppo e configurazione della HMI (pagine grafiche) deve essere contenuto nella stessa Repository centrale (Archivio di Oggetti) dei Templates di progetto, ovvero:

3.2 HMI e sviluppo Object-Oriented

L'applicazione di Telecontrollo stessa, come HMI/Front end, deve essere gestita secondo un ciclo di vita e manutenzione simile a quello dei Templates di Oggetti applicativi. Dunque, anche per l'HMI dovrà esistere un Template, e tanti Oggetti applicativi (sue istanze) corrispondenti alle postazioni-operatore che ne fruiranno.

3.3 Architetture per HMI

L'HMI dovrà poter essere utilizzata come:

- Thick/Fat Client (istanza di applicazione HMI, fisicamente installata sulla workstation o PC o Panel PC dell'Operatore);
- Thin Client (una istanza di applicazione HMI risiede fisicamente su un nodo Server – Terminal Server - che permette l'accesso remoto concorrente a N Thin Clients via Microsoft Remote Desktop Services/RDP o via Web/http).

Nel caso di installazione dell'HMI su di una architettura a un Terminal Server HMI + N Thin clients, si richiede che:

- Esista la possibilità di avere un Server HMI ridondato e/o di load balancing
- Si possa far risiedere il Server di HMI in una DMZ, o comunque in una LAN diversamente segmentata rispetto alla rete di produzione (dove lavora la parte Server-side del sistema di Telecontrollo), che rimane protetta da Firewall o da Gateway che gestiscono il traffico HMI su particolari porte sorvegliate e monitorate.

L'installazione di una applicazione HMI, sia su un Terminal Server, sia su una stazione-operatore indipendente, deve avvenire via rete con comandi e manovre lanciati remotamente dall'IDE della Piattaforma di Telecontrollo.

Non deve essere obbligatoria la presenza di operatori in sito, per completare l'installazione di una istanza di HMI.

3.4 Aspetti di internazionalizzazione

Utilizzando il software di sviluppo del sistema di Telecontrollo, deve essere possibile costruire un'applicazione che, durante l'esecuzione (runtime), possa essere commutata dinamicamente da una lingua a un'altra.

Il sistema deve supportare qualsiasi impostazione linguistica attualmente disponibile nel sistema operativo.

Devono poter essere configurate diverse lingue (più di due).

Le stringhe utilizzate negli oggetti testuali con grafica devono poter essere configurate e visualizzate in diverse lingue, con possibilità di impostazione da parte del sistema all'accesso dell'operatore in base alle preferenze locali dell'operatore stesso.

I messaggi di allarme devono poter essere configurati e visualizzati nella lingua selezionata in ciascun momento.

Un testo privo di stringa tradotta nella lingua corrente deve essere visualizzato nella lingua di default.

3.5 Tool di sviluppo della Grafica

Lo strumento di sviluppo e configurazione della grafica deve essere integrato nell'ambiente di sviluppo (IDE) principale dell'applicazione di Telecontrollo. Le possibilità di modellazione grafica devono includere: primitive di disegno grafico vettoriale (linee, poligoni), linee curve/splines, possibile importazione di immagini esterne (per esempio files .JPEG o .PNG o altri formati comuni per file grafici).

Deve essere possibile applicare animazioni alle grafiche in sviluppo, animazioni collegate a referenze di Campo (items da PLC) o a condizioni calcolate, in base al quale la rappresentazione dell'oggetto cambia (per esempio: Ente visibile/non visibile, Ente cliccabile/non cliccabile, cambio di colore su condizione, o cambio di forma, blinking su allarme, comparsa di Alarm Border su evento di allarme ecc.)

Deve essere possibile includere, nei simboli grafici, degli oggetti software quali Controlli Client .NET (previa importazione e registrazione della .DLL che li espone, nel sistema di sviluppo del sistema di Telecontrollo), o in alternativa Controlli a tecnologia WPF (Microsoft Windows Presentation Foundation).

Lo strumento deve poter permettere l'utilizzo di grafica sviluppata con software esterni e di mercato (per esempio sistemi CAD) importando dati e metadati da questi sistemi in forma di files di formato .dwg, .xaml o .svg, da questi generati. Deve essere possibile convertire questi formati nella grafica nativa della Piattaforma e quindi, tramite l'editor grafico poter applicare le animazioni o le personalizzazioni necessarie.

Lo strumento deve poter supportare lo sviluppatore nel suo lavoro con features quali: comando di "undo" modifiche, utilizzo di una griglia per facilitare allineamenti reciproci di elementi della grafica, comandi per implementare all'istante allineamenti verticali, orizzontali, o equi-spaziare array di elementi sul piano verticale o orizzontale, possibilità di copy & paste / cut & paste di elementi grafici.

Lo strumento deve permettere di applicare al simbolo grafico in sviluppo una serie animazioni predefinite che lo sviluppatore dovrà soltanto configurare (es. Visibilità, lampeggio, riempimento orizzontale\verticale, linea multipunto, trascinamento, posizione X\Y ecc.), senza dovere scrivere codice.

3.6 Librerie di Stili Grafici

Al fine di ottenere un'interfaccia grafica coerente a livello di tutta l'intera applicazione (per tutte le pagine grafiche dell'HMI), deve essere possibile definire degli Stili Grafici salvabili e richiamabili all'occorrenza (Stili Grafici intesi come insiemi di valori predefiniti per proprietà quali: tipo/colore/peso di linee, tipo di colori di riempimento, font sizes, font colors eccetera). Dovrà essere possibile salvare Stili Grafici in Librerie-utente, e al momento del caricamento di uno Stile Grafico prescelto, tutte le caratteristiche di quello Stile devono essere applicate immediatamente "a caldo" a tutta l'applicazione (ovvero, non ci deve essere bisogno di mettere fuori linea una stazione-operatore, per applicare un nuovo Stile Grafico alla sua HMI).

3.7 Librerie grafiche di Simboli e *Situational Awareness*

Devono essere disponibili nel prodotto delle Librerie grafiche di base, con enti grafici che rappresentano sia oggetti elementari di navigazione come: pulsanti, bottoni, tabulazioni, ecc. sia rappresentazioni grafiche di asset produttivi comuni (basici): equipment elettrici, equipment meccanici/rotazionali, pompe, valvole, serbatoi, conveyors ecc.

A partire dagli Enti grafici di base, potranno essere definiti simboli via via più complessi e ricchi di proprietà e animazioni.

Deve essere disponibile - come Libreria di base – una Libreria di simboli orientata alla grafica "Situational Awareness", ovvero uno stile grafico di rappresentazione degli Asset produttivi svincolato da immagini fotorealistiche, quanto piuttosto legato a rappresentazioni P&ID o in stile ISA, di modo che l'operatore concentri l'attenzione sul processo produttivo in sé – o sull'esercizio in sé dell'Infrastruttura – piuttosto che sulla grafica fine a sé stessa.

Generalmente in una implementazione "Situational Awareness" di una HMI, l'attenzione dell'operatore deve essere attirata principalmente da anomalie del Processo Produttivo (scostamenti dalla norma dell'Esercizio). Se invece il processo si sviluppa entro parametri normali, nulla di particolare che richiami attenzione appare nella HMI.

3.8 Gestione di periferiche video multi-monitor

Lo strumento di sviluppo grafico deve permettere nativamente di sviluppare una applicazione grafica Multi-monitor.

3.9 Gestione di Schermi applicativi (Screens) come insiemi di Display

Lo strumento di sviluppo grafico deve permettere di definire uno Screen layout come insieme di singoli display (intesi come monitor, hardware), per ognuno dei quali si registra la risoluzione e il posizionamento reciproco l'uno rispetto all'altro.

Ad esempio, dovrà essere possibile definire uno Screen Layout per videowall, fatto da un quadrato 3x3 di 9 display a risoluzione 3840x2160 pixels ciascuno.

Oppure uno Screen Layout fatto da un array da scrivania di due monitor disomogenei, uno di rapporto dimensionale 16/9 ed uno 4/3, messo alla destra del primo.

Lo strumento di sviluppo grafico integrato con l'IDE della Piattaforma deve permettere di potere salvare nel progetto anagrafiche (librerie) di possibili Screen Layouts, per modellare i possibili hardware video composti ove l'applicazione di Telecontrollo (la sua HMI) potrà essere visualizzata.

3.10 Gestione di Layout Applicativi (Application Layouts)

Lo strumento di sviluppo grafico deve permettere di definire un Application Layout inteso come suddivisione in Frames dell'applicazione HMI (esempio: un frame superiore a sviluppo orizzontale per indicazione della data e ora, logo del cliente e pulsanti per commutazione lingua; un frame centrale di visualizzazione contenuti, e un frame laterale a sinistra del centrale, per compiti di navigazione tra le pagine con un controllo grafico ad Albero), suddivisione grafica che – *a priori* – è indipendente da una particolare risoluzione del sistema Video.

Lo strumento di sviluppo grafico integrato con l'IDE della Piattaforma di Telecontrollo deve permettere di potere salvare nel progetto anagrafiche (librerie) di possibili Application Layouts.

3.11 Multi touch

Dovrà essere possibile sviluppare una HMI per l'applicazione DI TELECONTROLLO, gestibile da periferiche video con possibilità di interazione tattile multitouch.

3.12 Pan & Zoom

Dovrà essere possibile sviluppare una HMI per l'applicazione di Telecontrollo, che permetta di effettuare panoramiche sulle pagine grafiche e zoom-in / zoom-out di particolari zone della pagina. Le Panoramiche e gli Zoom dovranno essere praticabili da gestures della mano (con una o due dita, con funzionalità multitouch della periferica video), oppure con trascinarsi del mouse o con bottoni o link che compaiano sulla pagina automaticamente (come proprietà-utente da abilitare o disabilitare) nel caso che la periferica video non gestisca eventi di multitouch.

3.13 Ridimensionamento della Grafica e occultamento di particolari (cluttering / decluttering)

Per limitare il carico computazione dei processori grafici, e migliorare la capacità di rendering delle pagine grafiche in velocità di apertura, dovrà essere possibile configurare un comportamento di ridimensionamento della grafica, per cui quando si osserva una pagina con un grande valore di "ZOOM –

“(piccola scala), i particolari di piccole dimensioni della grafica non vengono visualizzati (esclusi dalla visualizzazione).

Quando invece si utilizza un valore di Zoom molto grande (“ZOOM +”: grande scala) i particolari prima di dimensione troppo piccola per essere percepiti tornano nel ciclo di visualizzazione del motore grafico della piattaforma.

Se un meccanismo come questo non è presente, alle piccole scale (grandi valori di “ZOOM –“) i particolari di piccole dimensioni sono comunque visualizzati dal motore grafico, ma sono comunque troppo piccoli perché l’occhio dell’Operatore possa percepirli.

3.14 Funzione di Alarm border

La rappresentazione grafica di un Ente (asset) allarmato, dovrà essere circondabile con un rettangolo (entro cui il simbolo allarmato rimane circoscritto), poligono che manifesterà colore e comportamenti (per esempio blinking) in base all Severità e allo Stato dell’Allarme in quel momento più prioritario per l’Ente.

Lo sviluppatore non si dovrà fare carico di disegnare materialmente l’Alarm border per ogni ente grafico: la presenza dell’Alarm border sarà scelta in base alla configurazione di una proprietà nativa del simbolo grafico, richiamabile dallo strumento di sviluppo della grafica.

3.15 Navigazione: configurazione automatica basata sul Modello d’Impianto

Dovrà essere possibile, in base al Modello di Impianto configurato in precedenza (rappresentazione topologica/gerarchica dell’impianto, organizzata per Sito produttivo, Aree contenute nel Sito, eventuali Sotto-aree contenute nell’Area corrente) configurare automaticamente dei Controlli grafici di navigazione: controlli ad esempio ad albero, o controlli a breadcrumbs).

I controlli grafici di navigazione non andranno configurati o programmati, ma desumeranno automaticamente dal Modello d’Impianto la loro configurazione e/o profondità.

3.16 Integrazione con Providers di Mappe e con Sistemi GIS

Lo strumento di sviluppo del sistema di Telecontrollo dovrà mettere a disposizione controlli grafici o applets visuali da integrare nelle pagine grafiche, a seguito della quale integrazione la HMI sarà in grado di visualizzare Mappe multilayer da tipici Map Providers quali ad esempio:

- OpenStreetMap, Yahoo, Bing, ArcGIS, Google ecc.
- Anche Map providers conformi allo standard OGC potranno essere agganciati (Providers che implementano servizi WMS/WMTS).

I Templates e Oggetti applicativi disseminati nel Modello di Impianto dovranno essere in grado di determinare (popolare) un Asset Registry per il Sistema GIS di riferimento.

In generale, l’ambiente di sviluppo e configurazione del Sistema GIS dovrà essere integrato (o comunque integrabile) con quello della Piattaforma.

L'ambiente di sviluppo e configurazione del Sistema GIS deve favorire il riutilizzo di codice mediante template che definiscono i suoi oggetti applicativi e/o il suo Asset registry. I template possono essere personalizzati per creare nuovi template di oggetti applicativi, mantenendo le relazioni genitore-figlio.

L'ambiente di sviluppo e configurazione del Sistema GIS deve supportare l'importazione e l'esportazione della sua modellistica in formati standard di file leggibili, ad esempio .XML, o comunque formati che facilitino la manipolazione all'interno di un foglio elettronico come Microsoft Excel.

Deve essere possibile creare risorse e definizioni di dati per il sistema GIS, utilizzando il modello di impianto e le istanze di oggetti applicativi già contenuti nell'applicazione di Telecontrollo.

Deve essere possibile importare definizioni di Assets a mezzo di files. BIM ("Building Information Modeling") o di CAD files 2D o 3D, anche con metadati all'interno (per esempio files .DWG con layers).

3.17 Controllo grafico di visualizzazione allarmi d'impianto

Gli allarmi devono essere visualizzati configurando un oggetto grafico di riepilogo, che può essere posizionato separatamente o insieme ad altri oggetti in una finestra.

Gli allarmi devono avere un codice colore in base allo stato e alla priorità dell'allarme: allarme ricevuto/rilevato, allarme non ricevuto/rilevato e allarme rientrato ma non ricevuto/rilevato. L'utente deve poter scegliere fra colori differenti per la visualizzazione degli stati di allarme. L'oggetto "visualizzatore allarmi" deve supportare anche la visualizzazione di Eventi, laddove il colore utilizzato per gli Eventi deve essere anch'esso identificabile in fase di configurazione.

Gli allarmi devono poter essere visualizzati come allarmi in tempo reale o allarmi storici. Lo stesso oggetto deve comunicare con il processo in tempo reale (live) o con il database degli allarmi storici.

3.18 Funzione di Replay (visualizzazione a time lapse dei dati storici nella HMI)

Dovrà essere possibile scollegare una postazione-operatore dal monitoraggio real time degli asset produttivi, per collegare gli attributi animati non a referenze di campo, ma ai loro corrispondenti historical tags nella base di dati storica (l'Historian del sistema di Telecontrollo).

Qualora l'operatore scelga di entrare in modalità replay/time lapse con la sua postazione, dovrà poter specificare:

- un intervallo di dati storici da rappresentare, scelto tra due coordinate temporali (data/ora "DA", data/ora "A")
- una velocità di time lapse (1X, 2X, ecc.)

e, attivata la modalità time lapse, si dovranno potere vedere variare gli enti grafici in aspetto, colore, valori rappresentati in funzione dei dati raccolti nell'intervallo di tempo specificato, e alla velocità desiderata.

L'opzione time lapse lavorerà solo su quegli attributi (stati, comandi, misure, setpoints ecc) che sono anche storicizzati (attributi non storicizzati non esibiranno valori che cambiano durante il time lapse).

L'opzione time lapse dovrà essere caratteristica della singola postazione-operatore (attivare il time lapse su una postazione non dovrà mandare per forza in time lapse le altre stazioni, che continueranno il monitoraggio in real time).

3.19 Multi-istanza della HMI

Dovrà essere possibile attivare più di una istanza dell'applicazione HMI sul medesimo host (Operator workstation o Terminal server HMI), per esempio nel caso si vogliono differenziare le HMI per diversi profili-operatore.

3.20 Gestione di applicazioni HMI distribuite in rete

Il software di gestione del Telecontrollo deve fornire funzionalità standard che semplifichino la configurazione, l'utilizzo, la correzione e la manutenzione dell'applicazioni HMI da posizione di sviluppo centralizzata (remota rispetto ai client).

Il software di configurazione e sviluppo deve consentire di mantenere una singola applicazione-master in forma di Template sul repository centralizzata.

L'Ambiente di Sviluppo deve consentire la distribuzione automatica dell'applicazione-master a tutti i nodi-operatore sulla rete di telecontrollo, oltre alla propagazione delle modifiche apportate all'applicazione master a tutte le sue istanze nel sistema.

3.21 Notifica di modifiche dell'applicazione HMI al client

Quando viene rilevata una modifica all'applicazione HMI "template", ogni nodo-utente registrato come HMI client deve essere avvisato della modifica.

L'Ambiente di Progettazione deve consentire all'utente di definire la modalità di notifica della modifica dell'applicazione al nodo client: il nodo client può caricare automaticamente la nuova applicazione, oppure segnalare e fare scegliere all'utente di caricare o ignorare le modifiche, oppure ignorare automaticamente le modifiche.

In caso di caduta di rete fra l'archivio (repository centrale di Templates) e il client che si deve aggiornare, il client deve continuare a eseguire l'ultima applicazione distribuita. Quando la rete viene ripristinata, il sistema distribuirà l'applicazione server eventualmente modificata al client.

3.22 File di Logs (per applicazione HMI/Front End e per nodi Server)

I file di Log dell'applicazione, a scopo di diagnostica e risoluzione dei problemi, devono risiedere sugli hard disk locali per un numero di giorni definibile dall'utente. Ogni nodo di rete (sia esso postazione-operatore che nodo Server) deve mantenere un file di Log indipendente per le applicazioni specifiche di ciascun nodo (locali).

Deve essere disponibile un visualizzatore del Log, possibilmente implementato come snap-in delle Management Consoles/System Console di Microsoft, per la visualizzazione del Registro Eventi. Il visualizzatore deve supportare differenti colori per diversi tipi di Evento (Informazione, Warning, Errore)

e la granularità del Log, almeno per i moduli e processi costituenti l'applicazione di Telecontrollo, deve potere essere selezionabile.

Un visualizzatore di Log locale deve poter visualizzare anche il Log di un nodo remoto, se la topologia di rete consente ai nodi di comunicare.

Il Log (porzione del Registro Eventi, estesa da una data ad un'altra) deve potere essere esportabile su file.

4 AMBIENTE RUNTIME

Questo capitolo descrive le varie funzioni e caratteristiche che devono esibire sia il Front end / HMI del Sistema, sia la parte Back End (nodi Server del Sistema) in situazione di Esercizio.

4.1 Gestione allarmi (acquisizione, classificazione, storicizzazione, Alarm Analytics)

Gli allarmi devono essere rilevati e segnalati da un servizio di Gestione Allarmi. Il servizio Gestione Allarmi deve supportare almeno duecento (200) schermate client simultanee di allarme. Nell'eventualità di una tempesta di allarmi (centinaia o migliaia di allarmi al secondo), la Gestione Allarmi deve notificare e il client deve essere in grado di visualizzare fino a mille (1000) nuovi allarmi entro dieci (10) secondi dal rilevamento degli allarmi stessi.

Il sistema deve consentire lo "shelving" degli allarmi (cioè la sospensione temporanea) in modo che gli operatori autorizzati possano rimuovere temporaneamente gli allarmi selezionati dalla lista di allarmi attivi, sopprimendoli per un determinato intervallo di tempo. Il sistema deve chiedere agli operatori di specificare il motivo di tale soppressione.

Il sistema deve offrire la possibilità di sopprimere allarmi in base a determinati stati dell'impianto, per evitare di visualizzare allarmi inutili in presenza di specifici stati operativi.

Il sistema deve essere in grado di indicare il numero totale di allarmi per ogni categoria di gravità (critica, alta, media e bassa) in ogni Area del Modello di Impianto.

Il sistema deve essere in grado di gestire allarmi per le risorse di sistema (utilizzo della CPU, memoria ecc.).

Gli allarmi devono poter essere registrati in un database Microsoft SQL Server (o altrimenti, salvati in una base di dati non relazionale ma che sia aperta a consultazioni di clients che intendono consumare il dato via-linguaggio SQL). Gli eventi registrati relativamente agli allarmi devono comprendere istanza dell'allarme, ritorno alla condizione di normalità e riconoscimento/rilevamento dell'allarme. I parametri da registrare oltre all'evento dell'allarme devono comprendere data e ora dell'allarme, gruppo di allarme, nome del tag dell'allarme, tipologia di tag dell'allarme (reale/intero/booleano), tipo di allarme (LoLo, Lo, Hi, HiHi, ROC, Deviazione ecc.), nome dell'operatore, nodo operatore di riconoscimento dell'allarme, e priorità dell'allarme.

Gli allarmi devono poter essere stampati su una stampante locale o in rete. Gli allarmi stampati da un determinato nodo possono essere tutti gli allarmi, solo gli allarmi non riconosciuti/ricevuti, solo gli allarmi riconosciuti/ricevuti, gli allarmi di uno o più gruppi (aree del Modello D'Impianto), gli allarmi con determinate priorità o gli allarmi provenienti da diverse fonti.

4.1.1 Classificazione degli Allarmi (Priorità e Severità: Standard EEMUA 191)

La Piattaforma di Telecontrollo deve implementare nativamente un Sotto-sistema di Gestione Allarmi in grado di catalogare gli allarmi per Priorità, e suddividere lo spazio totale delle Priorità in quattro (4) Categorie Generali di Severità d'allarme (cfr. Standard EEMUA 191 \ ISA 18.2).

La configurazione delle Categorie Generali di Severità d'allarme (come anche stili grafici associati alla rappresentazione visuale della Severità nell'HMI: colore, altri stili grafici ecc.) deve essere accentrata per tutta l'applicazione.

La configurazione delle Priorità di allarme invece si può considerare prerogativa del singolo Template o del singolo Oggetto applicativo (per ogni attributo allarmato di ogni Oggetto applicativo, sarà possibile definire una Priorità d'allarme).

4.1.2 Alarm Analytics

Deve essere disponibile nella Piattaforma di Telecontrollo una soluzione web (portale web, sito web) di reportistica predefinita per Analytics sugli allarmi. La soluzione web deve esporre dashboard predefinite quali ad esempio:

- I Top-N eventi di allarme che hanno avuto rientri spontanei dopo X minuti
- I Top-N eventi di allarme tacitati entro i primi X minuti
- Distribuzione per Priorità (o Severità) degli Allarmi su un Asset, in un dato periodo di tempo (Da... a...)

La soluzione di Alarm Analytics dovrà mettere in grado gli Operatori – con un profilo adeguato – di sviluppare nuovi contenuti (nuove e differenti dashboard) e pubblicarle sul Portale Web dedicato.

4.1.3 Alarm Alerting

Deve essere disponibile un sottosistema integrato nell'architettura con lo scopo di centralizzare gli allarmi di un sito supervisionato e diffonderli in maniera affidabile ed efficace al personale dedicato, tenendo conto i media di diffusione scelti, secondo la pianificazione di servizio scelta.

Tutti gli allarmi gestiti devono essere registrati in uno storico dettagliato che indica per ciascun allarme informazioni quali la data di attivazione, il tempo di attivazione, il nome dell'operatore che lo ha tacitato e i tempi di reazione e d'intervento.

Lo storico degli allarmi e degli interventi deve permettere anche di disporre di statistiche complete per un singolo allarme, un gruppo di allarmi o tutti gli allarmi, per un determinato periodo (un giorno, una settimana o un mese) consultare le seguenti statistiche:

- Il numero di fermi sul periodo
- La durata totale dei fermi
- La durata media dei fermi

Possono essere anche presentate delle statistiche sugli interventi per operatore per un determinato periodo (un giorno, una settimana o un mese) per ottenere:

- Il numero d'interventi
- Il tempo medio d'intervento
- Il tempo medio di reazione

Deve essere disponibile un'APP su Android e iOS che permetta la notifica degli allarmi e di poter interagire con il sistema di alerting, inoltre deve essere integrabile un server vocale che permetta la diffusione e la consultazione degli allarmi tramite messaggi vocali.

Per ultimo deve essere possibile notificare gli allarmi tramite messaggi di testo.

4.2 Architetture di comunicazione con Periferiche di Campo e sorgenti Dati (disponibilità di Drivers specifici)

L'ambiente runtime deve essere basato su un'architettura di sistema distribuita. Deve essere possibile scalare l'architettura da un singolo nodo a 100 nodi e oltre. L'architettura deve contenere un modello multi-computer che viene visto con un unico namespace distribuito all'interno dell'ambiente runtime e non richiede la replica di dati da un nodo a un altro.

Su alcuni dei nodi menzionati posso risiedere istanze di Driver (I/O Servers) o Gateway/Protocol Converters che comunicano con tutti gli Oggetti applicativi utilizzati dalla Piattaforma di Telecontrollo, senza per forza dovere essere installati su ogni nodo dell'architettura distribuita.

La piattaforma deve fornire (direttamente, o a mezzo di 3rd party - Partner Tecnologici certificati come compatibili) i Driver di comunicazione per protocolli/interfacce standard di comunicazione industriale e IoT come:

- **Modbus** (Seriale e TCP)
- Protocolli di famiglia **IEC 104**
- Protocolli di famiglia **618150**
- **DNP3**
- **OPC DA e UA**
- **MQTT**

Web Services SOAP oppure con RESTful API (scambio di stringhe JSON), ecc.

Qualora un driver specifico non esista, dovrà essere fornito un kit di strumenti di programmazione (un SDK o un Toolkit) per poterlo sviluppare, in linguaggi comuni per l'ambiente di sviluppo Microsoft Visual Studio (C#, C++, Visual Basic .NET).

A prescindere dai Protocolli da supportare, data la possibile presenza di un Campo eterogeneo (PLC e RTU di differenti fornitori, presenza di SCADA legacy con tecnologie proprietarie ecc.) la Piattaforma deve potere nativamente interfacciare Periferiche di Campo di fornitori quali:

- **Siemens** (S7-300(F), S7-400(H), S7-1200, S7-1500)
- **OMRON** (CJ1 Series, CJ2 Series, CS1 Series, CP1 Series, CV Series CVM1 e CVM1D)
- **Allen Bradley** (ControlLogix, GuardLogix, CompactLogix, FlexLogix, SoftLogix, MicroLogix, PLC-5, SLC50)
- **Mitsubishi** (MELSEC-Q, MELSEC-QnA, MELSEC-L)

- **Hardware "CODESYS" in generale** (Schneider Electric 241/M251, Bosch Rexroth L40 e L65, CODESYS SP WIN V3.5.1, PLCWinNT V2.4.4.0, EATON XC-CPU202 CODESYS v3.5, WAGO CPU 750-841)
- **Hardware "SoMac"** (Modicon M241 PLC, Modicon M251 PLC, PacDrive 3 LMC Eco/Pro/Pro 2)
- **General Electric / hardware "SRTP"** (Series 90-30, 90-70 Versamax, Micro, Nano PACSystems RX3i, RX7i)
- **Texas instruments** (Control Technology Inc. CTI2500, TI545 / 565 / 575)
- **Beckhoff** (ADS Library TC1000:TC3 Beckhoff TwinCAT PLC/IO CX1000 e CX1020, PLC BC9000, BX9000 e CP66xx)
- **Automation Direct** (DirectLOGIC DL05 e DL06, DirectLOGIC DL205 Family, Controllers Serie 2000 & 3000, Controllers Do-More H2 Serie)
- **Opto22** (SNAP PAC Controllers)

Il supporto all'hardware menzionato potrà essere nativo della Piattaforma, o fornito a mezzo di 3rd party - Partner Tecnologici certificati come compatibili con la Piattaforma stessa.

4.3 Disponibilità di Protocol Converters / Gateway

Nell'ottica di implementare connettività a sistemi eterogenei (non solo sistemi o asset di Campo contraddistinti da presenza di PLC e RTU, ma eventualmente sistemi informativi Gestionali o Amministrativi, che si espongono come Databases pubblici e/o Web Services), la Piattaforma deve fornire al riguardo strumenti nativi o di suoi diretti partner tecnologici, come Protocol Converters / Gateway atti a trasferire le conversazioni (traffico dati) da un protocollo all'altro di quelli precedentemente menzionati.

4.4 Internet of Things (IoT)

La piattaforma deve essere aperta alla connettività della Internet Of Things (IoT), potendo gestire comunicazioni su protocolli tipici di quell'ambito come il MQTT o Web Services SOAP o REST.

4.5 Driver multi-istanza, comunicazioni Driver-DI TELECONTROLLO, comunicazioni in MQTT

I Driver specifici della Suite di comunicazione della Piattaforma devono consentire la connessione da parte di più Client contemporaneamente, e per ragioni di bilanciamento del carico e robustezza dell'applicazione, devono potere essere presenti in più istanze sul medesimo nodo Server, se occorre (Driver multi-istanza).

Per la comunicazione tra Driver specifico e livello real time del Telecontrollo, il Driver si deve esporre come Server OPC, oppure (nativamente o con l'utilizzo di un opportuno Protocol Converter / Gateway out-of-the-box) deve presentarsi come un MQTT Publisher (in grado di effettuare upload di dati ad un Servizio "MQTT Broker" raggiungibile in rete).

Deve anche essere possibile (previo utilizzo di driver specifico o di Protocol Converter / Gateway out-of-the-box) effettuare il Subscribe ad un Servizio "MQTT" Broker in rete, per consumare il dato ed alimentare

con questo la Piattaforma DI TELECONTROLLO (comunicazioni MQTT nelle due direzioni: "da Telecontrollo a Broker" e "da Broker a Telecontrollo").

4.6 Analizzatore (visualizzatore) di dati in runtime

Il sistema deve mettere a disposizione una utility nativa (extra-HMI) per visualizzare lo stato in tempo reale, la qualità e il valore di qualsiasi attributo di Oggetto Applicativo in esercizio.

Lo strumento deve poter permettere anche la scrittura del dato (per esempio, forzatura di un attributo real time di un qualche Oggetto applicativo nel sistema di Telecontrollo)

Questa applicazione di utility deve poter essere utilizzata per scopi di debug o diagnosi del sistema, e non è intesa essere essa stessa uno strumento di supervisione (non si richiede che visualizzi simboli grafici o sinottici tipici di una HMI).

4.7 Ridondanza nativa e Failover del runtime del Telecontrollo

Il software del sistema di Telecontrollo deve garantire l'alta disponibilità di tutte le funzioni in un normale ambiente di monitoraggio e telecontrollo. I componenti specifici per i quali è richiesta la ridondanza nel sistema sono: Oggetti applicativi (Istanze dei Templates) e Host dell'Oggetto Applicativo (Server fisico o virtuale), comunicazioni con PLC/RTU e sotto-sistema di Allarmi. I requisiti di alta disponibilità si applicano anche al registro dei dati di processo storici. La configurazione di Failover deve prevedere un oggetto di sistema Primario e uno di Backup che gestisca gli oggetti Primario e Backup incorporati. Questo sistema deve eseguire (scandire) gli Oggetti attivi e sincronizzare lo stato degli Oggetti attivi sul nodo primario con quelli in standby. Qualora venga rilevato qualsiasi guasto/difetto nell'esecuzione di un oggetto attivo o nella comunicazione con l'oggetto attivo, gli Oggetti in standby devono entrare in funzione e iniziare comunicare all'interno del sistema, con le periferiche di Campo.

4.8 Eventi di guasto: acquisizione e segnalazione

Devono essere rilevabili e riportabili all'operatore i seguenti eventi all'interno del sistema di Telecontrollo e dei suoi processi e componenti:

- interruzione delle comunicazioni verso i PLC/RTU
- interruzioni delle comunicazioni verso il server di comunicazione
- guasto della logica applicativa
- guasto della gestione Allarmi
- interruzione delle comunicazioni verso lo storico dei dati
- spazio residuo ridotto in qualsiasi archivio storico in rete

4.9 Tracciabilità in runtime (autenticazione utente in modalità Singola / Doppia)

Il sistema deve poter essere configurato in modo che qualsiasi modifica a una variabile in fase di runtime venga tracciata con ID utente, nome completo dell'utente, valore precedente e valore nuovo.

L'implementazione di questa raccolta dati deve fare in modo che l'Utente, quando vuole inviare un comando critico o cambiare un setpoint critico, debba reinserire le sue credenziali di login, per esempio, in una finestra modale di popup (funzione nativa della Piattaforma: autenticazione in firma singola).

Per la doppia firma, un secondo Utente con privilegi operativi superiori al primo, dovrà inserire a sua volta le sue credenziali.

I dati raccolti dalle autenticazioni devono essere registrati come campi di record di Eventi di security, che vanno storicizzati su una base dati permanente e aperta a successive consultazioni.

4.10 Registro delle azioni degli operatori

Tutte le operazioni svolte dall'operatore devono essere registrate in un registro degli eventi. Il registro eventi deve tenere traccia di ogni attività dell'operatore: quando accede ed esce dal sistema, quando modifica setpoint, quando assume il controllo di dispositivi.

Ogni registrazione deve contenere data, ora, operatore collegato e tipo di azione (modifica setpoint, cambio di stato ecc.).

4.11 Reporting

Il sistema di Telecontrollo deve consentire la realizzazione di report completi e approfonditi.

Nel dettaglio delle funzionalità deve integrare un editor di report personalizzato, che conferisca ai report un'interfaccia grafica accattivante e professionale per visualizzare e analizzare eventi transazionali e in real time.

La funzionalità di reporting deve permettere di crea report in modo veloce e condividerli con il personale interessato. La gestione e la configurazione dei report deve utilizzare un approccio low code, offrendo una semplice interfaccia di configurazione drag and drop.

Deve essere presente la possibilità di creare report complessi su diverse tematiche:

- batch
- OEE,
- log operatore
- MES
- report di produzione
- efficienza energetica
- inserimento manuale dei dati

Dovrà essere possibile definire il template del report e come esso verrà generato:

- su richiesta dell'utente
- al verificarsi di una particolare condizione (es. fine batch)
- su cadenza temporale definita

I report dovranno essere generati in diversi formati:

- PDF
- CSV
- Excel
- XML
- Web Page

Dopo la generazione dei report dovrà essere possibile condividere i documenti tramite diverse modalità:

- File Server
- E-mail
- Print
- FTP Server

4.12 Dashboarding, Analytics e Data Collection

La piattaforma deve centralizzare i dati provenienti da più fonti in un unico ambiente cloud, integrando funzionalità di data collection, dashboarding e analytics per condividere le informazioni rendendole disponibili da qualsiasi posizione.

Deve fornire l'accesso ai dati operativi da remoto e consentire agli operatori di collaborare in real-time, lavorando in modo più efficace ed efficiente, monitorando in tempo reale le prestazioni degli asset e riducendo i costi di manutenzione.

Deve fornire la possibilità, attraverso l'utilizzo delle dashboard, di combinare i dati operativi con i dati di business, per identificare le opportunità di miglioramento delle Operations.

La piattaforma deve inoltre integrate funzioni di data Analytics permettendo di rilevare tempestivamente le anomalie del processo produttivo e di monitorare lo stato di salute degli asset garantendo agli operatori maggior consapevolezza delle condizioni dell'impianto, aumentando così l'efficienza e riducendo i downtime.

Un valore aggiunto è dato dalla possibilità di effettuare il monitoraggio e il controllo consentendo l'acquisizione e la supervisione, in tempo reale, dei consumi energetici e dei principali parametri di funzionamento degli impianti massimizzando così l'efficacia delle azioni di risparmio.

È importante che i dati siano crittografati e pubblicati sul cloud in modo sicuro, in modo che la rete sia protetta dall'accesso esterno.

Deve supportare la visualizzazione dei dati per:

- Smart Phone come iPhone, Android
- Tablet e Phablet (es. Samsung Galaxy Note)
- PC Desktop e Laptop mediante qualsiasi browser HTML5, tra cui Chrome, Safari, Edge
- Grandi video display come uno schermo di proiezione o HDMI TV

5 STORICO DEI DATI (SERVIZIO "HISTORIAN"): REQUISITI

I dati storici generati dal processo di Esercizio (si intendono sia Trends di variabili storiche come Record di Allarmi d'impianto storicizzati) devono essere immagazzinati in una base di dati centralizzata. La base di dati centralizzata, oltre che per mera consultazione dello Storico di processo/esercizio, dovrà potenzialmente servire anche come Training Set per algoritmi avanzati di Machine Learning (Esempio: analisi di storia passata di guasti, con apprendimento per Pattern Recognition a scopo di manutenzione predittiva), o come dataset per validare o istruire eventuali Sistemi di Supporto alle Decisioni, oppure come sorgente dati affidabile per Reportistica/Analytics.

Per queste premesse, la base di dati storica centralizzata dovrà essere implementata con una soluzione avente:

caratteristiche di estrema robustezza e fault tolerance sia per l'acquisizione del dato alla fonte (da periferiche real time) sia per la scrittura e conservazione sul supporto di archiviazione, poiché eventuali interruzioni dei Trends di dati storici invaliderebbero i processi di apprendimento dei servizi basati su Algoritmi, o i risultati di reportistica avanzata/analytics.

Apertura alla consultazione dei dati da parte di processi esterni, per tramite di interfacce standard di accesso (consumo) del dato.

Nel dettaglio, si richiedono alla soluzione le seguenti caratteristiche specifiche:

5.1 Grande velocità di archiviazione dei samples

Si richiede una performance di archiviazione di targa dell'ordine di grandezza dei 100.000 (centomila) samples/sec.

Tale livello di frequenza di archiviazione deve essere raggiunto senza perdita di risoluzione del dato, ovvero senza compressioni o aggregazioni artificiali/obbligatorie del dato. Tutti i samples in uscita da un I/O Server o Driver di Campo devono potere essere acquisiti.

5.2 Gestione efficiente dell'occupazione di spazio su disco

Si richiede un'occupazione di spazio-disco di circa 2 ordini di grandezza inferiore ad una soluzione basata su Database commerciali puramente relazionali (in assenza di compressioni aggiuntive indotte da sistemi operativi o utility dedicate), a parità di numero, tipo di dato e policy di archiviazione delle stesse historical tags. A questo proposito, per raggiungere tale livello di compressione del dato (compressione trasparente rispetto all'utilizzatore) sono anche accettate tecnologie non relazionali di archiviazione (basi di dati NO-SQL) per l'implementazione del requisito.

Non sono invece accettate compressioni del dato in termini di eliminazione di samples, allo scopo di minimizzare uso di spazio o di inseguire alte velocità di archiviazione diversamente non praticabili (l'eliminazione forzata dei samples "primitivi" in luogo di valori aggregati comporterebbe perdita di risoluzione sul fenomeno osservato), ovvero: non è accettato archiviare solo degli aggregati (per esempio Medie pesate o Sommarie/Integrali) in luogo della sequenza completa dei campioni.

La possibilità di archiviare dati aggregati (o di rappresentarli aggregati nei Trends) non deve essere in alternativa alla archiviazione con policy "full", ma deve essere un'opzione che semmai affianchi l'archiviazione "full".

Devono essere forniti strumenti o Tools di semplice uso, per stimare l'occupazione di spazio-disco di una procedura di archiviazione, specificando in input i tipi di dato e i numeri complessivi delle historical tags da gestire, nonché la loro policy di archiviazione (full, delta, delta/w deadband ecc). Questo strumento verrà utilizzato dal Cliente finale e dal suo Integratore per predisporre una architettura di archiviazione (spazio disco) adeguata al set di historical tags di interesse, e permanente per tutto il tempo che il Cliente desidera (dati storici online potenzialmente per anni addietro).

5.3 Gestione del dato archiviato per circolarità

Quando il dato è più vecchio di una certa soglia temporale configurabile, oppure quando si è prossimi alla saturazione del Folder di archiviazione, il dato più vecchio viene spostato su Folder alternativi o – se richiesto – viene cancellato: funzione di "auto purge".

5.4 Architettura nativamente fault tolerant della comunicazione "real time vs. Historian":

Se i processi real time del sistema di Telecontrollo perdono temporaneamente la comunicazione con l'Historian Server (per problemi di rete intermittente, o perchè l'Historian Server stesso sperimenta un downtime), onde evitare di perdere informazione storica, i nodi dei driver di comunicazione o comunque i nodi dove risiedono Oggetti applicativi devono trattenere l'informazione storica non ancora archiviata in una cache locale, limitata solo dallo spazio fisicamente disponibile sul disco locale del nodo.

Quando l'historian Server recupera il disservizio (o la rete è di nuovo disponibile) la cache dei nodi periferici viene progressivamente copiata sull'Historian e localmente cancellata.

Questa funzionalità deve essere nativa dell'intera Piattaforma di Telecontrollo: non si devono richiedere al Vendor del software licenze speciali per esercirla, nè si deve integrare con codice custom o scripting.

5.5 Interfacce standard per il consumo del dato storico

Si richiede:

- Comprensione del linguaggio SQL a mezzo di interfaccia ODBC per ambienti Microsoft
- Disponibilità di una WEB API o di una API RESTful per il consumo del dato (esempio: Protocollo OData, traffico di stringhe JSON).

5.6 Interfacce custom per l'accesso al dato storico (SDK e Toolkits)

Deve essere disponibile un kit di strumenti di programmazione (SDK o Toolkit) con il quale l'utente possa creare nuove modalità custom di accesso e di consumo del dato storico (interfacce non standard), utilizzando linguaggi di programmazione comuni di di Microsoft Visual Studio come Visual C++, Visual C#, Visual Basic .NET.

5.7 Versionamento del dato storico

In caso di modifica a posteriori del dato storico acquisito, la soluzione deve mantenere traccia sia del dato originale che dell'ultimo dato entrato.

5.8 Disponibilità di Clients desktop-based o Web-based (HTML5)

Per analisi di trend storici multi-variabile o rappresentazioni del dato orientate al dashboarding (Paretos, Torte, etc.), sia in ambiente desktop che Web.

5.9 Caratteristiche specifiche dei Clients

- Deve essere possibile rappresentare Trends in modalità storica o "near - real time" (autoplay) di historical tags singole o multiple. Per i Trends multipli, deve essere possibile rappresentarli su una scala comune (sovrapposti) o ciascuno con un suo proprio plot e scala (stacked).
- Su specifici punti dei Trends devono potere essere scritti e salvati permanentemente dei commenti degli Operatori.
- Deve essere possibile eseguire l'analisi di un Trend di una variabile "x" non solo in funzione del Tempo, ma anche in funzione di un'altra variabile "y" (stile Diagrammi di isteresi o Scatter Plots).
- Deve essere possibile definire graficamente nei Trends o negli Scatter Plots x/y delle Regioni Critiche (poligoni) che rappresentano l'area "di comfort" del processo di esercizio.
Se un Trend o uno Scatter Plot x/y si allontana dalla regione, deve essere evidente immediatamente all'Operatore che analizza il dato storico, che in quel momento era accaduta un'Anomalia.
- Deve essere possibile salvare la configurazione del Trend che si sta osservando su file (configurazione intesa come numero delle variabili e come stili di rappresentazione) per poterla richiamare all'occorrenza, come anche salvare su file (per esempio in formato tabellare, su file .CSV) i samples che costituiscono l'andamento di un Trend osservato.
- Deve essere possibile applicare interpolazioni ai dati del Trend, come Lineare o Stair-step.
- Deve essere possibile configurare un Trend applicando al rendering della linea un algoritmo di retrieval del dato, che può essere diverso dall'algoritmo (policy) di archiviazione del dato dalla periferica di campo. Per esempio, se i samples di un Trend sono stati acquisiti con policy "full" dalla periferica (acquisito ogni singolo sample rinfrescato dal Driver a ogni tempo-ciclo di lettura del PLC), si può scegliere di reindirizzare il Trend con algoritmo "Delta", applicando o meno deadband sui valori, oppure con un algoritmo di famiglia "Best-Fit". Questo al fine di velocizzare il rendering del Trend (e l'evasione della conseguente Query sottostante al Sottosistema Historian del Telecontrollo). Come algoritmi di retrieval del dato, a parte il "Delta" e "Best-Fit" deve essere possibile anche:
 - Rappresentare l'andamento di un Trend tra due date (da... a...) in stile "N samples equi spaziate nell'intervallo di tempo comunicato", con N parametrabile.

- Rappresentare l'andamento di un Trend tra due date (da... a...) in stile "samples equi spaziate di T secondi o millisecondi", con T parametrabile e numero di samples rappresentati che aumenterà al diminuire di T.
- Deve essere disponibile sui Client un Tool o un'interfaccia che possa aiutare l'operatore a profilare il dato storico (introducendo per esempio in un Form parametri come: la singola historical tag o il set di historical tags da cui estrarre i samples, intervallo di data/ora "da... a..." per l'estrazione del dato, eventuali filtri tipo eliminazione di valori NULLS o di Outliers, ecc.) e, in conseguenza della profilazione, generare il corrispondente codice SQL corretto con cui per esempio un'applicazione esterna non nativa del sistema di Telecontrollo, possa eseguire successivamente una Query e recuperare quel dato.

5.10 Ridondanza

Possibilità di implementare una architettura ridondata, in termini della presenza di un Server "Historian" gemello, trasparente rispetto alle richieste dei Client che non distinguono tra i Server membri del Pool. L'allineamento continuo dei dati tra i due Server deve essere implementato come servizio automatico.

5.11 Cloud readiness

La soluzione "Historian" deve essere cloud-ready: il nodo "Historian" della soluzione potrà essere un Server Historian locale "on premises" o un Server "Software as a Service" basato su un cloud system di tecnologia Microsoft "Azure".

5.12 Architetture Tiered

Possibilità di implementazione di architetture TIER-2 (TIER-N in generale) tra Historian multipli.

Ad esempio, un Historian di un sito periferico (TIER-1) replica i dati su un Historian di Posto Centrale (TIER-2), il quale a sua volta potrebbe replicare altri suoi dati ad un livello ancora superiore (per esempio TIER-3: Historian di livello Corporate / Intercompany).

All'estremo superiore di una architettura TIER-N deve esservi la possibilità di definire un cloud-based Historian (Historian Server "SaaS" su Cloud Microsoft "Azure").

Al livello superiore di una architettura Tiered, i server Historian possono essere multipli (ovvero, un Historian di un sito remoto deve poter replicare il dato a due server distinti al livello TIER-2, magari geograficamente distanti, implementando di fatto una funzione di "Disaster Recovery" sui dati storici).

Oltre che per architetture orientate al Recupero del Disastro, il Tiering permette di creare Basi di Dati replicati disaccoppiate ed indipendenti dagli Historian "di produzione", in modo che applicazioni avanzate di Reportistica oppure applicazioni di training per Algoritmi (Machine Learning) agiscano sui livelli superiori del Tiering, lasciando gli Historian "di produzione" più liberi dal punto di vista del carico computazionale (meno evasioni di richieste di possibili Clients).

5.13 Aggregazione spontanea del dato

Deve essere possibile generare degli historical tags "calcolati" (non direttamente acquisiti dal Campo collegato al sistema di Telecontrollo) che rappresentino dati aggregati di historical tags primitive (non aggregate). Per esempio:

- deve essere possibile aggregare una historical tag analogica calcolandone la media, il minimo, il massimo, l'integrale, la standard deviation ecc.
- deve essere possibile aggregare una historical tag discreta (valori che variano in un numero finito di stati, potenzialmente fino a 10, non soltanto tags booleane) calcolandone il numero di volte che è transitata in uno stato (per ogni stato), o il tempo complessivo di permanenza in uno stato (per ogni stato)

Questa caratteristica deve essere nativa della piattaforma "Historian" proposta. Non deve essere richiesto di sviluppare codice o scripting per implementare questi comportamenti. Chi configura queste caratteristiche di funzionamento dell'Historian, dovrà solo specificare quali sono le tags da aggregare, e l'intervallo di tempo di aggregazione (ogni 30 minuti, ogni ora, ogni giorno ecc.), intervallo che potrà essere diverso per ogni singola historical tag aggregata.

5.14 Motore ad Eventi

Deve possedere un motore ad eventi: a fronte di condizioni configurabili da verificare sulle historical tags (triggers) il Server deve potere eseguire una serie di azioni di risposta come:

- invio di e-mail
- esecuzione di codice SQL in linea o di stored procedures su un DB esterno
- registrazione (archiviazione) di uno snapshot di dati: valori istantanei di un vettore di historical tags configurabili a piacere, nel caso per esempio che l'evento "trigger" sia stata la registrazione di un guasto, e che i valori dei dati di snapshot possano aiutare il personale di Manutenzione a interpretare correttamente il guasto stesso.

5.15 Integrazione con la suite MS Office

La soluzione "Historian" deve essere compatibile e integrabile con Microsoft Office (Excel): le celle di fogli di calcolo Excel devono potere essere collegate (referenziate) ad historical tags disponibili nella base di dati storica del sistema di Telecontrollo.

5.16 Integrazione con MS "Reporting Services"

La soluzione "Historian" deve essere compatibile con la tecnologia Microsoft SQL Server Reporting Services (MSSRS), in altre parole deve essere possibile identificare la base di dati storica come una possibile Data Source in una soluzione MSSRS, onde potervi sviluppare della reportistica custom visibile sul portale web di prodotto di MSSRS (home page di "Report Manager" e relativa navigazione).

6 SERVIZI IN SUPPORTO ALLA SOLUZIONE

6.1 Garanzia, manutenzione e assistenza del software

Il fornitore del software deve offrire un programma di manutenzione e assistenza per garantire che l'utente possa sfruttare tutti i benefici del software lungo tutto il suo ciclo di vita. Il programma deve prevedere una copertura assicurativa di base e un'estensione di garanzia per l'assistenza prioritaria e gli aggiornamenti software che verranno rilasciati. Deve essere fornita assistenza telefonica attraverso un numero gratuito nei normali orari di ufficio. L'assistenza deve essere accessibile anche via fax, posta elettronica o un sito web di assistenza tecnica.

6.1.1 Assistenza in garanzia

Il fornitore di software deve garantire i prodotti per un periodo di 90 giorni dopo la consegna. Durante il periodo di garanzia il fornitore deve offrire assistenza tecnica telefonica gratuita nei normali orari di lavoro attraverso un numero gratuito. Tutti i difetti del software devono essere corretti tempestivamente.

6.1.2 Assistenza estesa e manutenzione del software

Dopo un periodo di garanzia di 90 giorni, l'utente può continuare a ricevere assistenza tecnica tramite fax, e-mail o accesso a un sito web di assistenza tecnica. Per garantire che l'utente abbia sempre accesso alle versioni del software più recenti, alla garanzia a lungo termine e all'assistenza tecnica, il fornitore deve offrire un programma di assistenza estesa con un canone fisso annuale.

6.2 Aggiornamenti software

Il programma di assistenza estesa deve dare all'utente il diritto di ricevere le versioni più recenti del software del sistema di Telecontrollo e gli aggiornamenti delle versioni, man mano che vengono resi disponibili. Per garantire un'assistenza di qualità a tutti gli utenti, tutte le licenze software installate in un sito devono essere allineate alla stessa versione.

6.3 Supporto delle patch per il sistema operativo

Il fornitore deve testare e supportare le patch del sistema operativo rilasciate periodicamente da Microsoft. Il fornitore deve avere una politica di assistenza definita per le patch di sicurezza.

6.4 Assistenza telefonica

Il programma di assistenza estesa deve prevedere l'assistenza telefonica nei normali orari di lavoro. Tale assistenza telefonica verrà fornita da un tecnico dell'assistenza certificato dal fornitore software con un programma di formazione certificato. Le chiamate nei normali orari di lavoro devono essere raccolte da un addetto che fornisca assistenza telefonica illimitata. Non è accettabile un sistema di assistenza con segreteria telefonica.

6.5 Assistenza via posta elettronica

Il programma di assistenza estesa deve comprendere l'assistenza tramite posta elettronica con risposta entro un giorno lavorativo e una priorità superiore agli utenti senza assistenza in garanzia, inoltrando la richiesta al più vicino centro di assistenza tecnica certificato. L'assistenza elettronica deve inoltre offrire l'accesso a servizi avanzati sulla nostra pagina web dei servizi tecnici. Il programma di assistenza estesa deve prevedere l'accesso in tempo reale a problemi presenti e passati registrati in un database delle chiamate, oltre alla possibilità di aprire nuove segnalazioni che verranno assegnate immediatamente a un tecnico per la risoluzione.

6.6 Scaricamento di file

Il programma di assistenza estesa deve prevedere l'accesso a un sito web sicuro per lo scaricamento di file. Su questo sito web sicuro devono essere disponibili nuove release software, service pack, patch, server I/O aggiornati e altri file per gli utenti con servizio di assistenza.

6.7 Assistenza via Web

Il fornitore software deve avere un sito web dedicato allo sviluppo e all'assistenza, che fornisca informazioni tecniche sul software del sistema di Telecontrollo, best practice di implementazione e forums per gli utenti.

6.8 Newsletter e CD dell'assistenza tecnica

Il fornitore software deve distribuire una newsletter e un CD con note tecniche a tutti gli utenti abbonati al programma di assistenza estesa, almeno due volte all'anno. Il CD dell'assistenza tecnica deve contenere un riepilogo completo di note tecniche, allarmi, applicazioni, utility applicative, utility diagnostiche, controlli ActiveX, driver, script, funzioni script, wizard e consigli utili per agevolare lo sviluppo di applicazioni.

6.9 Retrocompatibilità del software

Il fornitore del software deve garantire la compatibilità a ritroso per un periodo non inferiore a 10 anni e un percorso di migrazione graduale per proteggere l'investimento. Le applicazioni obsolete devono consentire una facile migrazione alle nuove versioni del software senza modifiche tecniche.

7 SOLUZIONE

Di seguito vengono forniti alcuni esempi di come dovrà essere sviluppata l'applicazione relativamente alle specifiche di automazione e processo previste all'interno della progettazione.

7.1 Sviluppo Enti Impianto, sue sezioni e Asset di automazione

Per lo sviluppo dell'intero scada di impianto il sistema software deve seguire un approccio object oriented, dando l'opportunità allo sviluppatore di creare dei modelli (Templates) che contengono tutte le caratteristiche principali dell'assets reale previsto all'interno dello specifico impianto/zona in una logica di processo:

- Canali di Ingresso
- Canali di scarico in battigia
- Grigliatura media
- Nuovo Impianto di pretrattamento e pompaggio a mare - Sollevamento alla dissabbiatura
- Nuovo impianto di pretrattamento e pompaggio a mare – Grigliatura
- Nuovo impianto di pretrattamento e pompaggio a mare - Sollevamento al torrino di carico
- Impianto di pretrattamento esistente - Primo sollevamento
- Impianto di pretrattamento esistente - Sollevamenti per Cuma / Sollevamento provvisorio
- Condotte prementi

Inoltre, dovranno essere previsti tutti gli oggetti di automazione come nella seguente tabella:

<i>Legenda macchine e strumentazioni</i>	
PA	Paratoia manuale
PM	Paratoia motorizzata
PAA	Paratoia automatica
GR	Griglia
P	Pompa
DS	Dissabbiatore
V	Valvola
VG	Vaglio rotante
NS	Nastro trasportatore o coclea
KA	Gruppo compressore
CA	Carroponte
CS	Classificatore sabbia
LTU	Misuratore di livello a ultrasuoni
IL	Galleggiante
DL	Misuratore di livello differenziale
MP	Misuratore di portata
M	Misuratore di pressione

La metodologia intrinseca del prodotto deve quindi permettere la creazione di una libreria, contenente i modelli dei vari assets, che diventi patrimonio aziendale e come tale possa essere riutilizzata nei progetti futuri.

La definizione dei templates deve poter seguire due linee guida principali:

- derivazione
- contenimento

La derivazione consente di creare una struttura gerarchica, all'interno della quale le caratteristiche del template padre vengono trasferite ai templates figli, con l'opportunità di aggiungere, man mano che si percorre la struttura verso il basso, attributi e funzionalità al modello rendendolo sempre più specifico.

Il concetto di contenimento definisce il template come contenitore, all'interno del quale verranno definite le caratteristiche salienti dell'asset:

- anagrafica degli attributi di I/O (comandi, stati, misure, setpoints ecc.)
- logica (codice, nel linguaggio di scripting della Piattaforma) per implementare funzioni da eseguirsi lato-Server quando in Esercizio
- grafica (uno o più simboli grafici per molteplici rappresentazioni dello stesso Asset in contesti diversi)
- configurazione della storicizzazione (quali attributi di I/O sono storicizzati e con che policy)
- configurazione della security (quali profili-operatore possono accedere quali attributi, ed in che modo)
- configurazione della comunicazione (indipendente dal protocollo di comunicazione di Campo che sarà effettivamente usato in Esercizio)

Le Istanze create dai Templates, a livello "topologico", dovranno popolare una descrizione gerarchica, a "Modello d'Impianto", del patrimonio di Asset dell'utente finale, organizzato con la seguente modalità:

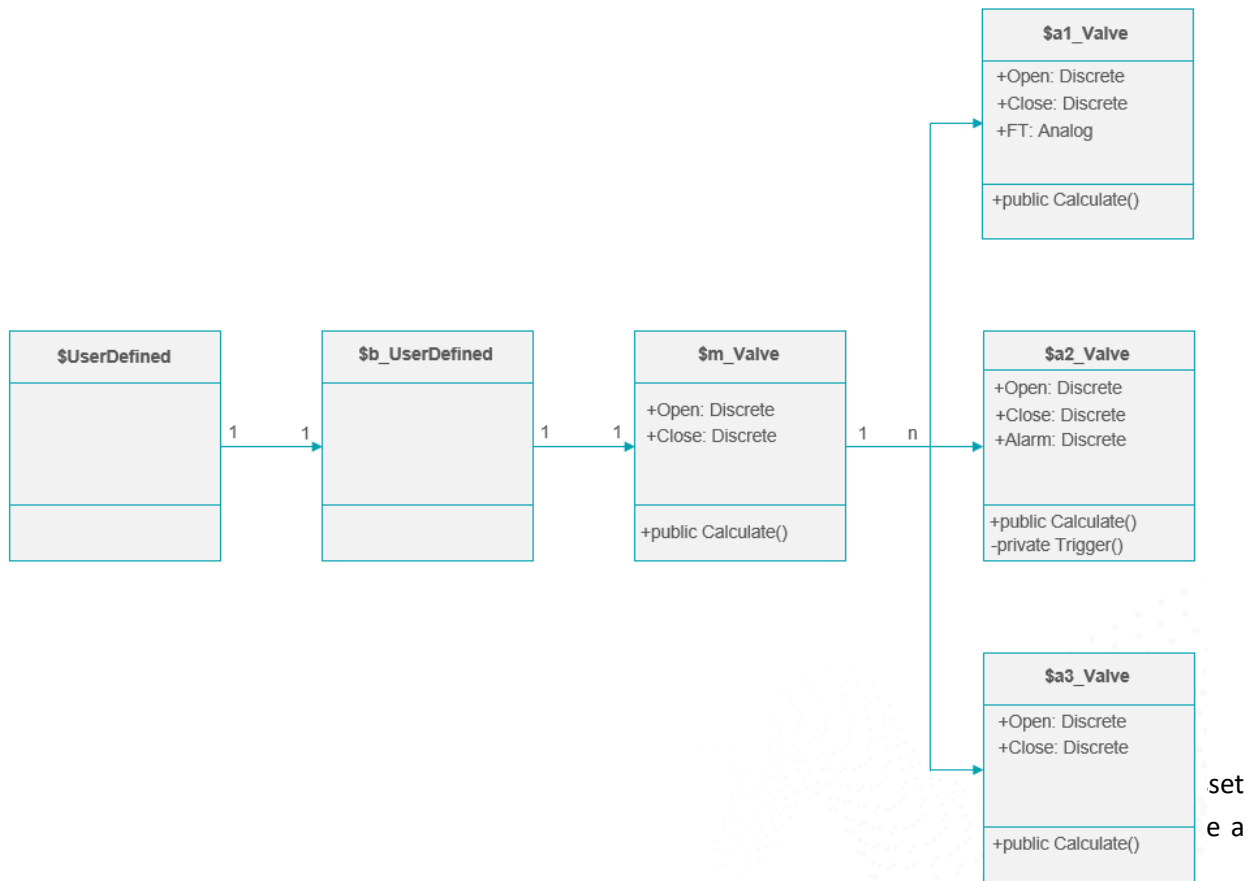
- Sito di Esercizio
 - Aree ...
 - Sotto-Aree ...
 - Equipment complessi ...
 - Equipment elementari (parti costituenti) ...
 - Singoli segnali (I/O)

Parallelamente ai templates deve essere possibile creare dei simboli grafici, che saranno poi inglobati nei modelli. Deve essere garantita la possibilità di creare simboli custom tramite l'interfaccia di sviluppo grafico, ma deve essere comunque disponibili delle librerie grafiche, anche di tipo "Situational Awareness" orientate a migliorare la user experience (UE).

7.2 Templates di esempio

In questo paragrafo vengono descritti i principi base dei Templates, e saranno mostrati alcuni esempi di template specifici per il caso in oggetto.

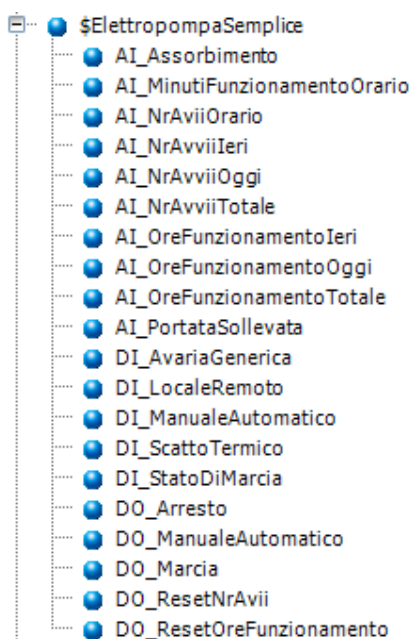
Di seguito viene proposto, utilizzando il diagramma delle classi, un esempio di utilizzo del System Template \$UserDefined per creare un modello relativo all'asset "Valvola".



Le caratteristiche configurate a livello master, vengono poi ereditate a livello template derivato ed istanze, con la possibilità di aggiungere funzionalità sempre più specifiche.

Ad esempio il template "ElettropompaSemplice" potrebbe essere realizzato come segue, utilizzando dei template di base più semplici:

Attributo	Tipo	Accesso
AI_Assorbimento	AnalogDeviceGenerico	Input
AI_MinutiFunzionamentoOrario	AnalogDeviceGenerico	Input
AI_NrAvviiOrario	AnalogDeviceGenerico	Input
AI_NrAvviiIeri	AnalogDeviceGenerico	Input
AI_NrAvviiOggi	AnalogDeviceGenerico	Input
AI_NrAvviiTotale	AnalogDeviceGenerico	Input
AI_OreFunzionamentoIeri	AnalogDeviceGenerico	Input
AI_OreFunzionamentoOggi	AnalogDeviceGenerico	Input
AI_OreFunzionamentoTotale	AnalogDeviceGenerico	Input
AI_PortataSollevata	AnalogDeviceGenerico	Input
DI_AvariaGenerica	DiscreteDeviceGenerico	Input
DI_LocaleRemoto	DiscreteDeviceGenerico	Input
DI_ManualeAutomatico	DiscreteDeviceGenerico	Input
DI_ScattoTermico	DiscreteDeviceGenerico	Input
DI_StatoDiMarcia	DiscreteDeviceGenerico	Input
DO_Arresto	DiscreteDeviceGenerico	Input\Output
DO_ManualeAutomatico	DiscreteDeviceGenerico	Input\Output
DO_Marcia	DiscreteDeviceGenerico	Input\Output
DO_ResetNrAvvii	DiscreteDeviceGenerico	Output
DO_ResetOreFunzionamento	DiscreteDeviceGenerico	Output



E potrebbe includere una o più rappresentazioni grafiche



Da questo template potrebbe derivare il template dell'

ElettropompaConAvviatore

Il template, oltre agli attributi/oggetti del template dell'Elettropompa semplice, potrebbe contenere i seguenti oggetti di automazione:

Attributi	Tipo	Accesso	Note
AI_ControlloVibrazioni	AnalogDeviceGenerico	Input	
AI_Frequenza	AnalogDeviceGenerico	Input	
AI_NrGiri	AnalogDeviceGenerico	Input	
AI_TemperaturaAvvolgimento	AnalogDeviceGenerico	Input	
AI_TemperaturaCuscinettiMotore	AnalogDeviceGenerico	Input	
AI_TemperaturaCuscinettiPompa	AnalogDeviceGenerico	Input	
AO_Velocità	AnalogDeviceGenerico	Input\Output	
DI_AvariaAzionamento	DiscreteDeviceGenerico	Input	Genera allarme
DI_GocciolamentoTenuta	DiscreteDeviceGenerico	Input	Genera allarme
DI_MarciaASecco	DiscreteDeviceGenerico	Input	Genera allarme
DI_Peridtaisolamento	DiscreteDeviceGenerico	Input	Genera allarme
DI_Umidità	DiscreteDeviceGenerico	Input	Genera allarme
DO_RegolazioneVelocità	DiscreteDeviceGenerico	Input\Output	

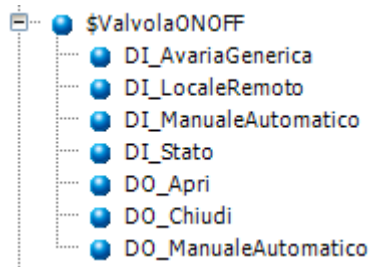
E la rappresentazione completa sarebbe quindi:

- \$ElettropompaConAvviatore
 - AI_Assorbimento
 - AI_ControlloVibrazioni
 - AI_Frequenza
 - AI_MinutiFunzionamentoOrario
 - AI_NrAvviiOrario
 - AI_NrAvviiIeri
 - AI_NrAvviiOggi
 - AI_NrAvviiTotale
 - AI_NrGiri
 - AI_NrInversioni
 - AI_OreFunzionamentoIeri
 - AI_OreFunzionamentoOggi
 - AI_OreFunzionamentoTotale
 - AI_PortataSollevata
 - AI_TemperaturaAvvolgimento
 - AI_TemperaturaCuscinettiMotore
 - AI_TemperaturaCuscinettiPompa
 - AO_AssorbimentoInversione
 - AO_Velocità
 - DI_AvariaAzionamento
 - DI_AvariaGenerica
 - DI_GocciolamentoTenuta
 - DI_LocaleRemoto
 - DI_ManualeAutomatico
 - DI_MarciaASecco
 - DI_PeridtaIsolamento
 - DI_ScattoTermico
 - DI_StatoDiMarcia
 - DI_Umidità
 - DO_Arresto
 - DO_Inversione
 - DO_ManualeAutomatico
 - DO_Marcia
 - DO_RegolazioneVelocità
 - DO_ResetNrAvvii
 - DO_ResetOreFunzionamento

La rappresentazione grafica dell'ente potrebbe essere la stessa del template padre, o essere diversa per riflettere gli oggetti/attributi specifici del template derivato.

Un ulteriore esempio:

Ipotizzando che il template Valvola ON OFF contenga i seguenti oggetti di automazione:



Il template della ValvolaRegolatrice potrebbe estendere il template di Valvola ON OFF con i seguenti oggetti di automazione:

