SCUOLA DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di laurea in INFORMATICA (L-31)

Anno accademico 2024/2025

**Progettazione e realizzazione di un sistema Software per la mappatura degli**

**elementi di un impianto ferroviario**

**Candidato:** Perricone Giuseppe

**Relatore**

Prof. Rovetta Stefano

**Co – Relatore**

Fabrizio Galletto

**Co – Tesista**

Bestoso Gabriele

Sommario

[Abstract ii](#_Toc212632865)

[1. Introduzione 1](#_Toc212632866)

# Abstract

In ambito dei moderni sistemi ferroviari, garantire la coerenza e l’affidabilità dei dati è fondamentale per assicurare la sicurezza operativa e il corretto funzionamento degli impianti. In questo contesto, strumenti differenti possono produrre rappresentazioni diverse di una medesima configurazione di stazione, con il rischio di introdurre discrepanze e potenziali errori nelle fasi di progettazione o manutenzione.

Il software da me sviluppato prende in input due file XML, prodotti da due tool esterni e contenenti due differenti configurazioni dello stesso impianto, analizzandone il contenuto di ognuno, con lo scopo ultimo di produrre come output un file di report, contenente l’esito dell’analisi, un file di log delle operazioni eseguite e un file di mappatura degli identificativi numerici degli elementi corrispondenti nei due file differenti.

Poiché il contesto applicativo ricade nell’ambito dei sistemi safety-critical, è stato inoltre affrontato il tema della *software diversity*.

A tal fine si cerca di verificare che lo stesso algoritmo, implementato in due linguaggi differenti deve andare a garantire lo stesso medesimo funzionamento e output. Ho quindi preso il progetto implementato precedentemente in C# dal mio collega “Bestoso Gabriele” e l’ho sviluppato nuovamente da zero in python (per vedere nel dettaglio il progetto andare al capitolo dedicato …). In seguito, abbiamo controllato e standardizzato l’output risultante, in modo da avere uno standard da seguire ed essere certi che l’output fosse identico in modo da facilitarne anche il controllo.

Infine, per entrambi i progetti è stata realizzata una suite di test automatici, con lo scopo di validare il comportamento del software rispetto ai requisiti iniziali e assicurare l’affidabilità complessiva del sistema

# 1. Introduzione

Nell’ambito della progettazione di sistemi complessi, informatici e non, la gestione corretta e coerente dei dati rappresenta un elemento fondamentale. Dati inconsistenti, incompleti o formattati in modo non uniforme possono compromettere l’affidabilità delle applicazioni e richiedere attività di rielaborazione che incidono sia sui tempi sia sulla qualità del risultato finale. Garantire integrità, coerenza e tracciabilità delle informazioni è quindi essenziale per assicurare il corretto funzionamento dei sistemi e per evitare errori che potrebbero propagarsi nelle fasi successive del processo.

Nel presente elaborato si andrà a trattare il caso specifico dei sistemi ferroviari.

In questo ambito la gestione della configurazione di un impianto rappresenta uno degli aspetti centrali. Infatti, ogni impianto viene descritto da una grande quantità di informazioni strutturate, che devono rimanere coerenti lungo tutto il ciclo di vita del sistema: dalla progettazione alla messa in servizio, fino alle attività di manutenzione ed evoluzione dell’infrastruttura, stiamo quindi parlando di un dominio molto complesso, dove la quantità di informazioni è elevatissima. La complessità del dominio, unita ai rigorosi requisiti di sicurezza propri del settore, rende essenziale disporre di strumenti affidabili per la verifica e il controllo della consistenza dei dati.

In questo contesto nasce il progetto da me trattato in questo elaborato.

Sono presenti due tool esterni differenti, denominati come TOOL\_A e TOOL\_B, entrambi producono una documentazione molto dettagliata dell’impianto ferroviario di Stroncone in formato xml. Tuttavia, i due tool operano su due basi di dati differenti generando quindi due file che possiedono la stessa medesima struttura, e quindi compatibili, ma vi è il rischio, che a stessi elementi dell’impianto vengano assegnati identificativi numerici, e dettagli generali, differenti tra i due file. Ciò è molto pericoloso, perché può portare a discrepanze e incoerenze nella documentazione, con potenziali ripercussioni sulle attività di manutenzione e, più in generale, sull’efficienza e la sicurezza dell’infrastruttura.

L’obiettivo del lavoro presentato è quindi la progettazione e lo sviluppo di un sistema software, implementato in linguaggio C#, in grado di confrontare le due configurazioni e di individuare eventuali incongruenze. Il software, in particolare, deve essere in grado di:

* Analizzare in input i due file XML contenenti le configurazioni dell’impianto;
* Andare comparare in modo efficiente e sicuro i due file;
* Generare un report testuale che contenga tutte le varie problematiche che sono state individuate
* Generare un file di log che contenga tutte le operazioni che vengono svolte all’interno del programma. Ciò serve per garantire quella che viene definita come tracciabilità delle informazioni
* Generare un file di mapping, che metta in corrispondenza gli identificativi numerici dei vari elementi, facilitando il confronto e la consultazione umana dei dati.

Dopo la realizzazione del software è stato affrontato anche l’aspetto legato al contesto safety-critical in cui esso si inserisce.

A differenza di molti ambiti di progettazione software, l’ambito ferroviario opera in un contesto dove sono richiesti livelli molto elevati di affidabilità, in quanto eventuali incoerenze o malfunzionamenti potrebbero avere conseguenze rilevanti sul piano operativo e sulla sicurezza di terzi. In questo contesto è stata messo sotto osservazione il concetto di *software diversity*, secondo cui lo stesso algoritmo viene implementato su differenti ambienti di sviluppo, quindi diversi linguaggi di programmazione, piattaforme o stili implementativi, con lo scopo di produrre in output gli stessi medesimi risultati in modo tale da ridurre la probabilità di errori comuni o correlati. A tal fine, partendo dal progetto originariamente sviluppato in C# dal collega Bestoso Gabriele, è stata realizzata in maniera completamente autonoma una seconda implementazione in linguaggio Python, basata unicamente sulle medesime specifiche funzionali richieste.

Completata la re-implementazione, è stata effettuata una fase di verifica del “doppio canale”, confrontando gli output prodotti dalle due versioni del software. Per facilitare tali confronti e consentire eventuali controlli automatici futuri, è stato definito e applicato un formato standardizzato di output, così da garantire uniformità nei risultati e facilitarne la confrontabilità.

Infine, per entrambe le implementazioni è stata sviluppata una suite di unit test, rispettivamente in C# e Python, con l’obiettivo di verificare il rispetto dei requisiti funzionali e aumentare l’affidabilità complessiva del sistema.