

Ricerca di Sistema elettrico



SCP: Implementazione dell'interoperabilità dell'ontologia SCPS
con altre specifiche Smart City (LA2.4)

Cristiano Aguzzi



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

SCP: Implementazione dell'interoperabilità dell'ontologia SCPS con altre specifiche Smart City

LA 2.4 SCP: Implementazione dell'interoperabilità dell'ontologia SCPS con altre specifiche Smart City

C. Aguzzi (DISI)

Dicembre 2024

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA Piano Triennale di Realizzazione 2022-2024

Obiettivo: Decarbonizzazione

Progetto: Tema di ricerca 1.7 – Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Linea di attività: 2.4

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work Package: Claudia Meloni, ENEA

Responsabile Linea di Attività: Michela Milano, UniBO DISI

Mese inizio previsto: 13

Mese inizio effettivo: 13

Mese fine previsto: 36

Mese fine effettivo: 36

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione: "Ontologia SCPS: controllo del ciclo di pubblicazione e interoperabilità con altre ontologie Smart City"

Indice

1	Risultati attesi	4
2	Risultati ottenuti.....	5
2.1	Test e Continuous Integration (CI) e Continuous Deployment/Delivery (CD)	5
2.2	Processo interattivo di modifica e sviluppo di nuove funzionalità.....	5
2.3	Versionamento e ontologia	6
2.4	Progettazione supporto a JSON-LD nella generazione di UrbanDataset templates...	6
2.5	Risoluzione bug	7
2.6	Deploy e manutenzione.....	7
2.7	Attività di gestione e documentazione	7
2.8	Pianificazione possibili attività future	7
3	Prodotti attesi	8
4	Prodotti sviluppati	9
5	Analisi degli scostamenti su attività e risultati.....	10
6	Sintesi delle attività svolte	11
7	Dettaglio delle attività svolte.....	12
7.1	Test Automatizzati e Continuous Integration/Continuous Deployment	12
7.2	Sviluppo di Nuove Funzionalità	12
7.3	JSON-LD analisi e progettazione	13
7.3.1	Fase 1: Introduzione del modello senza SOSA.....	14
7.3.2	Fase 2: Transizione Graduale verso JSON-LD.....	14
7.3.3	Fase 3: Adozione di SOSA e Integrazione con Sistemi W3C-compliant	14
7.4	Gestione del Versionamento dell'Ontologia.....	17
7.5	Risoluzione di Bug e Ottimizzazione del Sistema	17
7.6	Aggiornamenti e Deploy dell'Applicazione.....	17
7.7	Documentazione e Report	17
7.8	Progettazione Futura.....	18
8	Conclusione.....	19

Indice delle figure

Figura 1 - Gestione del bilinguismo e aggiunta del campo Note.....	6
Figura 2. Esempio di esecuzione test d'integrazione.....	12
Figura 3. Messaggio UrbanDataset in JSON-LD utilizzando l'allineamento con SOSA	15
Figura 4. UrbanDataset in JSON-LD ma mantenendo il più possibile la struttura corrente	16
Figura 5. Fotogramma da uno dei video di analisi del codice	18

1 Risultati attesi

I risultati attesi della linea di attività prevedono lo sviluppo software (codici sorgenti, eseguibili, librerie utilizzate, etc.) della Web Application basata su tecnologia Spring e Gradle per l'editing e la consultazione dell'ontologia in oggetto.

Tali sviluppi software sono pensati con il fine di:

- permettere la gestione della creazione e pubblicazione di versioni differenti dell'ontologia (o anche supporto al versionamento);
- di implementare il supporto a JSON-LD nella generazione di UrbanDataset templates;
- di implementare il supporto alla trasformazione di dati da un insieme di altri applicativi selezionati (es. FIWARE Data Model, SAREF, PNDI) verso il formato UrbanDataset
- di rendere interattivo il processo di proposta modifiche. Invece di mandare a mano il documento Excel creare una piattaforma in cui gli admin ricevono notifiche di modifica che possono gestire.

2 Risultati ottenuti

Durante il periodo di attività, abbiamo raggiunto una serie di risultati significativi che contribuiscono al consolidamento e all'ottimizzazione dell'applicativo UrbanDataset Web Library. In questa sezione, evidenzieremo in che misura ciascun risultato è stato ottenuto e il beneficio che ne deriva per il sistema elettrico nazionale e i suoi utenti. Inoltre, includeremo eventuali risultati non previsti nel capitolato.

I principali beneficiari dei risultati ottenuti sono:

- **L'amministrazione comunale:** grazie alla *UrbanDataset WebLibrary*, l'amministrazione ha a disposizione un linguaggio condiviso costituito dagli *UrbanDataset*. Questo consente di richiedere dati alle aziende, ad esempio sull'uso dell'energia elettrica negli edifici cittadini, evitando il rischio di *vendor lock-in* e garantendo l'utilizzo di un formato unificato.
- **I produttori di software (solution providers):** la *WebLibrary* permette un accesso semplificato agli *UrbanDataset* definiti dagli SCPS, agevolando lo sviluppo di applicazioni, come strumenti per il monitoraggio dei consumi elettrici. Questo approccio favorisce la replicabilità delle soluzioni software da una città all'altra.
- **I cittadini:** possono beneficiare di maggiore trasparenza e di nuovi servizi che la municipalità può implementare e rendere disponibili grazie agli *UrbanDataset*.

I risultati di questo SAL (Stato Avanzamento Lavori) hanno portato diversi vantaggi:

- **Migliore usabilità:** l'applicativo è ora più semplice ed efficiente da utilizzare, grazie al miglioramento dell'interfaccia utente.
- **Maggiore affidabilità e stabilità:** il potenziamento dei test automatizzati e l'operazione di eliminazione dei big hanno reso l'applicativo più robusto, riducendo il rischio di errori e interruzioni operative.

I risultati raggiunti nel secondo SAL rappresentano un passo significativo verso gli obiettivi del progetto e sottolineano il nostro impegno nel fornire un'applicazione di qualità per la gestione dei dati urbani in ambito energetico e Smart City. La digitalizzazione delle infrastrutture energivore ha infatti permesso di creare una base di conoscenza standardizzata, utile per una gestione più efficiente e sostenibile, con conseguente risparmio energetico.

2.1 Test e Continuous Integration (CI) e Continuous Deployment/Delivery (CD)

Potenziamento dei test automatizzati che vengono eseguiti ogni volta che viene effettuato il rilascio di una nuova versione dell'applicazione su Git-Lab.

2.2 Processo interattivo di modifica e sviluppo di nuove funzionalità

È stato implementato un sistema che permette di notificare i responsabili dell'ontologia quando viene proposto un nuovo UrbanDataset, tramite invio di indirizzo e-mail.

Inoltre, sono state implementate funzionalità significative inizialmente non previste, come il bilinguismo (italiano e inglese) nell'interfaccia grafica e nei template generati. È stato anche migliorato il sistema di editing parallelo che era stato implementato nel primo SAL perché era risultato in efficace.

Sono stati inoltre introdotti nuovi campi, come "Note", e pulsanti aggiuntivi nelle interfacce per migliorare la gestione delle proposte (Figura 1).

UrbanDataset WEB LIBRARY

UrbanDataset Search

HOME ONTOLOGY SERVICES EDITOR PROPOSE

Italian English

Suggest updates

On this page, users can request changes to the ontology. Currently, from the page, it is possible to obtain an Excel report containing the completed request. The report must then be sent via email to info@enea.it where it will be validated and compared with other requests. Possible changes may include adding a new UrbanDataset or modifying an existing one. Additionally, in the completion, it is possible to request the addition of properties that are not defined in the current ontology.

Scenario

UrbanDataset Name: UrbanDataset Name:

Update UrbanDataset name: Use an already existing UrbanDataset

Scope:

In what context will the UrbanDataset be used

Category: Select an option

Main category of involved applications

Notes

Insert additional context about why you are requesting these changes

Properties

In this form, you need to describe the characteristics of line properties.

Property	Description	Default Unit	Type	Required	Actions
ApplianceCode	Type of appliance connected to the smart plug/switch (e.g. washing machine, television, lamp)		string	✓	✖
ElectricConsumption	The electric energy consumption of the appliance in the period	kilowattHour	double	✓	✖
RoomCode	A code identifying the type of room (e.g. kitchen, bathroom)		string		

Figura 1 - Gestione del bilinguismo e aggiunta del campo Note

2.3 Versionamento e ontologia

Un'importante discussione ha riguardato il sistema di versionamento, che è stato definito a tre livelli: modifiche non compatibili, parzialmente compatibili e pienamente compatibili. È stato anche valutato l'uso di namespace versionati per supportare multiple versioni di ontologie, consentendo così un'integrazione più flessibile con applicazioni esterne.

2.4 Progettazione supporto a JSON-LD nella generazione di UrbanDataset templates

Negli ultimi anni, JSON-LD si è affermato come standard per la rappresentazione di dati strutturati e semantici, migliorando l'interoperabilità tra sistemi e l'integrazione con il Web Semantico. Per supportare questo formato, abbiamo avviato un progetto di aggiornamento dei template UrbanDataset, valutando le modifiche necessarie e l'impatto sul processo di validazione. La transizione prevede un approccio graduale: inizialmente si introdurrà JSON-LD senza modificare il modello esistente, successivamente si migreranno i sistemi e si deprecherà il vecchio formato JSON, per poi integrare l'ontologia SOSA e favorire l'interoperabilità con ecosistemi W3C-compliant come Fiware.

2.5 Risoluzione bug

Numerosi bug sono stati individuati e risolti, tra cui problemi con caratteri speciali, spazi errati negli URL e duplicazioni nei commenti dell'ontologia. È stata migliorata la validazione semantica e XSD, assicurando la conformità del sistema agli standard previsti.

2.6 Deploy e manutenzione

Il deploy dell'applicazione è stato aggiornato periodicamente, risolvendo problemi di coerenza del codice e garantendo l'introduzione di nuove funzionalità. Durante questi aggiornamenti, particolare attenzione è stata data alla verifica dei bug riscontrati e alla loro risoluzione.

2.7 Attività di gestione e documentazione

Sul fronte della documentazione, sono stati prodotti report dettagliati che includono descrizioni delle attività svolte, contatti utili e proposte per sviluppi futuri. Questi report sono stati arricchiti con immagini e video esplicativi, utilizzati anche per agevolare il passaggio di consegne tra collaboratori.

In particolare è stato registrato un ciclo di video che analizzano nel dettaglio la struttura e il codice del software.

Le comunicazioni tra il team sono state continue, con un intenso scambio di e-mail e call per discutere le problematiche tecniche e i progressi. Questo ha consentito di mantenere una chiara visione delle priorità e dei risultati raggiunti.

2.8 Pianificazione possibili attività future

Infine, sono state proposte e pianificate diverse funzionalità future, come notifiche automatizzate per gli amministratori e il supporto a formati come JSON-LD. Anche la gestione delle query e il miglioramento del sistema di versionamento sono stati identificati come ambiti di interesse per i prossimi sviluppi.

3 Prodotti attesi

Il prodotto atteso per la LA2.4 oltre è la versione aggiornata dell'Applicativo Web UrbanDataset Web Library.

4 Prodotti sviluppati

L'applicativo Web "Urban Dataset Web Library" è stato aggiornato a una nuova versione con nuove funzionalità e consolidata.

5 Analisi degli scostamenti su attività e risultati

Rispetto a quanto pianificato non sono è stata sviluppata l'implementazione al supporto alla trasformazione di dati da un insieme di altri applicativi selezionati. Inoltre, la gestione del versionamento non è stata implementata ma solo discussa e analizzata a livello teorico fino ad arrivare alla scelta di una nuova soluzione rispetto a quella ipotizzata inizialmente.

Questo scostamento è stato dovuto alla maggior quantità di lavoro richiesta per gli aggiornamenti software dell'applicativo. Durante le attività si è, infatti, ritenuto prioritario lavorare sul consolidamento dell'applicazione che ormai viene usata ampiamente da diversi soggetti (sia città, che aziende) interessati a utilizzare gli UrbanDataset e dunque deve avere livelli di affidabilità e robustezza molto alti.

Inoltre Cristiano Aguzzi che seguiva le attività ha dato le dimissioni dal suo contratto di assegno di ricerca prima della fine del progetto e non ha potuto completare il lavoro previsto.

6 Sintesi delle attività svolte

Il progetto ha affrontato con successo sfide tecniche e gestionali, migliorando la stabilità e l'usabilità di un'applicazione web. Sono stati implementati test automatizzati tramite runner e risolti problemi complessi legati a Docker e validazione semantica. Si è avviato un progetto di aggiornamento dei template UrbanDataset in modo che utilizzi JSON-LD, valutando le modifiche necessarie e l'impatto sul processo di validazione. Funzionalità chiave che sono state sviluppate includono il bilinguismo dell'interfaccia, la gestione sicura dell'editing parallelo e un sistema di notifiche via e-mail per le proposte. Sono stati corretti bug critici come caratteri speciali negli URL, duplicazioni nei commenti ed errori di validazione. È stata analizzato il problema del sistema di versionamento, arrivando a definire, a livello di progetto, un sistema a tre livelli che possa permettere gestire le evoluzioni dell'ontologia. La documentazione, arricchita da report e video esplicativi, ha facilitato il coordinamento e posto le basi per future implementazioni. Grazie a un approccio collaborativo, il progetto ha prodotto un sistema stabile e scalabile, garantendo continuità e prospettive di sviluppo.

7 Dettaglio delle attività svolte

Durante le varie fasi del progetto, il team ha affrontato numerose sfide tecniche e organizzative, ottenendo risultati concreti in termini di stabilità del sistema, nuove funzionalità e miglioramenti nella gestione del software. Grazie a un approccio collaborativo e all'adozione di tecniche iterative, ogni attività ha prodotto avanzamenti misurabili, che vengono dettagliati di seguito.

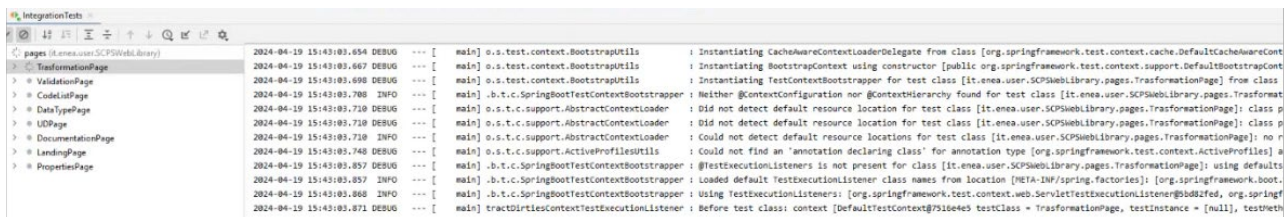
Il progetto ha richiesto un'intensa comunicazione tra i membri del team, che si è concretizzata in frequenti call ed e-mail per discutere problematiche tecniche e definire le priorità. Questo approccio collaborativo ha permesso di mantenere un allineamento costante sulle attività e sulle scadenze.

Le decisioni sono state prese rapidamente e in modo condiviso, migliorando l'efficienza complessiva. La trasparenza nella comunicazione ha ridotto al minimo le incomprensioni e favorito una collaborazione efficace tra i membri del team.

7.1 Test Automatizzati e Continuous Integration/Continuous Deployment

L'introduzione di test automatizzati è stata affrontata configurando runner dedicati per GitLab. Sono state eseguite analisi sui problemi riscontrati con Docker-in-Docker ed è stata trovata una soluzione temporanea per eseguire i test in ambienti locali, mentre si cercava una configurazione permanente sul registry di ENEA. Inoltre, è stato aggiunto un blocco try-catch nel codice per gestire eventuali eccezioni e prevenire crash.

Questa attività ha permesso di automatizzare il controllo del comportamento delle pagine e dei dati trasmessi, migliorando l'affidabilità generale dell'applicazione. I crash sono stati ridotti significativamente, e il processo di test è stato stabilizzato per le esecuzioni locali, rendendo più rapido il rilevamento di bug.



```
2024-04-19 15:43:03.654 DEBUG --- [ main ] o.s.t.c.support.BootstrapUtils : Instantiating CacheAwareContextLoaderDelegate from class [org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate]
2024-04-19 15:43:03.667 DEBUG --- [ main ] o.s.t.c.support.BootstrapUtils : Instantiating BootstrapContext using constructor [public org.springframework.test.context.support.DefaultBootstrapContext(java.lang.Class,org.springframework.test.context.CacheAwareContextLoaderDelegate)]
2024-04-19 15:43:03.698 DEBUG --- [ main ] o.s.t.c.support.BootstrapUtils : Instantiating TestContextBootstrapper for test class [it.enea.user.SCPSWebLibrary.pages.TrasformationPage] from class [org.springframework.test.context.support.AnnotationMethodTestContextBootstrapper]
2024-04-19 15:43:03.708 INFO --- [ main ] .b.t.c.SpringBootTestContextBootstrapper : Neither @ContextConfiguration nor @ContextHierarchy found for test class [it.enea.user.SCPSWebLibrary.pages.TrasformationPage], using default
2024-04-19 15:43:03.718 DEBUG --- [ main ] o.s.t.c.support.AbstractContextLoader : Did not detect default resource location for test class [it.enea.user.SCPSWebLibrary.pages.TrasformationPage]: class path resource [it/enea/user/SCPSWebLibrary/pages/TrasformationPage.properties] not found
2024-04-19 15:43:03.719 INFO --- [ main ] o.s.t.c.support.AbstractContextLoader : Did not detect default resource location for test class [it.enea.user.SCPSWebLibrary.pages.TrasformationPage]: class path resource [TrasformationPage.properties] not found
2024-04-19 15:43:03.748 DEBUG --- [ main ] o.s.t.c.support.ActiveProfilesUtils : Could not find an 'annotation declaring class' for annotation type [org.springframework.test.context.ActiveProfiles], using defaults
2024-04-19 15:43:03.857 DEBUG --- [ main ] .b.t.c.SpringBootTestContextBootstrapper : @TestExecutionListeners is not present for class [it.enea.user.SCPSWebLibrary.pages.TrasformationPage]: using defaults
2024-04-19 15:43:03.857 INFO --- [ main ] .b.t.c.SpringBootTestContextBootstrapper : Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener, org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener, org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener, org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcWebDriverTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcTestExecutionListener, org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.AnnotationMethodTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DelegatingTestExecutionListener, org.springframework.test.context.support.SpringJUnit4TestRunner$SpringMethodTestExecutionListener]
2024-04-19 15:43:03.868 INFO --- [ main ] .b.t.c.SpringBootTestContextBootstrapper : Using TestExecutionListeners: [org.springframework.web.servlet.TestExecutionListener@b082fed, org.springframework.test.context.support.AnnotationMethodTestExecutionListener@7916e4e5, org.springframework.test.context.support.DelegatingTestExecutionListener@7916e4e5]
2024-04-19 15:43:03.871 DEBUG --- [ main ] tractors.t.c.SpringBootTestContextBootstrapper : Before test class: context [DefaultTestContext@7916e4e5 testClass = TrasformationPage, testInstance = null, testMethod = null]
```

Figura 2. Esempio di esecuzione test d'integrazione

7.2 Sviluppo di Nuove Funzionalità

Per supportare un'utenza internazionale, è stata sviluppata un'interfaccia bilingue (italiano e inglese), lavorando sul multilinguismo a livello di ontologia e template.

La gestione dell'editing parallelo, che era risultata problematica con la versione progettata al primo SAL, è stata risolta introducendo meccanismi di avviso e blocco, basati su condizioni verificabili in tempo reale. Inoltre, sono stati aggiunti pulsanti e campi utili come "Note" per migliorare la gestione delle proposte.

L'applicazione è ora accessibile a un pubblico più ampio grazie al bilinguismo, e il rischio di conflitti durante l'editing parallelo è stato eliminato, migliorando la sicurezza e la coerenza dei dati. La gestione delle proposte è stata ottimizzata, consentendo agli utenti di fornire motivazioni e migliorando la comunicazione con gli amministratori.

7.3 JSON-LD analisi e progettazione

Negli ultimi anni, il formato JSON-LD si è affermato come uno standard chiave per la rappresentazione di dati strutturati e semantici sul Web. JSON-LD è compatibile con JSON ma offre la possibilità di collegare le informazioni a ontologie esistenti e di rappresentare i dati secondo il modello RDF (Resource Description Framework). Questo consente un'integrazione più efficace con il Web Semantico e favorisce l'interoperabilità tra sistemi diversi.

A inizio anno, ci siamo impegnati a migliorare la generazione di template per il formato UrbanDataset, analizzando le modifiche necessarie per supportare JSON-LD. Questa modifica potrebbe anche influenzare il processo di validazione dei dati, richiedendo aggiornamenti nei vari servizi di validazione offerti correntemente dall'applicativo. L'approccio corrente del presentare gli UrbanDataset attraverso JSON ha le seguenti limitazioni:

- Mancanza di contesto esplicito: Le definizioni delle proprietà sono separate dal payload e non sono direttamente interpretabili da strumenti di elaborazione semantica.
- Maggiore complessità nella trasformazione dei dati e interoperabilità limitata: La conversione bidirezionale tra il formato attuale e RDF richiede strumenti dedicati, aumentando il rischio di perdita di informazioni o interpretazioni errate e limitando l'integrazione con tool esistenti di processing semantico.
- Difficoltà nell'arricchimento semantico e nella condivisione dei dati: Senza l'uso di JSON-LD, è più complesso integrare e riutilizzare i dati in ambienti eterogenei, limitando la possibilità di creare collegamenti con altre fonti di dati aperti e arricchire il dataset con informazioni esterne.

L'adozione di JSON-LD porta diversi benefici:

- Aggiunta di un @context esplicito: Permette di dichiarare le definizioni delle proprietà direttamente nel documento, rendendolo autoesplicativo.
- Maggiore interoperabilità: I dati possono essere facilmente integrati in altri sistemi semantici e facilmente interrogati con SPARQL

La transizione da JSON a JSON-LD per la rappresentazione dei dati in UrbanDataset può essere realizzata tramite diverse strategie, ognuna con vantaggi e svantaggi specifici. A seconda delle esigenze di interoperabilità, impatto sui sistemi esistenti e complessità del processo di validazione, è possibile adottare un approccio radicale o uno più conservativo. In quello più conservativo si tratta di attenersi il più possibile alla struttura esistente e tentare di descrivere i vari punti di misurazione (il campo line del formato corrente) attraverso un oggetto json-ld con riferimenti alle proprietà dell'ontologia. Nell'approccio più radicale si potrebbe allineare l'ontologia degli UrbanDataset a SOSA (ontologia W3C nata appunto per descrivere misurazioni di valori dal mondo reale) e così rendere "standard" la descrizione dei valori descritti nel UrbanDataset. Ovviamente l'approccio radicale renderebbe necessaria una migrazione di tutti i servizi di validazione dell'applicativo e anche dei vari microservizi e applicazioni che utilizzano l'ontologia degli UrbanDataset come formato di scambio messaggi.

La transizione a JSON-LD potrebbe avvenire in modo graduale, partendo dall'introduzione del modello senza SOSA e parallelamente al formato esistente. Man mano che gli altri applicativi migreranno al nuovo modello, si inizierà a deprecare il vecchio formato JSON, per poi adottare

definitivamente JSON-LD. In seguito, verrà integrato anche il modello SOSA, per aprire la porta all'integrazione con altri sistemi W3C-compliant come Fiware.

7.3.1 Fase 1: Introduzione del modello senza SOSA

- Definizione del @context JSON-LD per UrbanDataset senza SOSA, mantenendo il modello esistente.
- Mappatura delle proprietà esistenti con le ontologie di riferimento.
- Integrazione di JSON-LD nei sistemi esistenti come opzione aggiuntiva, senza modificare l'architettura dei dati.
- Monitoraggio della transizione iniziale e raccolta di feedback dagli utenti.

7.3.2 Fase 2: Transizione Graduale verso JSON-LD

- Progressiva migrazione degli applicativi e dei sistemi verso il formato JSON-LD senza SOSA.
- Deprecazione del vecchio formato JSON una volta che la maggior parte degli applicativi avrà adottato JSON-LD.
- Aggiornamento del processo di validazione per supportare esclusivamente JSON-LD.
- Continuazione del monitoraggio e della raccolta di feedback per garantire una transizione fluida.

7.3.3 Fase 3: Adozione di SOSA e Integrazione con Sistemi W3C-compliant

- Integrazione del modello SOSA come rappresentazione alternativa per la gestione dei dati sensoriali, per aumentare l'interoperabilità con sistemi IoT avanzati e altre piattaforme semantiche.
- Implementazione di un framework per l'integrazione con sistemi W3C-compliant come Fiware, utilizzando SOSA come base per la rappresentazione dei dati.
- Monitoraggio dell'integrazione e miglioramento continuo delle funzionalità di interoperabilità.

```

{
  "@context": {
    "schema": "http://schema.org/",
    "sosa": "http://www.w3.org/ns/sosa/",
    "scps": "smartcityplatform:enea:semantic:ontology:urbandataset#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "unitOfMeasure": "qudt:unit",
    "value": "schema:value"
  },
  "@type": ["sosa:ObservationCollection", "scps:AirMonitoring"],
  "@id": "urn:dataset:urban:airmonitoring:12354",
  "sosa:hasMember": [
    {
      "@type": "sosa:Observation",
      "@id": "urn:observation:1",
      "sosa:observedProperty": {
        "@id": "scps:C02"
      },
      "sosa:resultTime": "2000-12-31T00:00:00",
      "sosa:hasSimpleResult": {
        "@value": " ",
        "@type": "xsd:double"
      },
      "unitOfMeasure": "qudt:PartsPerMillion"
    },
    {
      "@type": "sosa:Observation",
      "@id": "urn:observation:2",
      "sosa:observedProperty": {
        "@id": "scps:NO2"
      },
      "sosa:resultTime": "2000-12-31T00:00:00",
      "sosa:hasSimpleResult": {
        "@value": " ",
        "@type": "xsd:double"
      },
      "unitOfMeasure": "qudt:MicrogramPerCubicMeter"
    }
  ]
}

```

Figura 3. Messaggio UrbanDataset in JSON-LD utilizzando l'allineamento con SOSA

```

{
  "@context": {
    "scps": "smartcityplatform:enea:semantic:ontology:urbandataset#",
    // other context values
    "unit": "https://schema.org/unitCode"
  },
  "UrbanDataset": {
    "values": {
      "line": [
        {
          "@type": ["scps:AirMonitoring"],
          "period": {
            "start_ts": "2000-12-31T00:00:00",
            "end_ts": "2000-12-31T23:59:00"
          },
          "CO2": {
            "@value": 4.2,
            "unitCode": "ppm"
          },
          "NO2": {
            "@value": 50,
            "unitCode": "ppb"
          },
          "PM10": {
            "@value": 35.5,
            "unitCode": "ug/m3"
          },
          "PM2": {
            "@value": 25.1,
            "unitCode": "ug/m3"
          },
          "SO2": {
            "@value": 12.4,
            "unitCode": "ppb"
          },
          "LocationID": "Station-001",
          "MonitoringStationName": "Downtown Monitoring Station"
        }
      ]
    }
  }
}

```

Figura 4. UrbanDataset in JSON-LD ma mantenendo il più possibile la struttura corrente

7.4 Gestione del Versionamento dell'Ontologia

Il sistema di versionamento è stato definito basandosi su una strategia a tre livelli (non compatibile, parzialmente compatibile, compatibile). Il team ha discusso l'uso di namespace versionati per consentire il supporto di diverse versioni dell'ontologia, analizzando i pro e i contro di questa implementazione. È stata posta particolare attenzione all'adeguamento delle query per gestire il versionamento in modo efficiente.

È stato definito un framework concettuale per il versionamento, che consentirà una gestione più ordinata delle ontologie in futuro. Sebbene l'implementazione non sia stata completata, la base teorica è pronta per essere sviluppata nelle fasi successive.

7.5 Risoluzione di Bug e Ottimizzazione del Sistema

Il team ha esaminato sistematicamente i problemi rilevati nei test e nelle sessioni di utilizzo, adottando un approccio iterativo. Tra le azioni intraprese vi sono state la correzione di caratteri speciali negli URL, la risoluzione di duplicazioni nei commenti dell'ontologia e la sistemazione delle etichette delle codelist. Per la validazione, è stato analizzato il comportamento dei file di schema e implementate soluzioni per garantire la conformità.

Sono stati eliminati errori critici che compromettevano il funzionamento del sistema, migliorando la qualità e la sicurezza complessiva. La validazione semantica e XSD ora funziona correttamente, e l'esperienza utente è stata resa più fluida grazie alla sistemazione delle etichette e degli URL.

7.6 Aggiornamenti e Deploy dell'Applicazione

Il team ha mantenuto un ciclo continuo di aggiornamenti e deploy per introdurre nuove funzionalità e risolvere bug. Ogni aggiornamento è stato preceduto da test e verifiche approfondite per garantire la stabilità del sistema. In caso di problemi critici, sono state effettuate analisi post-deploy per identificare e correggere rapidamente gli errori.

L'applicazione è stata aggiornata regolarmente, migliorando la sua stabilità e integrando funzionalità richieste dagli utenti. Il processo di deploy è stato reso più affidabile, con una gestione efficace degli errori emersi durante le operazioni.

7.7 Documentazione e Report

È stata prodotta una documentazione dettagliata delle attività svolte, arricchita da immagini comparative tra vecchie e nuove interfacce e sono stati registrati una serie di video dettagliati ed esplicativi esplicativi sul funzionamento del codice.

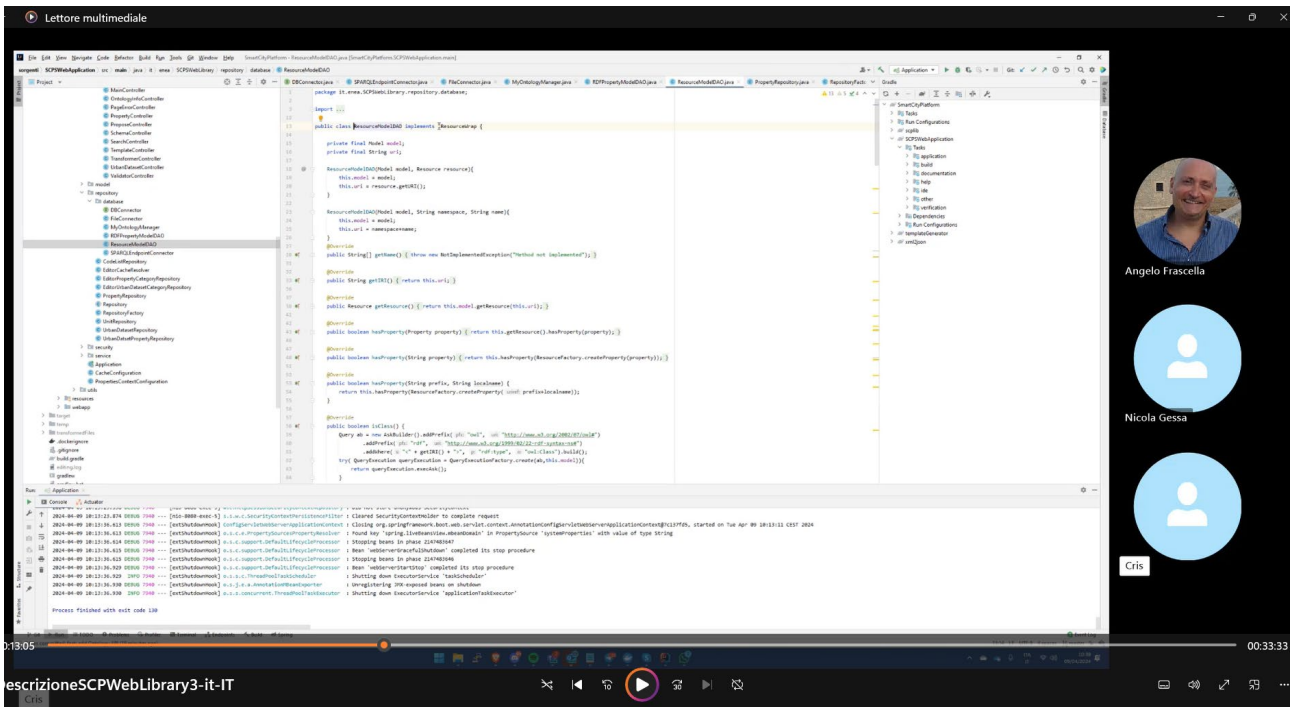


Figura 5. Fotogramma da uno dei video di analisi del codice

La documentazione ha reso il progetto facilmente comprensibile e trasferibile, agevolando il passaggio di consegne tra i collaboratori. I report dettagliati forniscono una base solida per eventuali revisioni o ampliamenti futuri dell'applicazione.

7.8 Progettazione Futura

Il team ha delineato diverse aree di miglioramento per il futuro, tra cui l'introduzione di notifiche automatizzate per gli amministratori, il supporto a formati avanzati come JSON-LD e un ulteriore affinamento del sistema di versionamento. Sono stati inoltre suggeriti miglioramenti per l'integrazione con applicazioni esterne e per la gestione delle query.

Sono state gettate le basi per uno sviluppo continuo dell'applicazione, con una visione chiara delle priorità e delle opportunità di innovazione. Questo garantirà una maggiore flessibilità e scalabilità del sistema nel lungo termine.

8 Conclusione

Il progetto ha prodotto risultati significativi in termini di stabilità, funzionalità e qualità del software. Ogni attività è stata affrontata con un approccio collaborativo e orientato ai risultati, consentendo di sviluppare un'applicazione robusta e pronta per ulteriori miglioramenti. Le solide basi teoriche e pratiche poste durante il progetto rappresentano un valore aggiunto per i futuri sviluppi.